Juniper Networks | Education Services

Advanced Junos Service Provider Routing

STUDENT GUIDE

Revision V19A

Education Services Courseware

Advanced Junos Service Provider Routing

V19A

Student Guide
Volume 2 of 2

Education Services

1133 Innovation Way
Sunnyvale, CA 94089 USA
408-745-2000
www.juniper.net

Course Number: EDU-JUN-AJSPR

This document is produced by Juniper Networks, Inc.

This document or any part thereof may not be reproduced or transmitted in any form under penalty of law, without the prior written permission of Juniper Networks Education Services.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Advanced Junos Service Provider Routing Student Guide, Revision V19A

Copyright © 2020 Juniper Networks, Inc. All rights reserved.

Printed in USA.

Revision History:

Revision 10.a — March 2011

Revision 10.b—September 2011

Revision 11.a—January 2012

Revision 12.a—September 2013

Revision V-17.a—June 2017

Revision V19A—February 2020

The information in this document is current as of the date listed above.

The information in this document has been carefully verified and is believed to be accurate for software Release 19.3. Juniper Networks assumes no responsibilities for any inaccuracies that may appear in this document. In no event will Juniper Networks be liable for direct, indirect, special, exemplary, incidental, or consequential damages resulting from any defect or omission in this document, even if advised of the possibility of such damages.

Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

YEAR 2000 NOTICE

Juniper Networks hardware and software products do not suffer from Year 2000 problems and hence are Year 2000 compliant. The Junos operating system has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using Juniper Networks software are described in the software license provided with the software, or to the extent applicable, in an agreement executed between you and Juniper Networks, or Juniper Networks agent. By using Juniper Networks software, you indicate that you understand and agree to be bound by its license terms and conditions. Generally speaking, the software license restricts the manner in which you are permitted to use the Juniper Networks software, may contain prohibitions against certain uses, and may state conditions under which the license is automatically terminated. You should consult the software license for further details.

Contents

Chapter 8: Multilevel IS-IS Networks 8-1
Level 1 and Level 2 Operations 8-3
Multilevel Configuration 8-9
Lab: Configuring a Multilevel IS-IS Network 8-30

Chapter 9: Troubleshooting IS-IS 9-1
IS-IS Overview 9-3
Case Study 9-21
Lab: Troubleshooting IS-IS 9-31

Chapter 10: BGP 10-1
Review of BGP 10-3
BGP Operations 10-12
BGP Path Selection 10-17
Load Balancing Options 10-20
Configuration Options 10-27
Lab: Configuring BGP 10-45

Chapter 11: BGP Attributes and Policy—Part 1 11-1
BGP Policy Overview 11-3
BGP Attributes Overview 11-9
Local Preference 11-14
Using Local Preference for Cold Potato Routing 11-22
AS Path 11-29
Lab: Configure and Manipulate BGP Attributes 11-54

Chapter 12: BGP Attributes and Policy—Part 2 12-1
Origin 12-3
Multi Exit Discriminator 12-7
Communities 12-12
Lab: BGP Attributes: Local Preference and Communities 12-43

Chapter 13: Route Reflection and Confederations 13-1
Route Reflection Operation 13-3
Configuration of Route Reflection 13-8
Virtual Route Reflector 13-16
Optimal Route Reflection 13-18
Route Reflection Case Study 13-24
BGP Confederations 13-33
Lab: Scaling BGP 13-41

Chapter 14: BGP FlowSpec 14-1
DDoS Overview 14-3
DDoS Mitigation Techniques 14-6
FlowSpec Overview 14-12
Configuring and Monitoring FlowSpec 14-19
Lab: BGP FlowSpec 14-39

Chapter 15: Troubleshooting BGP 15-1
BGP Troubleshooting 15-3
BGP Case Study 15-21
Lab: Troubleshooting BGP 15-37

Chapter 16: Policy Troubleshooting 16-1
Routing Policy Overview 16-3
Policy Structure 16-8
Using Regular Expressions 16-18
Useful Commands 16-25
Case Studies 16-31
Lab: Troubleshooting Routing Policy 16-40

Appendix A: BGP Route Damping A-1
Route Flap and Damping Overview A-3
Route Damping Parameters A-8
Configuring and Monitoring Route Damping A-13
Lab: BGP Route Damping A-23

Appendix B: Lab Diagrams B-1
Lab Diagram: Network Management B-2
Lab Network Diagram: Configuring OSPF B-2
Lab Network Diagram: Advanced OSPF B-3
Lab Network Diagram: Advanced OSPF Options – Parts 1-3 . . . . . B-3
Lab Network Diagram: Advanced OSPF Options – Part 4 . . . . . B-4
Lab Network Diagram: Troubleshooting OSPF B-4
Lab Network Diagram: IS-IS Configuration and Monitoring B-5
Lab Network Diagram: Advanced IS-IS Configuration Options and Routing Policy B-5
Lab Network Diagram: Configuring a Multilevel IS-IS Network B-6
Lab Network Diagram: Troubleshooting IS-IS B-6
Lab Network Diagram: Configuring BGP B-7
Lab Network Diagram: BGP Attributes — Part 1 . . . . . B-7
Lab Network Diagram: BGP Attributes — Part 2 . . . . . B-8
Lab Network Diagram: Scaling BGP — Parts 1-2 . . . . . B-8
Lab Network Diagram: Scaling BGP — Part 3 . . . . . B-9
Lab Network Diagram: BGP FlowSpec B-9
Lab Network Diagram: BGP Troubleshooting B-10
Lab Network Diagram: Troubleshooting Routing Policy B-10
Lab Network Diagram: BGP Route Damping B-11

Acronym List ACR-1

Course Overview

This five-day course is designed to provide students with detailed coverage of OSPF, IS-IS, BGP, and routing policy. Through demonstrations and hands-on labs, students will gain experience in configuring, monitoring, and troubleshooting the Junos operating system and in monitoring device and protocol operations.

This course uses Juniper Networks vMX Series Routers for the hands-on component, but the lab environment does not preclude the course from being applicable to other Juniper hardware platforms running the Junos OS. This course is based on the Junos OS Release 19.3.

Course Level

Advanced Junos Service Provider Routing (AJSPR) is an advanced-level course.

Intended Audience

This course benefits individuals responsible for implementing, monitoring, and troubleshooting Layer 3 components of a service provider's network.

Prerequisites

Students should have intermediate-level networking knowledge and an understanding of the Open Systems Interconnection (OSI) model and the TCP/IP protocol suite. Students should also attend the *Introduction to the Junos Operating System* (IJOS) and *Junos Intermediate Routing* (JIR) courses prior to attending this class.

Objectives

After successfully completing this course, you should be able to:

- Describe the various OSPF link-state advertisement (LSA) types.
- Explain the flooding of LSAs in an OSPF network.
- Describe the shortest-path-first (SPF) algorithm.
- Explain OSPF link metrics.
- Describe the various OSPF authentication methods.
- Explain the key differences between OSPFv2 and OSPFv3.
- Describe OSPF area types and operations.
- Configure various OSPF area types.
- Summarize and restrict routes.
- Configure OSPF multi-area adjacencies
- Configure OSPF virtual links.
- Explain OSPF external reachability.
- List useful commands that are used to troubleshoot and verify OSPF.
- Isolate different OSPF issues.
- Explain the concepts and operation of IS-IS.
- Describe various IS-IS link-state protocol data unit (LSP) types.
- List IS-IS adjacency rules and troubleshoot common adjacency issues.
- Configure and monitor IS-IS.
- Display and interpret the link-state database (LSDB).
- Perform advanced IS-IS configuration options.
- Implement IS-IS routing policy.
- Explain the default operation in multilevel IS-IS.
- Describe address summarization methods used in IS-IS.

- Describe address summarization methods used in IS-IS.
- Configure and monitor a multilevel IS-IS network.
- List useful commands to troubleshoot and verify IS-IS problems.
- Troubleshoot and isolate different IS-IS issues.
- Describe basic BGP operation.
- List common BGP attributes.
- Explain the route selection process for BGP.
- Describe how to alter the route selection process.
- Configure some advanced options for BGP peers.
- Explain how policies function in BGP.
- Describe BGP attributes and explain how these attributes can be used to manipulate traffic.
- Show how Regex can be used in policies to manipulate AS-path.
- Describe the BGP attributes Origin, MED and communities in detail and explain the operation of those attributes.
- Manipulate these BGP attributes using routing policy.
- Describe the operation of BGP route reflection.
- Configure a route reflector.
- Describe the virtual route reflector.
- Describe the operation of optimal route reflection.
- Describe the operation of a BGP confederation.
- Configure confederations.
- Describe peering relationships in a confederation.
- Describe DDoS attacks.
- Describe DDoS mitigation techniques.
- Describe FlowSpec Operations.
- Configure and Monitor FlowSpec.
- Review common BGP troubleshooting procedures.
- List common BGP troubleshooting commands.
- Identify issues with BGP peering.
- Isolate problems on routing policy structure and configuration.
- Identify common commands for troubleshooting routing policy.
- Explain the causes for route instability.
- Describe the effect of damping on BGP routing.
- Explain the default behavior of damping on links.
- Control damping using routing policy.
- View damped routes using command-line interface (CLI) commands.

Course Agenda

Day 1

Chapter 1: Course Introduction

Chapter 2: OSPF

Lab 1: Configuring OSPF

Chapter 3: OSPF Areas

Lab 2: OSPF Route Summarization

Chapter 4: Advanced OSPF Options

Lab 3: Advanced OSPF

Day 2

Chapter 5: Troubleshooting OSPF

Lab 4: Troubleshooting OSPF

Chapter 6: IS-IS

Lab 5: IS-IS Configuration and Monitoring

Chapter 7: Advanced IS-IS Operations

Lab 6: Advanced IS-IS Configuration Options

Day 3

Chapter 8: Multilevel IS-IS Networks

Lab 7: Configuring a Multilevel IS-IS Network

Chapter 9: Troubleshooting IS-IS

Lab 8: Troubleshooting IS-IS

Chapter 10: BGP

Lab 9: Configuring BGP

Day 4

Chapter 11: BGP Attributes and Policy—Part 1

Lab 10: BGP Attributes—Part 1

Chapter 12: BGP Attributes and Policy—Part 2

Lab 11: BGP Attributes—Part 2

Chapter 13: Route Reflection and Confederations

Lab 12: Scaling BGP

Day 5

Chapter 14: BGP FlowSpec

Lab 13: BGP FlowSpec

Chapter 15: Troubleshooting BGP

Lab 14: Troubleshooting BGP

Chapter 16: Troubleshooting Policy

Lab 15: Troubleshooting Routing Policy

Appendix A: BGP Route Damping

Lab AA: BGP Route Damping

Document Conventions

CLI and GUI Text

Frequently throughout this course, we refer to text that appears in a command-line interface (CLI) or a graphical user interface (GUI). To make the language of these documents easier to read, we distinguish GUI and CLI text from chapter text according to the following table.

Style	Description	Usage Example
Franklin Gothic	Normal text.	Most of what you read in the Lab Guide and Student Guide.

`Courier New`	Console text:
	• Screen captures
	• Noncommand-related syntax
	GUI text elements:
	• Menu names
	• Text field entry
		
		`commit complete`
		`Exiting configuration mode`
		
		Select `File > Open`, and then click `Configuration.conf` in the `Filename` text box.

Input Text Versus Output Text

You will also frequently see cases where you must enter input text yourself. Often these instances will be shown in the context of where you must enter them. We use bold style to distinguish text that is input versus text that is simply displayed.

Style	Description	Usage Example
`Normal CLI`	No distinguishing variant.	`Physical interface:fxp0, Enabled`
`Normal GUI`		View configuration history by clicking `Configuration > History`.
`CLI Input`	Text that you must enter.	`lab@San_Jose>` **`show route`**
`GUI Input`		Select `File > Save`, and type **`config.ini`** in the `Filename` field.

Undefined Syntax Variables

Finally, this course distinguishes syntax variables, where you must assign the value (undefined variables). Note that these styles can be combined with the input style as well.

Style	Description	Usage Example
`CLI Undefined` *`GUI Undefined`*	Text where the variable's value is the user's discretion or text where the variable's value as shown in the lab guide might differ from the value the user must input according to the lab topology.	Type **`set policy`** ***`policy-name`***. **`ping 10.0.`*****`x.y`*** Select `File > Save`, and type ***`filename`*** in the `Filename` field.

Additional Information

Education Services Offerings

You can obtain information on the latest Education Services offerings, course dates, and class locations from the World Wide Web by pointing your Web browser to: http://www.juniper.net/training/education/.

About This Publication

This course was developed and tested using the software release listed on the copyright page. Previous and later versions of software might behave differently so you should always consult the documentation and release notes for the version of code you are running before reporting errors.

This document is written and maintained by the Juniper Networks Education Services development team. Please send questions and suggestions for improvement to training@juniper.net.

Technical Publications

You can print technical manuals and release notes directly from the Internet in a variety of formats:

- Go to http://www.juniper.net/techpubs/.
- Locate the specific software or hardware release and title you need, and choose the format in which you want to view or print the document.

Documentation sets and CDs are available through your local Juniper Networks sales office or account representative.

Juniper Networks Support

For technical support, contact Juniper Networks at http://www.juniper.net/customers/support/, or at 1-888-314-JTAC (within the United States) or 408-745-2121 (from outside the United States).

Education Services

Advanced Junos Service Provider Routing

Chapter 8: Multilevel IS-IS Networks

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - Explain the default operation in a multilevel IS-IS network
  - Describe address summarization methods used in IS-IS
  - Configure and monitor a multilevel IS-IS network

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- The default operation in a multilevel IS-IS network;
- IS-IS address summarization methods; and
- The configuration and monitoring of a multilevel IS-IS network.

Agenda: Multilevel IS-IS Networks

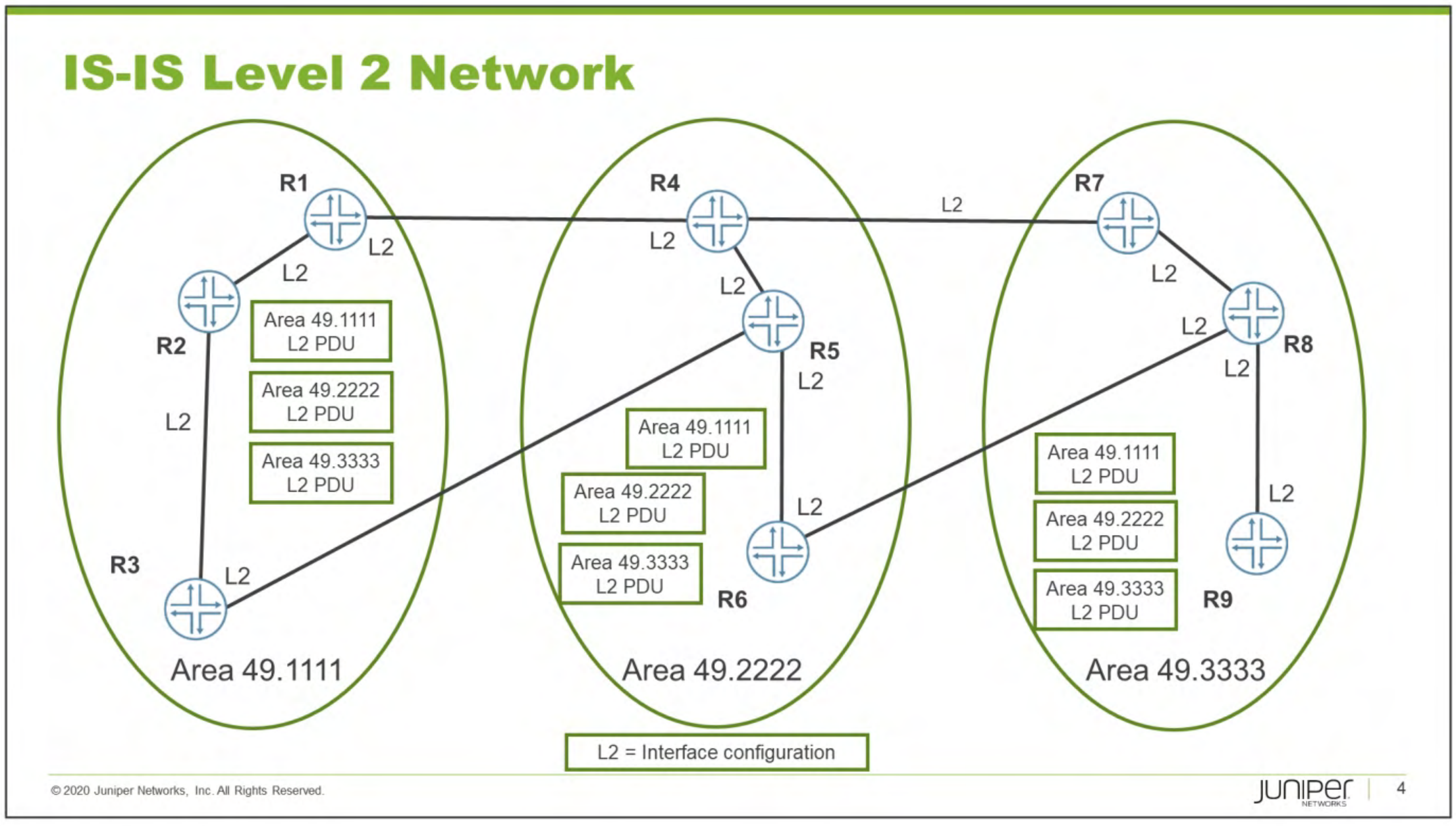
→Level 1 and Level 2 Operations

- Multilevel Configuration

© 2020 Juniper Networks, Inc. All Rights Reserved.

Level 1 and Level 2 Operations

The slide lists the topics we will discuss. We discuss the highlighted topic first.

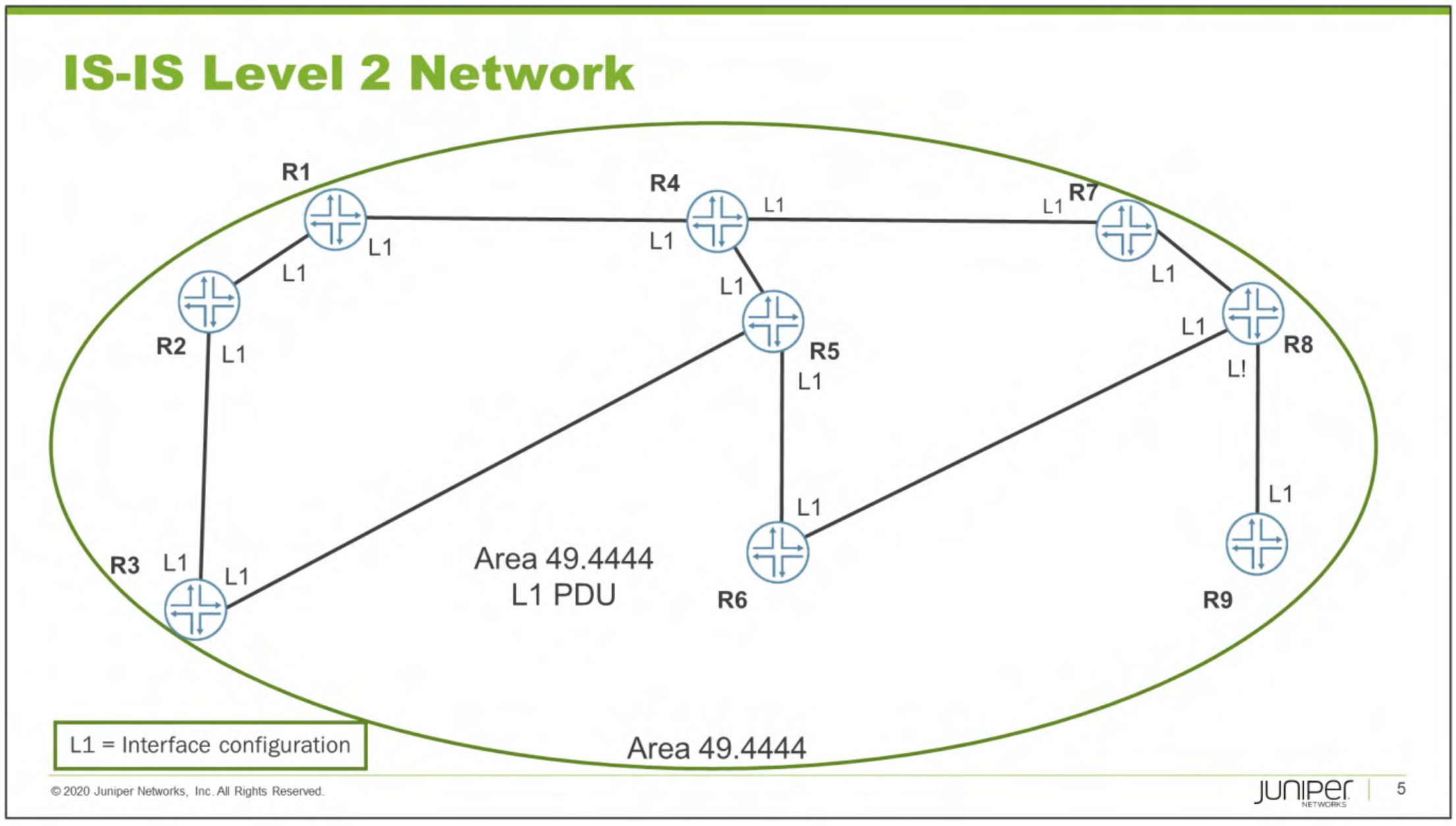


IS-IS Level 2 Operation

As discussed in a previous chapter, the IS-IS protocol advertises either a Level 1 LSP or a Level 2 LSP for each adjacency formed with a neighbor. The type of LSP advertised depends on the level at which the adjacency is formed.

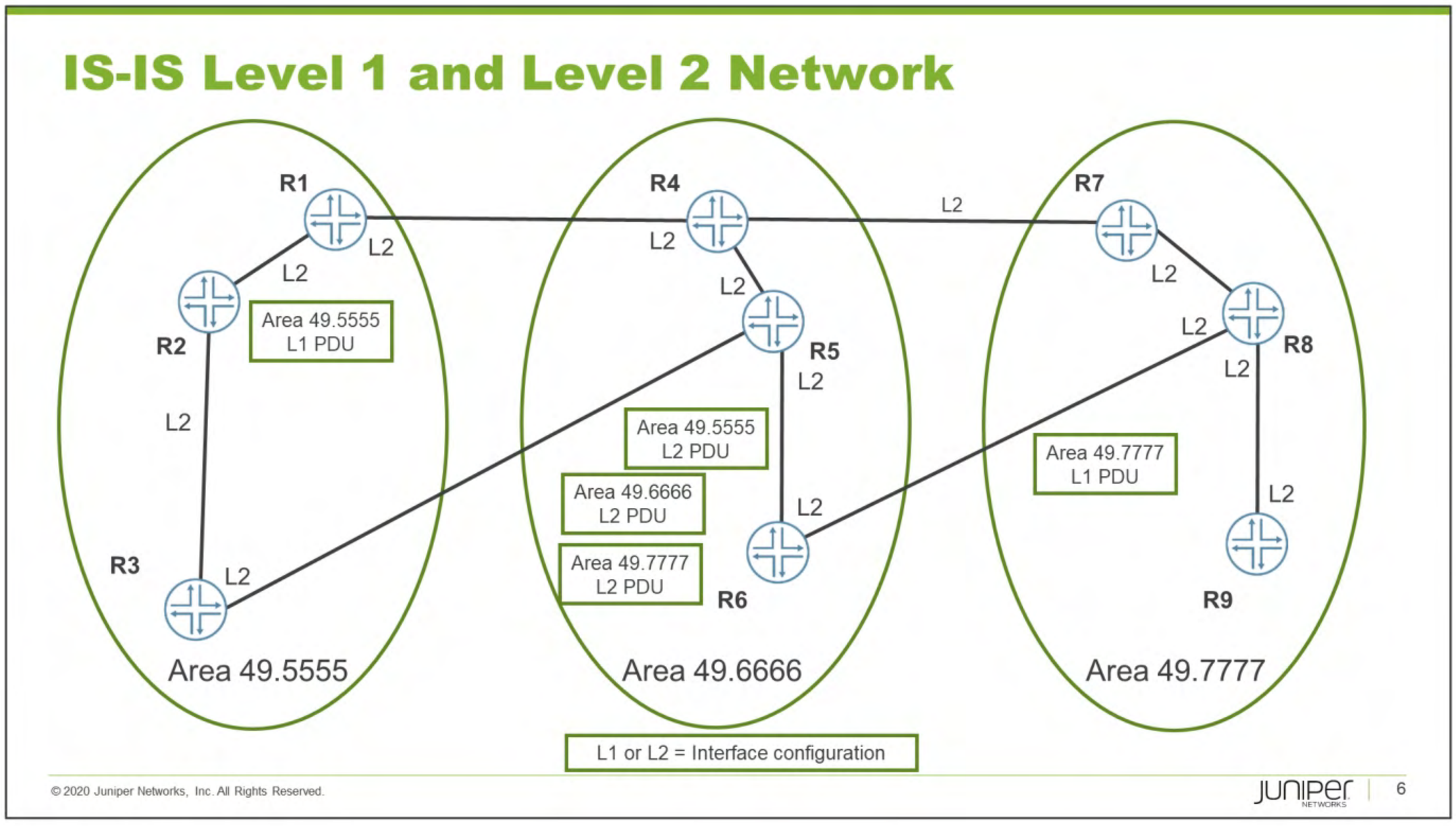
Also recall that an IS-IS Level 1 LSP can be flooded only within a specific area because a Level 1 adjacency cannot form across an area boundary. Level 2 LSPs include the routing information carried in Level 1 LSPs, which results in the L2 backbone knowing routes for all areas and levels.

In the example on the slide, routing information from all routers is present in all databases in the network. The presence of a single L2 database shared by all routers occurs because all of the adjacencies in the network are at Level 2, and Level 2 LSPs are flooded both within, and across, IS-IS area boundaries.



IS-IS Level 1 Operation

This slide details a single area Level 1 IS-IS network. In this example, all routers in the network share a Level 1 database containing identical information. The presence of a common Level 1 database in all routers occurs in this case because all adjacencies are Level 1 in nature, and all routers are within the same IS-IS area (49.4444). Level 1 LSP flooding will reach all routers in the network because of the single area.



Multilevel IS-IS Operation

In this example, routing information for each router is present in all Level 2 databases in the network. This routing information is present because Level 1 routing information can be summarized at the L1/L2 boundary and flooded throughout the Level 2 backbone in Level 2 link-state protocol data units (PDUs). The Level 1 routers within each Level 1 area have a single Level 1 database that contains routing information for that area only. The Level 1 routers use the attached bit in an advertised Level 1 LSP to install a local default route. The Level 1 router forwards packets to the metrically closest attached router when routing to destinations outside of their Level 1 area.

Level 1 routers are isolated from routing changes in other areas, and summarization of Level 1 information prevents Level 2 routers from having to perform a full SPF calculation for topology changes within a Level 1 area. This isolation and summarization of routing information improves the scalability of a multilevel IS-IS network.

Multilevel Operation

- An L1/L2 IS-IS network operates in a similar fashion to an OSPF NSSA with no summaries
  - Local L1 routes are advertised into Level 2
  - External routes can be advertised into an L1 area
- The L1/L2 border is a natural route boundary
  - L2 routes are not advertised into L1 areas by default
  - External L1 routes are not advertised to Level 2 by default
    - Route leaking policies are used to modify this default behavior
    - Using only wide metrics eliminates internal/external distinction
- L1/L2 attached routers set the attached bit in their L1 LSPs
  - L1 routers install a locally generated 0/0 default route to the closest L2 attached router
    - Disable with **`ignore-attached-bit`** command

© 2020 Juniper Networks, Inc. All Rights Reserved.

Multilevel IS-IS Operation Is Similar to OSPF NSSA

You can compare the operation of a multiarea IS-IS network to an OSPF not-so-stubby area (NSSA) with the **`no-summaries`** and **`default-metric`** options configured. In a multiarea IS-IS network, each Level 1 IS-IS router has complete routing knowledge of the routes local to its Level 1 area *only*. Level 1 routers reach other IS-IS destinations by using a 0.0.0.0/0 default route generated by the detection of L1/L2 attached routers. As with an OSPF NSSA, you can inject external routing information into the Level 1 area. The Level 2 LSPs of the attached routers in the area advertise the internal Level 1 routes to other IS-IS Level 2 areas.

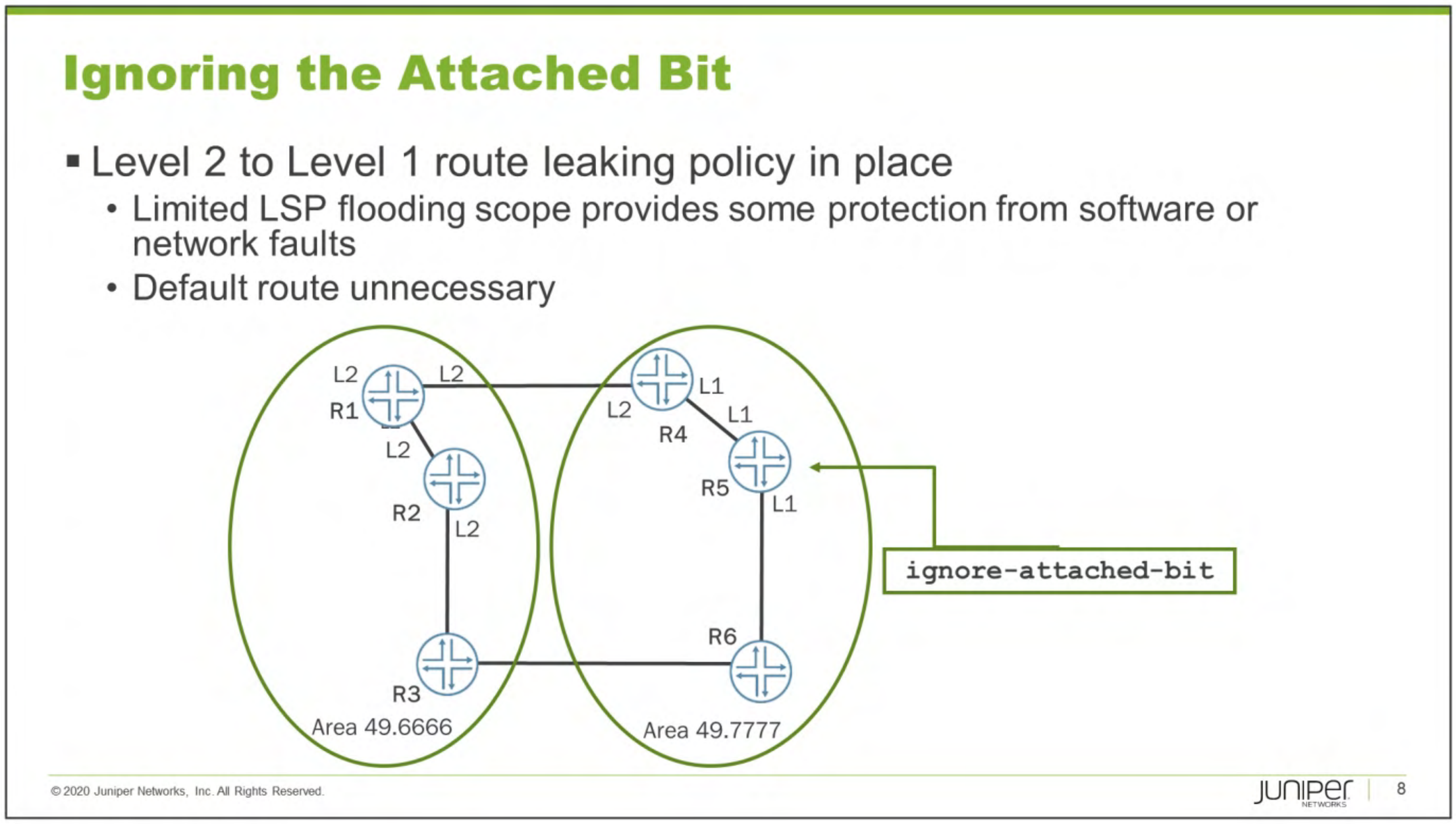
L1/L2 Border Router Is a Natural Boundary

Although a Level 2 LSP advertises all Level 1 internal routes, routing information for the Level 2 backbone is constrained by the L1/L2-attached router. Thus, Level 2 routes are *not* advertised into the Level 1 area by default; hence the need for a default route in the Level 1 area. Level 1 routes advertised as external routes into Level 1 are not advertised to any Level 2 routers by default; routing policy is needed to effect the leaking of Level 1 externals into the L2 backbone. Note that the use of **`wide-metrics-only`** alters the *natural* L1/L2 boundary in that routes are no longer distinguishable as being internal or external. The use of wide metrics therefore results in the automatic leaking of all Level 1 routes into Level 2, because they will all appear to be internal routes.

L2 Routers Set the Attached Bit

To provide interarea reachability for Level 1 routers, an L1/L2 router with a Level 2 adjacency to a router in another area sets its attached bit in its Level 1 LSPs. Level 1 routers install a 0.0.0.0/0 default route to the metrically closest attached router when they detect Level 1 LSPs with the attached bit set. Note that while each possible metric type (default, delay, expense, and error) is associated with its own attached bit, Junos supports only the default metric type.

You can disable the generation of a default route by including the **`ignore-attached-bit`** statement at the `[edit protocols isis]` configuration hierarchy.



Ignoring the Attached Bit

In some corner cases, you might want to prevent the installation of a default route based on the presence of Level 1 LSPs with attached bits. The slide provides an example of one such application in which a multilevel IS-IS network with Level 2 to Level 1 route leaking in place.

Because the leaking of Level 2 routing information into the Level 1 areas provides all routers with complete IS-IS routing information, a default route is no longer needed for routing to destinations outside of a given Level 1 area. Because the goal of a multilevel IS-IS design is normally to reduce database size for routers in Level 1 areas, you might ask yourself why someone would design a multilevel IS-IS topology only to leak Level 2 routes into Level 1.

In this example, the network operator wants to leverage the built-in LSP flooding scope of a multilevel IS-IS network to provide some level of isolation in the event that a malformed LSP is generated. For example, if a malformed Level 1 LSP is generated in area 49.7777, this LSP will not be flooded into the Level 2 backbone (the contents of Level 1 LSPs are repackaged into a Level 2 LSP for submission to the Level 2 backbone by an attached router, but the Level 1 LSP itself is not flooded into Level 2).

Another application for the **`ignore-attached-bit`** option relates to the fact that using the metrically closest attached router might not always yield optimal interarea routing. In these cases, it might be desirable to use a locally defined static or generated route, in which case the IS-IS derived default route might no longer be needed.

Agenda: Multilevel IS-IS Networks

- Level 1 and Level 2 Operations
- → Multilevel Configuration

© 2020 Juniper Networks, Inc. All Rights Reserved.

Multilevel Configuration

The slide highlights the topic we discuss next.

IS-IS Multilevel Configuration

- Each IS-IS interface operates at both Level 1 and Level 2 by default
  - Disable a specific level to stop an interface from operating at that level
  - `lo0` interface will be passive at both levels in this example
    - Disable at a particular level to prevent `lo0` address advertisement in that level

```
protocols {
    isis {
        interface so-0/0/0.0 {
            level 1 disable;
        }
        interface ge-0/1/0.0 {
            level 2 disable;
        }
        interface lo0.0;
    }
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

IS-IS Interfaces Operate in L1/L2 Mode

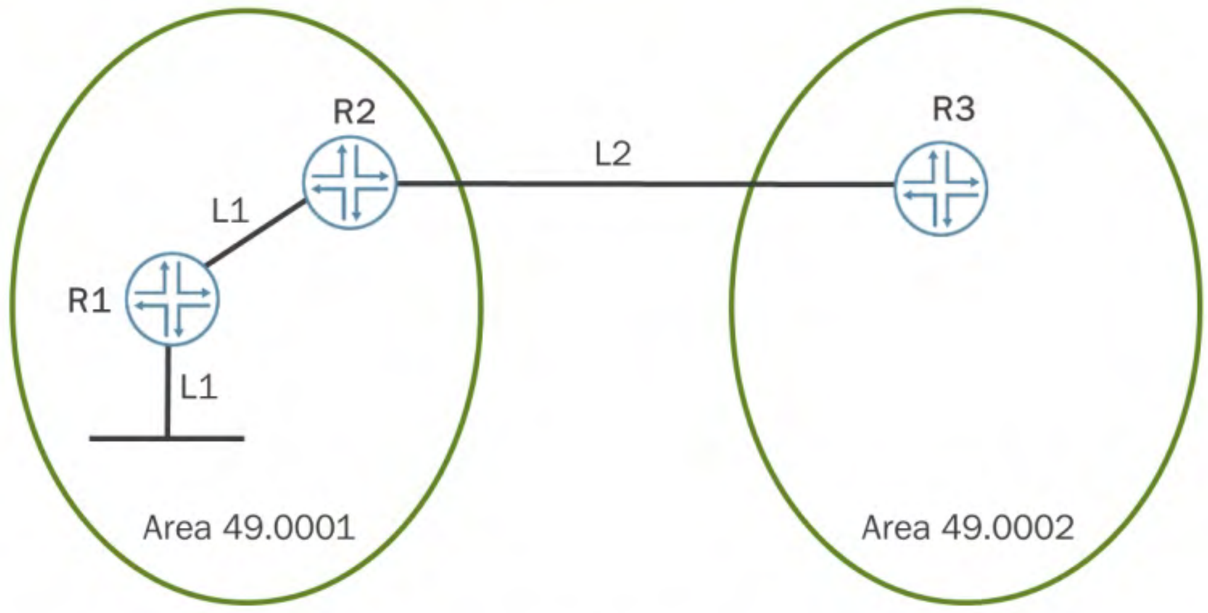
The default operation of the IS-IS protocol within Junos is to enable both Level 1 and Level 2 capabilities for all interfaces. This default behavior is designed to promote connectivity with all neighbors. If an adjacency can be formed between two routers, it will. One consequence of this default, however, is that you might form both a Level 1 and a Level 2 relationship with a given neighbor, which results in two separate adjacencies and two separate LSP flooding topologies.

To disable the operation of a particular level on an interface, use the **`disable`** keyword as shown on the slide. The `so-0/0/0.0` interface only operates at Level 2, and the `ge-0/1/0.0` interface only operates at Level 1. As a shortcut, you can disable all Level 1 or Level 2 processing on the router, which will result in all interfaces being Level 2, or Level 1, respectively. For example, the **`set protocols isis level 1 disable`** statement will result in all interfaces operating at Level 2 only.

We recommend that you explicitly configure the `lo0.0` interface within the IS-IS protocol, even when the router's network entity title (NET) is assigned to another interface. Although its omission does not harm the operational aspects of IS-IS (adjacencies still form), the IP address configured on the `lo0` interface will not be advertised in TLV 128 or TLV 135, making the loopback address unreachable. Note that in most cases you must run the IS-IS protocol on the `lo0` interface for proper operation because the router's NET is normally assigned to loopback interface for resiliency reasons.

Because the loopback interface operates in passive mode, you do not need to disable a particular level on that interface. By default, the IP address on the interface is advertised in both the Level 1 and Level 2 LSPs generated by the router. You can restrict the advertisement of the router's loopback address in a particular level by disabling that level in the `lo0.0` statement in the `isis` stanza.

Case Study: Route Leaking



- Level 2 routes are to be advertised into Area 49.0001
  - Requires routing policy on L1/L2 area border router R2
  - Use the **`from level 2/to level 1`** syntax

© 2020 Juniper Networks, Inc. All Rights Reserved.

Case Study: Route Leaking

As previously discussed, Level 2 routes are not advertised into Level 1 areas by default. In this example, the network operator wants to advertise, or *leak*, Level 2 routes into Area 49.0001. This action will require a routing policy on R2, the L1/L2 area border router (ABR), specifying that the matching routes are Level 2 and will be advertised in Level1.

Route Leaking

- Use a policy to advertise (leak) routes across an L1/L2 area border
- Routes advertised from an L2 area into an L1 area have the up/down bit set to down
  - Ensures that another L2 router will *not* re-advertise the route back into an L2 area to avoid routing loops

```
[edit policy-options]
user@R2# show
policy-statement route-leak {
    term L2-to-L1 {
        from {
            protocol isis;
            level 2;
            route-filter 192.168.16.0/20 orlonger;
        }
        to level 1;
        then accept;
    }
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Policy Used to Advertise Routes

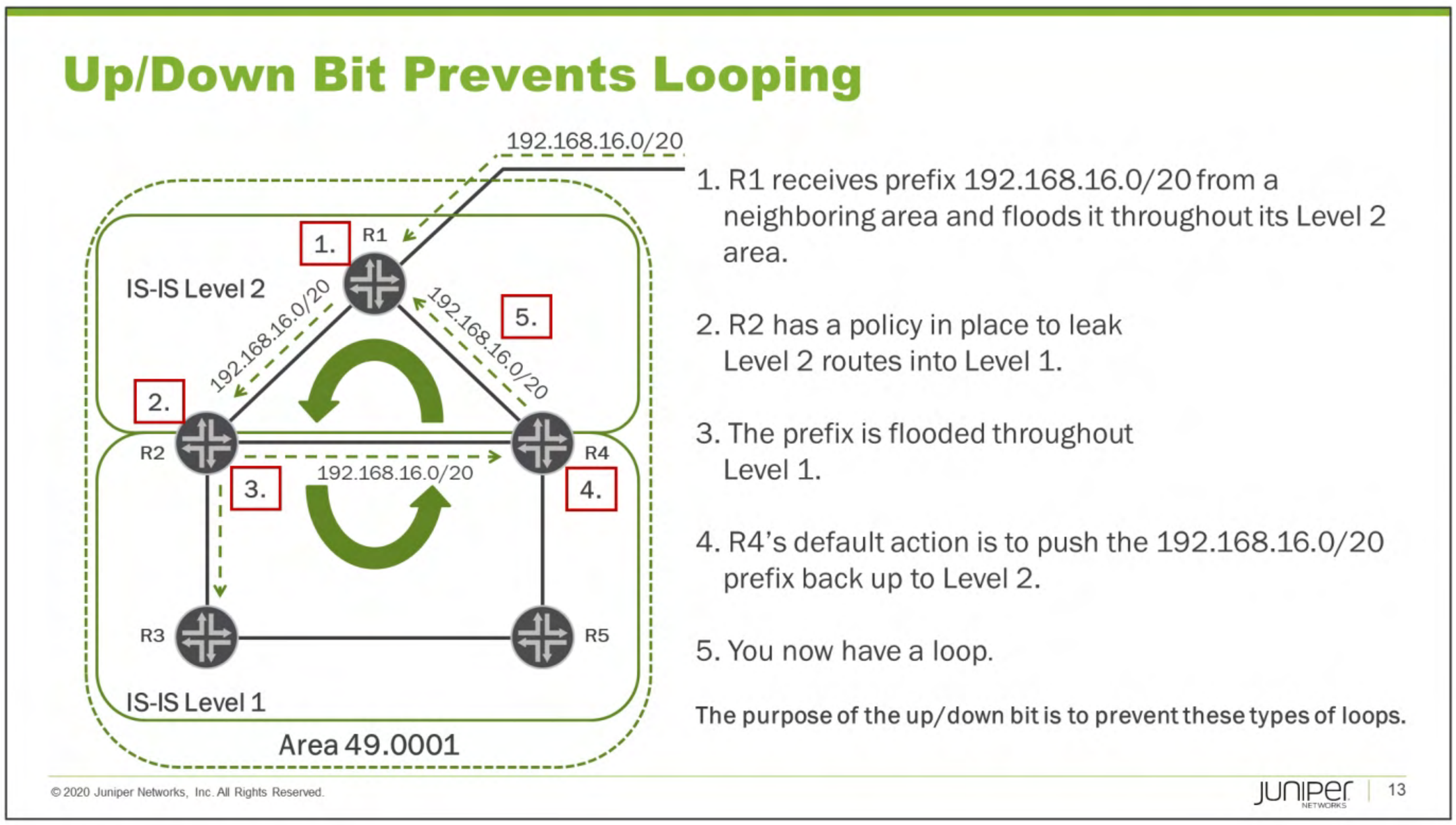
Because the L1/L2 border router naturally stops the transmission of Level 2 routes into a Level 1 area, it is the logical location to override that default. You can accomplish this goal with a Junos routing policy.

You configure this policy within the `[edit policy-options]` configuration hierarchy, and then apply the policy to the IS-IS instance at the global IS-IS level, that is, `[edit protocols isis]`.

In the example on the slide, the match criterion within the `route-leak` policy is all IS-IS routes within the subnet 192.168.16.0/20 that are currently Level 2 routes and are eligible to be sent to Level 1. Once these routes are found, the configured action is to accept these routes. The use of the **`from`** and **`to`** keywords allows granular control about the desired direction of route leaking.

Once the routing policy is exported into the IS-IS protocol, the Level 2 routes are inserted into the Level 1 LSP of the L1/L2 border router and are advertised into the Level 1 area.

Recall from a previous slide that the L1/L2 border router also blocks external Level 1 routes from being advertised into Level 2. A similar policy is used to advertise Level 1 external routes into the Level 2 backbone. This new policy simply reverses the Level 2 and Level 1 notations and makes use of an appropriate route filter statement. Once you apply this policy, the external routes are included in the Level 2 LSPs.



Up/Down Bit Prevents Looping

Previous slides described the default action of an L1/L2 router with regard to the advertisement of internal Level 1 routes within its Level 2 LSP. Conceptually, the policy referenced on the slide could interact with this default action to create a routing loop.

For example, consider the case illustrated on the slide. If R1 has a policy to advertise Level 2 routes into Level 1, then R1 will include the Level 2 routes in its Level 1 LSP. As this LSP is flooded throughout the Level 2 area, it eventually arrives at R2. R2 has a policy in place that will leak Level 2 LSPs into its Level 1. Eventually, this information makes its way around to R4. If R4 advertises the Level 2 routes back into Level 2, a routing loop can form.

The potential for route leaking-induced routing loops is averted by a bit in the LSP known as the up/down (U/D) bit. The purpose of this bit is to inform the L1/L2 routers whether a configured policy can advertise a route. Only routes marked with the up direction are eligible for advertisement from Level 1 to Level 2. All internal Level 1 routes will have the up/down bit set in this manner. If the up/down bit is set to down, the route has already been leaked from Level 2 into a Level 1 area and, as such, the route cannot be sent back into the Level 2 backbone by R4.

Route Summarization

- The L1/L2 area border is a natural place to summarize routing information
  - Override the default route flooding between the areas with a routing policy
- Create aggregate routes in local routing table
  - Policy required to advertise aggregate routes into another level—use **from /to level** for maximum control

```
[edit policy-options]
user@router# show policy-statement external-L1-summary-route
term on-the-L1L2-router {
    from {
        protocol aggregate;
        route-filter 172.16.20.0/22 exact;
    }
    to level 2;
    then accept;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Summarize Routes at the L1/L2 Router

Routes that are naturally bound by the L1/L2 border router are eligible for route summarization. These routes include external Level 1 routes and Level 2 routes from other IS-IS areas. In addition, you can also summarize internal Level 1 routes that are normally advertised into Level 2 automatically.

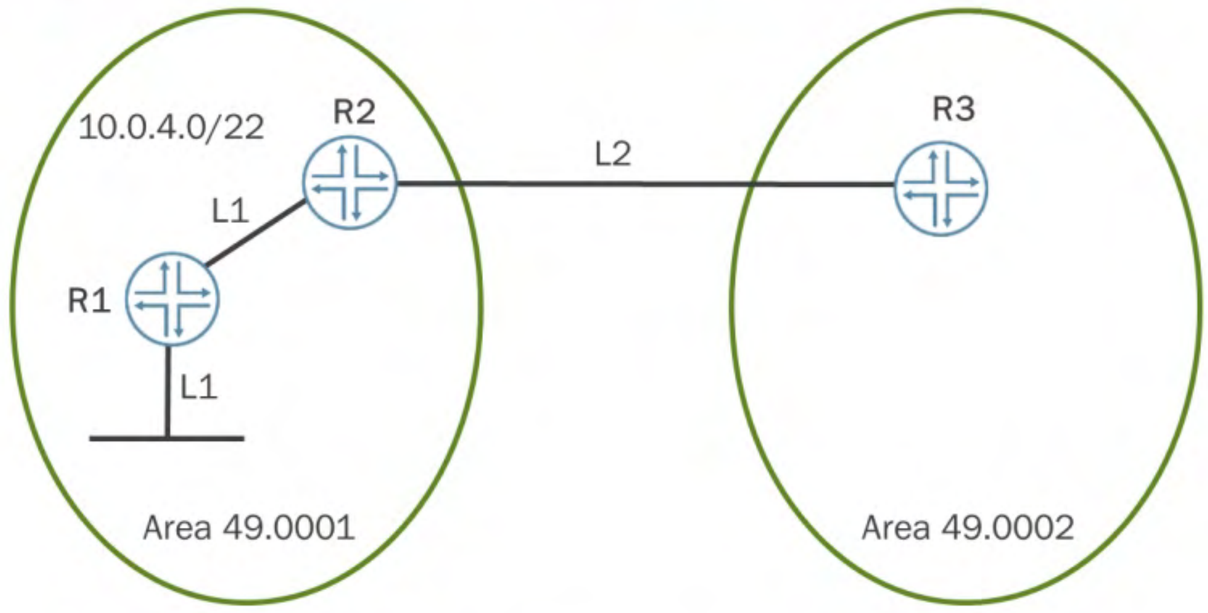
Create an Aggregate Route and Advertise It with Policy

No concept of an **area-range** command exists in IS-IS. To summarize routes, you must create an aggregate route on the L1/L2 border router within the `[edit routing-options]` hierarchy that encompasses the routes you want to summarize.

To advertise the aggregate route, you create a policy similar to the example shown on the slide. This policy is applied as an export to the IS-IS instance at the global `[edit protocols isis]` level. In this example, the goal is to advertise a 172.16.20.0/22 aggregate into the Level 2 backbone to represent Level 1 external routing information in the Level 1 area.

When summarizing routes from one level into another, you might need to alter the default IS-IS export policy to ensure that specific prefixes are not advertised along with the corresponding aggregate. You can accomplish the altering of the export policy with a **reject** action associated with a route filter that will match on the specific routes in question.

Case Study: Internal Level 1 Route Summarization



- Internal Level 1 routes can be summarized
  - Requires routing policy and a local aggregate route
  - For example, suppress specific routes in the 10.0.4.0/22 block and advertise a single 10.0.4.0/22 summary route

© 2020 Juniper Networks, Inc. All Rights Reserved. 15

Internal Level 1 Route Summarization

Internal Level 1 routes are automatically advertised in a Level 2 LSP into the Level 2 backbone. Junos provides a method for altering this default action with a routing policy. The example on the slide shows that the Level 1 Area 49.0001 contains multiple internal routes within the 10.0.4.0/22 address space. These routes are currently advertised individually to R3, as shown in the following output:

```
user@R3# show route 10.0.4/22

inet.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.4.12/30       *[IS-IS/18] 00:28:50, metric 20
                    > to 10.0.2.2 via ge-0/2/1.0
10.0.5.0/24        *[IS-IS/18] 00:28:50, metric 30
                    > to 10.0.2.2 via ge-0/2/1.0
10.0.6.1/32        *[IS-IS/18] 00:28:50, metric 20
                    > to 10.0.2.2 via ge-0/2/1.0
```

Administratively, we want to suppress these specific internal Level 1 routes while advertising a single 10.0.4.0/22 summary in their place. We accomplish this configuration by using a combination of a routing policy and a local aggregate route defined on R2.

Case Study: Level 1 Route Summarization Policy

- Sample policy for summarizing internal Level 1 routes
  - Requires local 10.0.4.0/22 aggregate definition on R2
  - Use of `to` ensures that summary route is not injected into the Level 1 area

```
[edit]
user@R2# show policy-options policy-statement internal-L1-summary-route
term local-summary-route {
    from {
        protocol aggregate;
        route-filter 10.0.4.0/22 exact;
    }
    to level 2;
    then accept;
}
term suppress-specifics {
    from {
        route-filter 10.0.4.0/22 longer;
    }
    to level 2;
    then reject;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Level 1 Route Summarization Policy

The sample policy shown on the slide meets our administrative requirements of advertising only a single summary route for the internal Level 1 routes. The first term in the policy matches and accepts the locally defined summary route on R2 for advertisement to the Level 2 backbone. The second policy term serves to override the default IS-IS export policy for routes matching the 10.0.4.0/22 route filter. It specifies that these routes will not be advertised to R3 in the Level 2 LSP generated by R2.

After applying the **`internal-L1-summary-route`** policy as an export policy in R2's IS-IS instance, we can confirm its success on R3:

```
user@R3# show route 10.0.4/22

inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.4.0/22        *[IS-IS/165] 00:00:20, metric 20
                    > to 10.0.2.2 via ge-0/2/1.0
```

Applying IS-IS Policies

- Apply IS-IS policies at the global level of the **`isis`** stanza
  - Multiple export polices can be applied, or a single policy with multiple terms can be used

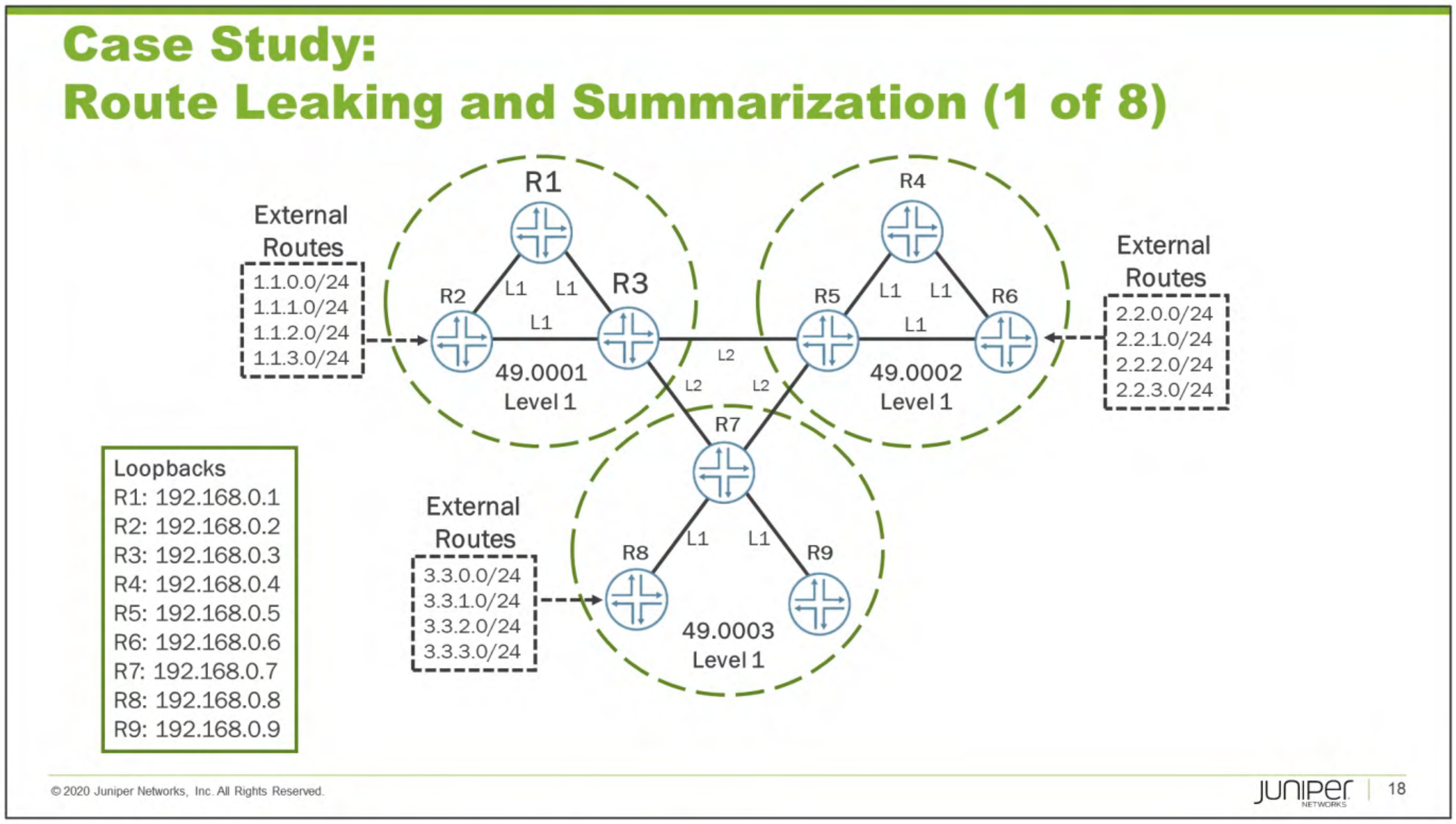
```
[edit protocols isis]
user@R2# show
export [ external-L1-summary-route internal-L1-summary-route ];
interface ge-0/2/0.0 {
    level 1 disable;
}
interface ge-0/2/1.0 {
    level 2 disable;
}
interface lo0.0;
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Apply Export Policy at Global Level of the IS-IS Instance

One or more export policies can be applied at the global level of an IS-IS instance, as shown on the slide. Both the `external-L1-summary-route` and `internal-L1-summary-route` policies will be used to control the routes advertised by the local router.

When wanted, you can apply multiple export policies to the same IS-IS instance. The same effect is normally possible through the use of a single policy containing multiple terms, but in some cases it might be easier to reuse existing policies in such a manner. Note that normal policy processing will proceed from left to right, and that policy processing will terminate once a given route meets with either an `accept` or `reject` action.



Case Study: Route Leaking and Summarization—Part 1

This slide shows the sample topology used for this case study. Current highlights include the following:

- There are three separate IS-IS Level 1 areas, with three routers per area.
- The R3, R5, and R7 routers are connected through IS-IS Level 2 connections.
- Three different sets of external routes are introduced through the R2, R6, and R8 routers.
- No route leaking or summarization is currently configured. That is, default IS-IS behavior is in effect.

Case Study: Route Leaking and Summarization (2 of 8)

- R3 router view

Visibility of external routes limited to own Level 1 area *only*

Visibility of *all* interconnect routes in the network

Visibility of *all* loopbacks in the network

```
user@R3> show route terse table inet.0

inet.0: 28 destinations, 28 routes (28 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A V Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* ? 1.1.0.0/24         I 160         10            >10.0.1.10
* ? 1.1.1.0/24         I 160         10            >10.0.1.10
* ? 1.1.2.0/24         I 160         10            >10.0.1.10
* ? 1.1.3.0/24         I 160         10            >10.0.1.10
* ? 10.0.1.0/30        D   0                       >ge-1/0/7.0
* ? 10.0.1.1/32        L   0                        Local
* ? 10.0.1.4/30        I  15         20             10.0.1.2
                                                   >10.0.1.10
* ? 10.0.1.8/30        D   0                       >ge-1/0/8.0
* ? 10.0.1.9/32        L   0                        Local
* ? 10.0.2.0/30        I  18         20            >10.0.10.2
* ? 10.0.2.4/30        I  18         30            >10.0.10.2
* ? 10.0.2.8/30        I  18         20            >10.0.10.2
* ? 10.0.3.0/30        I  18         20            >10.0.10.6
* ? 10.0.3.4/30        I  18         20            >10.0.10.6
* ? 10.0.10.0/30       D   0                       >ge-1/1/2.0
* ? 10.0.10.1/32       L   0                        Local
* ? 10.0.10.4/30       D   0                       >ge-1/0/4.0
* ? 10.0.10.5/32       L   0                        Local
* ? 10.0.10.8/30       I  18         20            >10.0.10.6
                                                    10.0.10.2
* ? 192.168.0.1/32     I  15         10            >10.0.1.2
* ? 192.168.0.2/32     I  15         10            >10.0.1.10
* ? 192.168.0.3/32     D   0                       >lo0.3
* ? 192.168.0.4/32     I  18         20            >10.0.10.2
* ? 192.168.0.5/32     I  18         10            >10.0.10.2
* ? 192.168.0.6/32     I  18         20            >10.0.10.2
* ? 192.168.0.7/32     I  18         10            >10.0.10.6
* ? 192.168.0.8/32     I  18         20            >10.0.10.6
* ? 192.168.0.9/32     I  18         20            >10.0.10.6
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Case Study: Route Leaking and Summarization—Part 2

First, we verify default IS-IS behavior is occurring on an arbitrary Level 2 router—R3 in this case. The sample shows that R3 has visibility to external routes originating in its own Level 1 area. Moreover, because it's a Level 2 router connected to the other Level 2 routers, it has visibility to all loopbacks and interconnect routes from all three areas. Outputs from the other Level 2 routers, R5 and R7, would be similar.

Case Study: Route Leaking and Summarization (3 of 8)

- R1 router view

Default route created from the attached bit from R3

Specific visibility of external routes limited to *own* Level 1 area *only*

Specific visibility of interconnect routes in *own* Level 1 area only

Specific visibility of loopbacks in it *own* Level 1 area *only*

Note that, even though R1 doesn't have specific routes to the other area's loopbacks and interconnect routes, it can still reach them through the default route

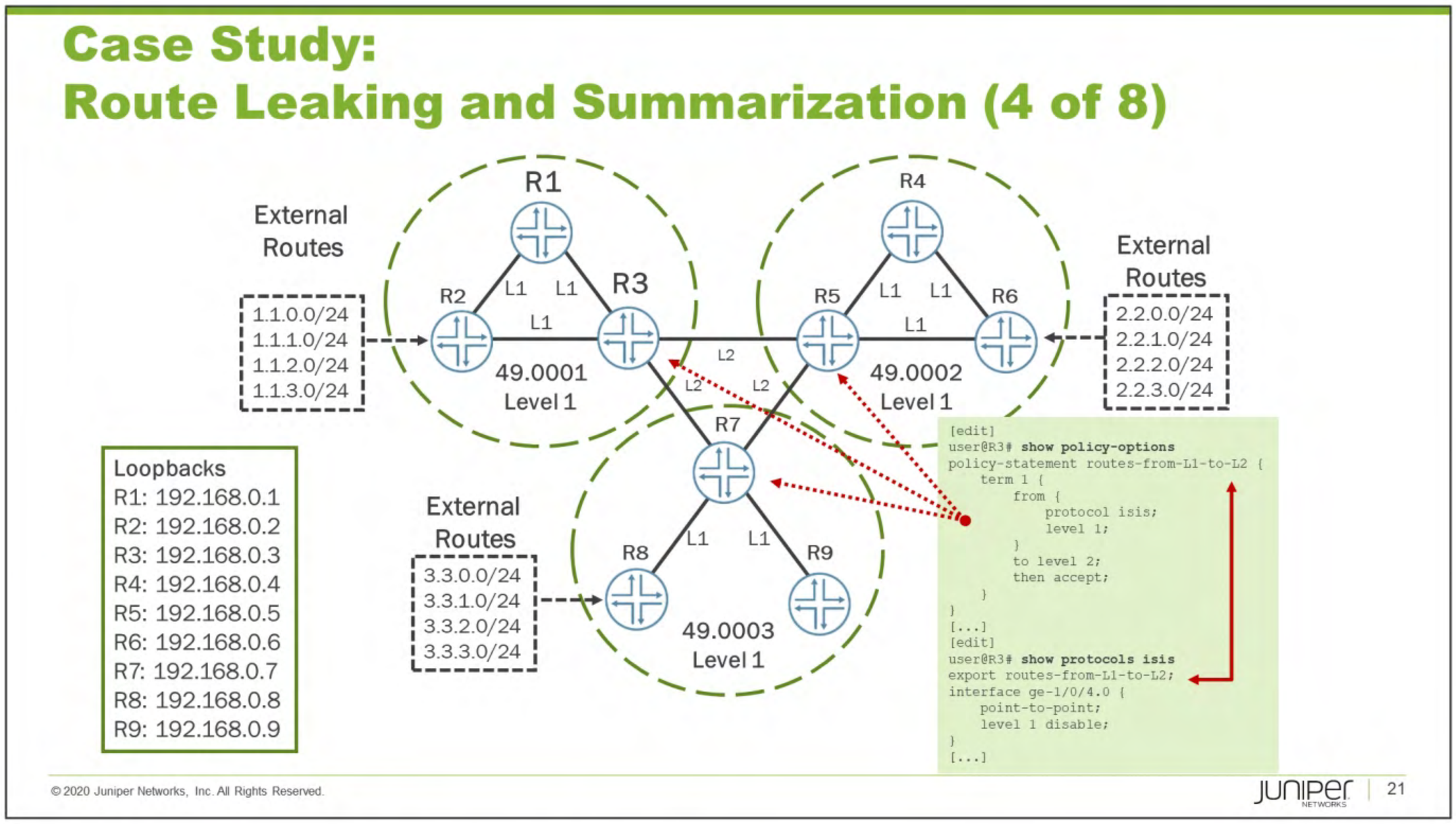
```
user@R1> show route terse table inet.0

inet.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A V Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* ? 0.0.0.0/0          I  15         10             >10.0.1.1
* ? 1.1.0.0/24         I 160         10             >10.0.1.6
* ? 1.1.1.0/24         I 160         10             >10.0.1.6
* ? 1.1.2.0/24         I 160         10             >10.0.1.6
* ? 1.1.3.0/24         I 160         10             >10.0.1.6
* ? 10.0.1.0/30        D   0                        >ge-1/1/7.0
* ? 10.0.1.2/32        L   0                         Local
* ? 10.0.1.4/30        D   0                        >ge-1/0/9.0
* ? 10.0.1.5/32        L   0                         Local
* ? 10.0.1.8/30        I  15         20              10.0.1.6
                                                    >10.0.1.1
* ? 192.168.0.1/32     D   0                        >lo0.1
* ? 192.168.0.2/32     I  15         10             >10.0.1.6
* ? 192.168.0.3/32     I  15         10             >10.0.1.1

user@R1> show route 192.168.0.9

inet.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[IS-IS/15] 03:49:23, metric 10
                    > to 10.0.1.1 via ge-1/1/7.0

user@R1> ping count 5 rapid 192.168.0.9
PING 192.168.0.9 (192.168.0.9): 56 data bytes
!!!!!
--- 192.168.0.9 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.516/3.532/15.504/5.986 ms
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Case Study: Route Leaking and Summarization—Part 3

This slide shows sample output from the viewpoint of an arbitrary Level 1 router—the R1 router, in this case. The R1 router has specific routes to only the loopbacks, interconnects, and external routes in its area. Compared to the R3 router, the R1 router appears to have limited visibility. However, this isn't the case as it can reach the other area loopbacks and interconnect routes through the default route installed because of the attached bit it receives from the R3 Level 2 router. This functionality does not apply to the external routes from the other areas, though. Outputs from the other Level 1 routers in the network would be similar. You can see the attached bit attribute in the following output.

```
user@R1> show isis database
IS-IS level 1 link-state database:
LSP ID                    Sequence Checksum Lifetime Attributes
mxA-1-R1.00-00                0x1c   0xb0ec     1191 L1 L2
mxA-1-R2.00-00                0x21   0x5342      644 L1 L2
mxA-1-R3.00-00                0x1b    0x57b      388 L1 L2 Attached
  3 LSPs

IS-IS level 2 link-state database:
LSP ID                    Sequence Checksum Lifetime Attributes
mxA-1-R1.00-00                0x1d   0xc117      940 L1 L2
  1 LSPs
```



Case Study: Route Leaking and Summarization—Part 4

Now that we've established a default IS-IS behavior baseline, let's leak some routes from all Level 1 areas to our Level 2 area. As mentioned previously, route leaking is accomplished, in IS-IS, through the use of a policy. As shown in the slide, the same policy is created and applied to each of the Level 2 routers (R3, R5, and R7).

Case Study: Route Leaking and Summarization (5 of 8)

- R3 router view

Full visibility of all external routes in the network

```
user@R3> show route terse table inet.0

inet.0: 36 destinations, 36 routes (36 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A V Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* ? 1.1.0.0/24         I 160         10            >10.0.1.10
* ? 1.1.1.0/24         I 160         10            >10.0.1.10
* ? 1.1.2.0/24         I 160         10            >10.0.1.10
* ? 1.1.3.0/24         I 160         10            >10.0.1.10
* ? 2.2.0.0/24         I 165         20            >10.0.10.2
* ? 2.2.1.0/24         I 165         20            >10.0.10.2
* ? 2.2.2.0/24         I 165         20            >10.0.10.2
* ? 2.2.3.0/24         I 165         20            >10.0.10.2
* ? 3.3.0.0/24         I 165         20            >10.0.10.6
* ? 3.3.1.0/24         I 165         20            >10.0.10.6
* ? 3.3.2.0/24         I 165         20            >10.0.10.6
* ? 3.3.3.0/24         I 165         20            >10.0.10.6
[...]
```

- R1 router view

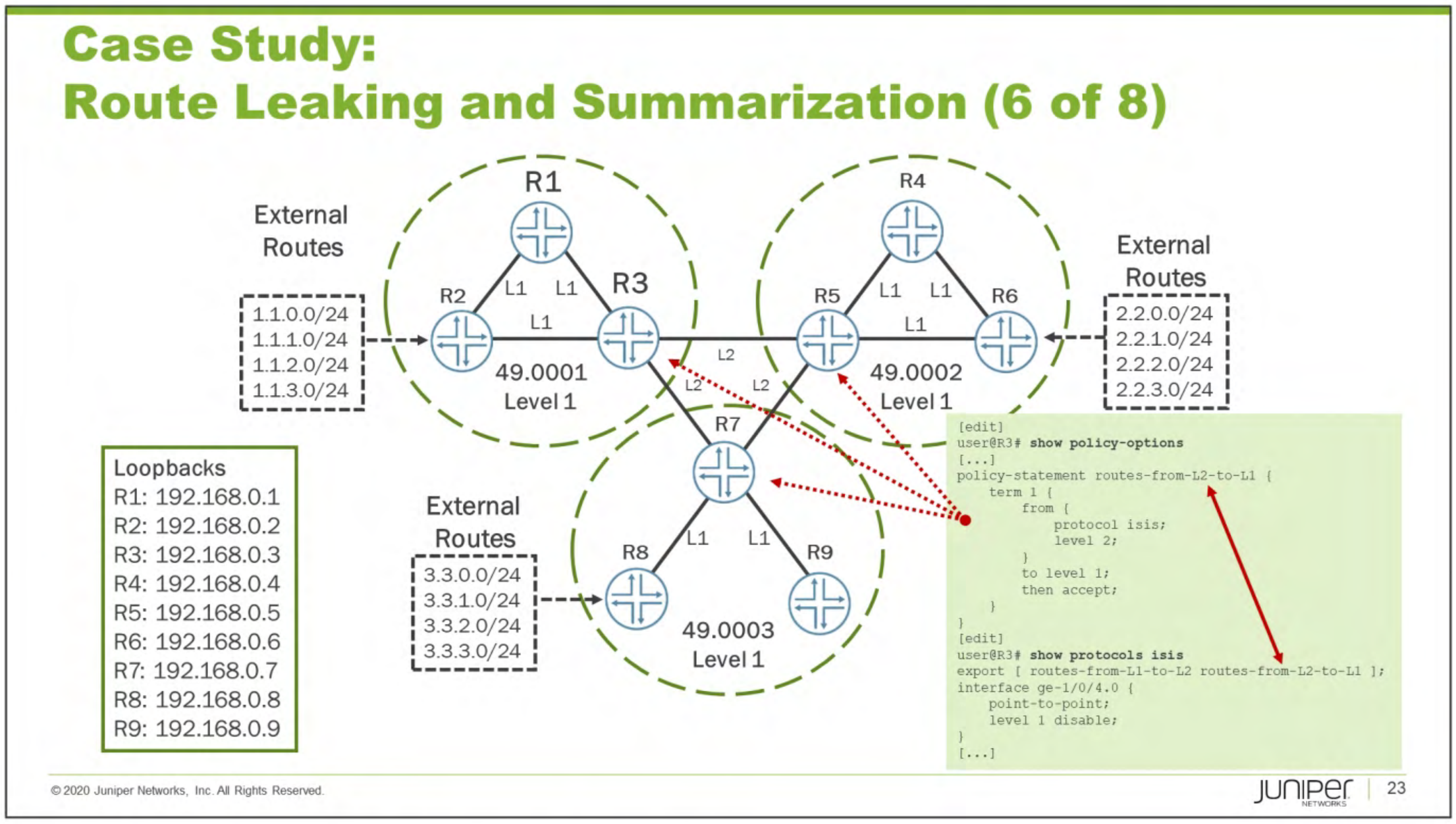
R1 routing table remains unchanged

```
user@R1> show route terse table inet.0

inet.0: 13 destinations, 13 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A V Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* ? 0.0.0.0/0          I  15         10            >10.0.1.1
* ? 1.1.0.0/24         I 160         10            >10.0.1.6
* ? 1.1.1.0/24         I 160         10            >10.0.1.6
* ? 1.1.2.0/24         I 160         10            >10.0.1.6
* ? 1.1.3.0/24         I 160         10            >10.0.1.6
* ? 10.0.1.0/30        D   0                       >ge-1/1/7.0
* ? 10.0.1.2/32        L   0                        Local
* ? 10.0.1.4/30        D   0                       >ge-1/0/9.0
* ? 10.0.1.5/32        L   0                        Local
* ? 10.0.1.8/30        I  15         20             10.0.1.6
                                                   >10.0.1.1
[...]
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

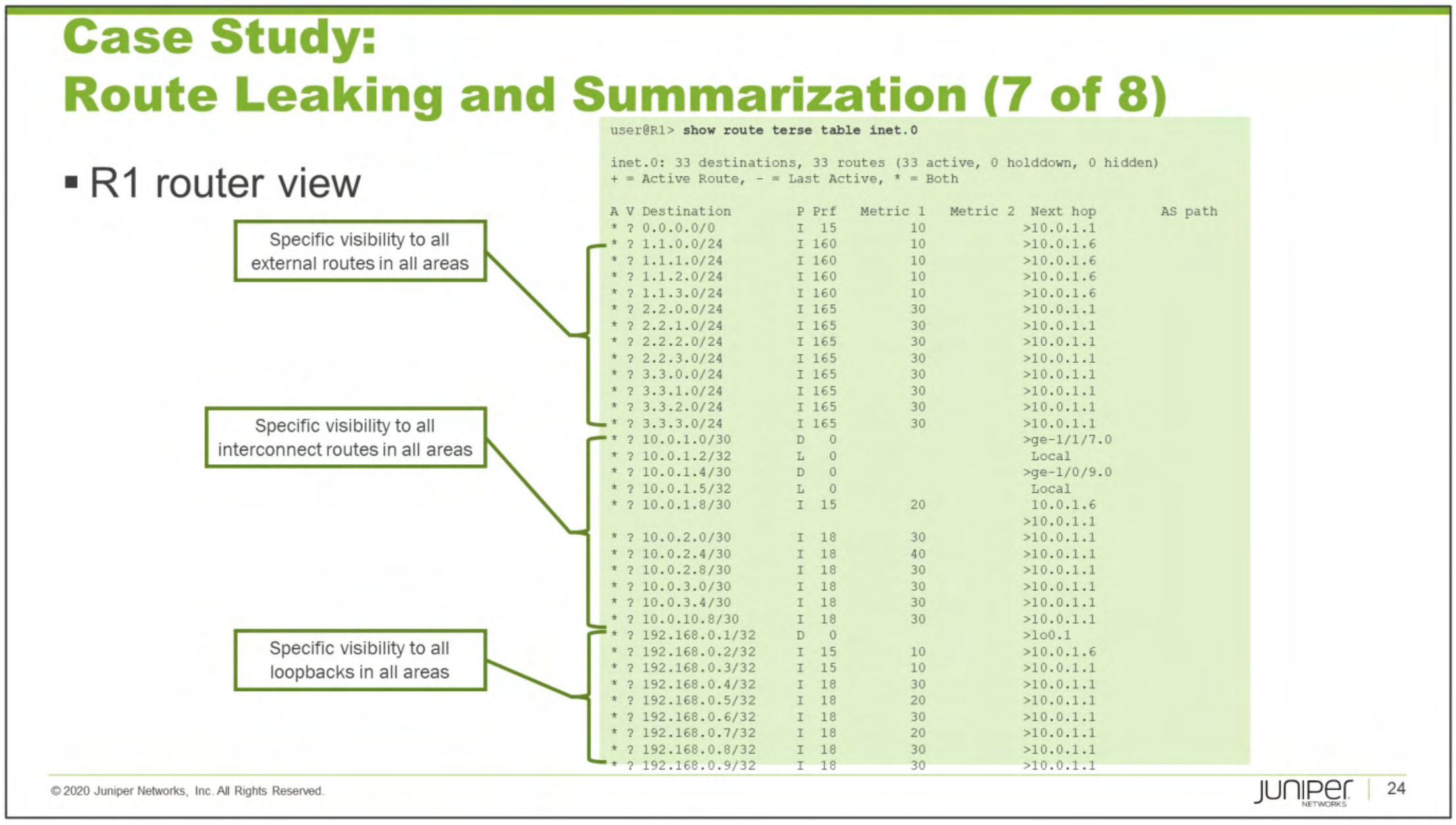
Case Study: Route Leaking and Summarization—Part 5

The slide shows the results of our changes. The R3 router now has specific routes for all the external routes in the Level 1 areas. Routers R5 and R7 would see similar changes. All Level 1 routers' routing tables remain unchanged. We show a sample output from R1 on slide.



Case Study: Route Leaking and Summarization—Part 6

To leak routes from Level 2 to Level 1, the same type of policy is used. Of course, the levels are switched in this policy. Note that you can export multiple policies under IS-IS but, as an alternative, you could write a single policy with multiple terms. As before, the policy is applied on the R3, R5, and R7 Level 2 routers.



Case Study: Route Leaking and Summarization—Part 7

The slides shows the changes to the R1 router which now has specific routes to all external routes, loopbacks, and interconnects. That is, there is no more reliance on the 0/0 default route as shown in the following output.

```
user@R1> show route 192.168.0.9
inet.0: 33 destinations, 33 routes (33 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.0.9/32      *[IS-IS/18] 00:30:29, metric 30
                     > to 10.0.1.1 via ge-1/1/7.0
```

You could use the **ignore-attached-bit** option on R1 (or any Level 1 router) to remove the default route from the table.

```
[edit]
user@R1# set protocols isis ignore-attached-bit

[edit]
user@R1# commit
commit complete

[edit]
user@R1# run show route 0/0 exact

[edit]
user@R1#
```

Case Study: Route Leaking and Summarization (8 of 8)

- Three-step process
  - Create aggregate route
  - Create matching policy
  - Export policy into IS-IS

```
[edit]
user@R3# show routing-options
aggregate {
    route 1.1.0.0/22;
}

[edit]
user@R3# show policy-options
policy-statement agg-route {
    term match-agg-route {
        from {
            protocol aggregate;
            route-filter 1.1.0.0/22 exact;
        }
        to level 2;
        then accept;
    }
    term suppress-specifics {
        from {
            route-filter 1.1.0.0/22 longer;
        }
        to level 2;
        then reject;
    }
}
[...]

[edit]
user@R3# show protocols isis
export [ agg-route routes-from-L1-to-L2 routes-from-L2-to-L1 ];
[...]
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Case Study: Route Leaking and Summarization—Part 8

Unlike OSPF, there is no **area-range** statement to summarize routes in IS-IS. Summarizing routes in IS-IS requires a three step process: creating an aggregate route, creating a matching policy, and then exporting that policy into IS-IS. The slide example shows how to do this from the R3 router. Similar policies would be created and applied on R5 and R7. The following output shows an abbreviated view of R3's routing table. Note that the /24 addresses from Area 49.0002 and Area 49.0003 are gone and only the /22 summary route is visible. As before, outputs on R5 and R7 would be similar.

```
user@R3> show route terse table inet.0

inet.0: 31 destinations, 31 routes (31 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A V Destination          P Prf   Metric 1   Metric 2  Next hop        AS path
* ? 1.1.0.0/22           A 130                        Reject
* ? 1.1.0.0/24           I 160        10              >10.0.1.10
* ? 1.1.1.0/24           I 160        10              >10.0.1.10
* ? 1.1.2.0/24           I 160        10              >10.0.1.10
* ? 1.1.3.0/24           I 160        10              >10.0.1.10
* ? 2.2.0.0/22           I 165        20              >10.0.10.2
* ? 3.3.0.0/22           I 165        20              >10.0.10.6
[...]
```

IS-IS Best Practices (1 of 2)

- Enable wide metrics

```
[edit]
user@router# set protocols isis level 2 wide-metrics-only
```

- Increase the LSP lifetime, from 1200 seconds (default), to reduce the amount of control traffic generated

```
[edit]
user@router# set protocols isis lsp-lifetime 4000
```

- Adjust how quickly IS-IS performs an SPF calculation after detecting a topology change (200 ms default)

```
[edit]
user@router# set protocols isis spf-options delay 50
```

- Use the **overload timeout** *value* option to prevent traffic from transiting a newly booted router

```
[edit]
user@router# set protocols isis overload timeout 600
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

IS-IS Best Practices—Part 1

This slide lists some examples of best practices to use in regard to IS-IS. These examples include the following:

- Enable wide metrics using the **wide-metrics-only** option. Using the default narrow metrics, you can only specify metric values between 1-63. This is not very scalable in today's service provider networks. Enable wide metrics allows you to specify metric values between 1-16777215.
- You can reduce the amount of control traffic generated by IS-IS using the **lsp-lifetime** option.
- Adjust how quickly IS-IS performs an SPF calculation after detecting a topology change using the **spf-options delay** command. Most routers nowadays have plenty of CPU resources and can easily handle a lower setting than the default value.
- Use the **overload timeout** option to prevent transit traffic through a newly booted router, giving it time to get all protocols up and running. Keep in mind that this option also takes effect if routing is restarted on a currently running router.

IS-IS Best Practices (2 of 2)

- Use the **ignore-attached-bit** option to avoid certain cases of sub-optimal routing

```
[edit]
user@router# set protocols isis ignore-attached-bit
```

- Enable BFD on interfaces to reduce failure detection times

```
[edit protocols isis]
user@router# show
interface ge-1/1/1.0 {
    bfd-liveness-detection {
        minimum-interval 30;
        multiplier 3;
    }
```

- Use authentication

```
[edit protocols isis]
user@router# show interface ge-1/1/2
level 2 {
    hello-authentication-key "$9$km5FCtOcyKn/yKM8dVqmf"; ## SECRET-DATA
    hello-authentication-type md5;}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

IS-IS Best Practices—Part 2

This slide lists some further examples of best practices to use in regard to IS-IS. These additional examples include the following:

- Use the **ignore-attached-bit** option to avoid certain cases of suboptimal routing where the default route selected through the attached bit behavior is not the one you want to use. Also, there may be times, such as during a denial of service (DoS) attack, that you do not want a Level 1 router to be able to forward traffic based on a default route.
- Enabling Bidirectional Forwarding Detection (BFD) between interfaces can drastically reduce failure detection times in comparison to the default protocol timers. In BFD, hello packets are sent at a specified, regular interval. A neighbor failure is detected when the router stops receiving a reply after a specified interval. Keep in mind that BFD can result in churn, so these timers are also adaptive and can be adjusted to be more or less aggressive. For example, the timers can adapt to a higher value if the adjacency fails, or a neighbor can negotiate a higher value for a timer than the one configured. Note that BFD works with a wide variety of network environments and topologies, not just IS-IS.
- Even though it might go without saying, use authentication.

Summary

- In this content, we:
  - Examined the default operation of a multilevel IS-IS network
  - Described address summarization methods used in IS-IS
  - Configured and monitored a multilevel IS-IS network

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- The default operation of a multilevel IS-IS network;
- IS-IS summarization methods; and
- The configuration and monitoring of a multilevel IS-IS network.

Review Questions

1. How are areas segmented in an IS-IS L1/L2 network?
2. An IS-IS L1/L2 network is similar to which type of OSPF environment?
3. Where is IS-IS route leaking performed? What steps are needed to accomplish this leaking?
4. Where is IS-IS route summarization performed? How is this route summarization accomplished? What types of routes can be summarized?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

3.

4.

Lab: Configuring a Multilevel IS-IS Network

- Configure, monitor, and troubleshoot the operation of a multilevel IS-IS network.
- Once operational, configure route leaking and address summarization.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: Configuring a Multilevel IS-IS Network

The slide provides the objectives for this lab.

Answers to Review Questions

1.

Areas are segmented in an IS-IS L1/L2 network using L1/L2 ABRs.

2.

An IS-IS L1/L2 network is similar to an OSPF NSSA with the **`no-summaries`** and **`default-metric`** options configured.

3.

Route leaking is performed on the L1/L2 ABR. A routing policy is written matching the Level 2 routes to be leaked. This policy is then applied at the `[edit protocols isis]` hierarchy.

4.

Summarization is performed on the L1/L2 ABR. An aggregate route encompassing the desired more specific routes must be defined. Then a routing policy is created matching the aggregate with the **`to level 2`** and **`then accept`** actions. The policy should include a term to reject more specific routes. The policy is applied at the `[edit protocols isis]` hierarchy.

Education Services

Advanced Junos Service Provider Routing

Chapter 9: Troubleshooting IS-IS

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - List useful commands to troubleshoot and verify IS-IS problems
  - Troubleshoot and isolate different IS-IS issues

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- Useful commands that can be used to troubleshoot and verify IS-IS problems; and
- Troubleshooting and isolating different IS-IS issues.

Agenda: IS-IS Troubleshooting

→IS-IS Troubleshooting Overview

- Case Study

© 2020 Juniper Networks, Inc. All Rights Reserved.

IS-IS Overview

The slide lists the topics we will discuss. We discuss the highlighted topic first.

IS-IS Troubleshooting Checklist (1 of 2)

- Adjacency
  - Family ISO missing
  - Level 1 area mismatch
  - Level 1 versus Level 2 interface
  - IP subnet mismatch
  - Authentication parameters mismatch
  - Minimal MTU of 1492
  - MTU must match on both sides
  - Passive link is configured
  - Point-to-point versus Broadcast interface
- LSDB integrity
  - Unique NET must be configured at every router

© 2020 Juniper Networks, Inc. All Rights Reserved.

IS-IS Adjacency Troubleshooting

IS-IS troubleshooting generally falls into one of two categories; adjacency troubleshooting and routing troubleshooting. Recall that IS-IS uses family ISO packets for its PDUs. If `family ISO` is not configured on the router interfaces that are to carry IS-IS PDUs, the router will not be able to establish IS-IS adjacencies.

For Level 1 adjacencies to form, the routers must be in the same area. If adjacencies fail to form between Level 1 routers, check the area portion of the NET address. A Level 1 router will never form an adjacency with a Level 2 router.

IS-IS will not establish an adjacency if the IP subnets configured on each side are unable to forward traffic to one another. While there is no requirement for IS-IS to have network masks that match exactly, as long as the subnets are overlapping, an adjacency can be formed. This is part of the sanity checks the routers use to ensure that they can forward IP packets across the link.

At the beginning of the adjacency formation, IS-IS hello packets are padded to the size of 1492 bytes to ensure that the link can handle the largest PDU. If a neighboring interface has an MTU lower than 1492 bytes an adjacency will not form.

Mismatched authentication parameters is another possible cause of a malfunctioning IS-IS adjacency.

Marking an interface as passive allows IS-IS to advertise the IP subnet of a specific link as internal prefixes, however this setting prevents IS-IS hellos from being transmitted ensuring an adjacency with another IS-IS route will not occur.

When an IS-IS Ethernet link only has two routers, a more efficient connection can be formed by configuring the interface types as point-to-point. If the interface types do not match on both sides, the adjacency will not form.

LSDB Integrity

An IS-IS issue that is hard to troubleshoot is having the same System Id configured on two different routers. The System Id plays the same role in IS-IS as the Router ID does in OSPF. The System Id is part of the NET address configured on the router's loopback. If two routers have the same System Id and are directly connected, they will not form an adjacency. If two routers that have the same System Id are not physically connected, adjacencies will form but it can create a more difficult problem to detect and resolve.

This type of issue can be detected by monitoring the LSDB. If you discover that there are regular routing flaps, this could be the result of routers with the same System Id advertising different pieces of routing information. Every router will see the updates as coming from the same device. Each update will result in another LSDB churn.

Another indicator of this issue would be continuous SPF runs which can be determined by viewing the SPF log.

IS-IS Troubleshooting Checklist (2 of 2)

- Routing
  - Metrics
    - Wide or narrow
  - lo0.0 interface
  - Route summarization
    - External versus internal
  - Route leaking
  - Export policy
    - Watch out for multiple redistribution points
    - Prefix export limit

© 2020 Juniper Networks, Inc. All Rights Reserved.

IS-IS Routing Troubleshooting

Troubleshooting IS-IS routing is more of an optimization task than troubleshooting a complete failure. For example, routers in your network could follow suboptimal routing from the network design perspective. To optimize traffic flows for certain destinations, metrics can be manipulated. By default, IS-IS uses a metric of 10 for all links irrespective of the interface bandwidth, except for the loopback interface, which has a metric of 0. Using consistent metric values provides a way to control traffic flows.

If **`wide-metrics-only`** is configured, it should be implemented on all routers in the level. If it is configured on some routers while other routers are using narrow metrics, the routers using narrow metrics will consider any wide metrics greater than 63 equal to the maximum value of 63.

Ensuring loopback reachability within IS-IS is useful for testing device reachability which aids in troubleshooting. Failing to configure the lo0 interface in IS-IS results in the loopback address missing from the IS-IS routing domain.

Route summarization at the L1/L2 border is an optimization task aimed at reducing the LSDB size of the Level 2 routers. It is also a common place for routing issues. IS-IS, unlike OSPF, does not have built-in configuration commands to provide summarization. A routing policy is used in Junos to provide this functionality.

When using wide metrics, the default forwarding rules of IS-IS is changed. This results in the automatic leaking of external IS-IS routes from Level 1 into Level 2. Any policy aimed at summarizing routes from Level 1 to Level 2 will need to include a `to level` statement and include a `reject` action for the more specific internal and external routes.

Continued on the next page.

IS-IS Routing Troubleshooting (contd.)

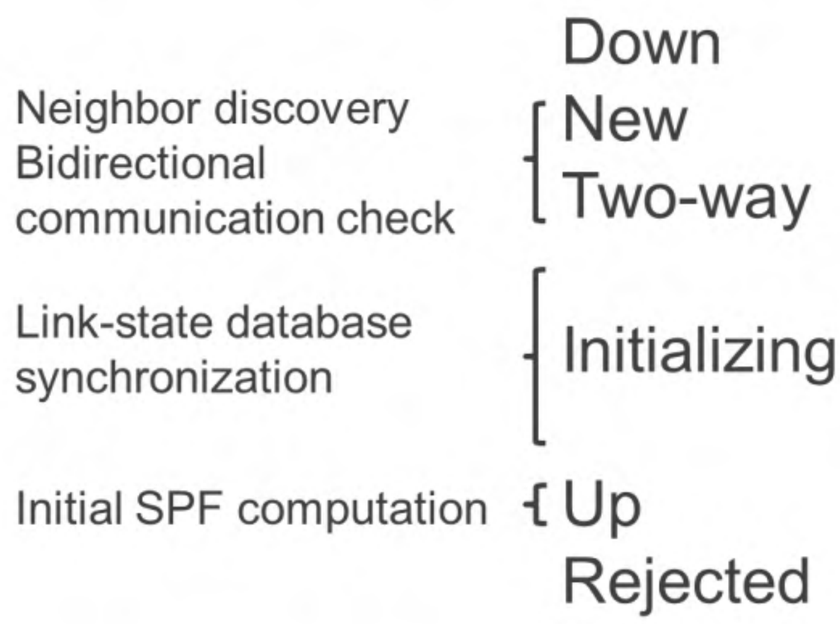
The following is an example of a policy that can be used to summarize either internal or external IS-IS routes at a L1/L2 border router:

```
[edit policy-options policy-statement L1-summary-route]
user@router# show
term local-summary-route {
    from {
        protocol aggregate;
        route-filter 10.0.4.0/22 exact;
    }
    to level 2;
    then accept;
}
term suppress-specifics {
    from {
        route-filter 10.0.4.0/22 longer;
    }
    to level 2;
    then reject;
}
```

Be careful when configuring mutual redistribution between IS-IS and other IGPs using multiple border routers. It can lead to the formation of a routing loop at worst or suboptimal routing. The general rule of thumb is, when redistribution is performed from a protocol with a lower (better) preference to a protocol with a higher (worse) preference, loops are generally not an issue. If redistribution is performed from a protocol with a higher preference to a protocol with a lower preference, be careful as routing loops are much more likely.

IS-IS Adjacencies

- The IS-IS adjacency state machine



© 2020 Juniper Networks, Inc. All Rights Reserved.

IS-IS Adjacency States

The slide shows the state machine associated with IS-IS adjacencies.

A transition to the `New` state is triggered whenever a valid hello packet from a neighbor is received on an interface; the state moves then to `two-way` once the communication is confirmed to be bidirectional.

After this check, the machine moves into the `Initializing` state—where the link-state databases of both nodes are synchronized. Finally, the adjacency moves into the `Up` state.

The `rejected` state is only used in cases where the adjacency is refused; it is an error state.

Useful IS-IS Operational Commands (1 of 8)

- You want to check that IS-IS is operational

```
user@router> show isis interface
IS-IS interface database:
Interface             L CirID Level 1 DR        Level 2 DR        L1/L2 Metric
ae0.0                 1   0x1 R2.02             Disabled               10/10
ge-0/0/4.301          1   0x3 R1.03             Disabled               10/10
ge-0/0/6.0            2   0x2 Disabled          R1.02                  10/10
lo0.0                 0   0x1 Passive           Disabled                0/0


user@router> show isis statistics
IS-IS statistics for R1:
PDU type       Received    Processed        Drops         Sent       Rexmit
LSP                  10           10            0           14            0
IIH                 426           13            0          587            0
CSNP                 86           86            0          103            0
PSNP                  1            1            0            0            0
Unknown               0            0            0            0            0
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Checking IS-IS Operational Status

When troubleshooting IS-IS, there are a number of useful CLI commands that can used to find the root cause of the problem.

Use the **`show isis interface`** command to display the IS-IS parameters associated with an interface. This command displays the following:

- Interface that is running IS-IS;
- Level of the interface—Level 1 (1), Level 2 (2), or 3 for both;
- The Level 1 and 2 designated intermediate system (DIS); and
- The metrics for the L1/L2 interfaces.

This command is useful to verify the IS-IS configuration on the router.

The **`show isis statistics`** command displays the counters for various IS-IS PDUs sent and received. The Drops field can be useful in viewing which IS-IS packets are being dropped. This can help you focus on where to look for adjacency issues.

Useful IS-IS Operational Commands (2 of 8)

- You want to check IS-IS summary information

```
user@router> show isis overview
Instance: master
  Router ID: 10.222.1.2
  Adjacency holddown: enabled
  Maximum Areas: 3
  LSP life time: 1200
  Attached bit evaluation: enabled
  SPF delay: 200 msec, SPF holddown: 5000 msec, SPF rapid runs: 3
  IPv4 is enabled, IPv6 is enabled
  Traffic engineering: enabled
---(more)---
```

- You want to check IS-IS adjacency status

```
user@router> show isis adjacency
Interface             System         L State        Hold (secs) SNPA
ae0.0                 R2               1  Up                   7  b0:c6:9a:73:3a:0
ge-0/0/4.301          R3               1  Up                  25  0:22:83:9c:21:81
ge-0/0/6.0            R4               2  Up                  26  0:19:e2:55:24:89
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Displaying IS-IS Summary Information

The **show isis overview** command displays the IS-IS summary information.

Use the **show isis adjacency** command to display the status of IS-IS adjacencies. The command output lists the local router interfaces, neighboring routers by their host names, level, state of adjacency, and remaining hold time.

Sometimes it is useful to restart an IS-IS adjacency. You can clear an adjacency with the following command:

```
user@router> clear isis adjacency neighbor-name
```

Note that thisuser@router> **show isis hostname**

```
IS-IS hostname database:
System Id          Hostname                        Type
1921.6800.4201     R1                              Dynamic
1921.6800.4202     R2                              Dynamic
1921.6800.4203     R3                              Dynamic
```

Useful IS-IS Operational Commands (3 of 8)

- You want to check that LSDB is consistent

```
user@router> show isis database
IS-IS level 1 link-state database:
LSP ID                      Sequence Checksum Lifetime Attributes
R3.00-00                         0x5   0x43c2      479 L1 L2
R1.00-00                         0x4   0x55ab     1130 L1 L2 Attached
R1.03-00                         0x1   0x7b23      455 L1 L2R
R2.00-00                         0x3   0xaf2d      479 L1 L2 Attached
R2.02-00                         0x2   0xeeaa      838 L1 L2
R2.04-00                         0x1   0x9cfd      479 L1 L2
  6 LSPs

IS-IS level 2 link-state database:
LSP ID                      Sequence Checksum Lifetime Attributes
R1.00-00                         0x4   0xdf14      460 L1 L2
R1.02-00                         0x2   0x5345     1014 L1 L2
R2.00-00                         0x4   0x2f7e     1041 L1 L2
  3 LSPs
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Viewing the IS-IS Database

The **`show isis database`** command displays a brief view of all the LSPs in the IS-IS LSDB. The output shows separate LSDB entries for Level 1 and Level 2 databases. This view is useful when attempting to discover missing routers.

The seven bytes that make up the IS-IS LSP ID is encoded in the following format: System Id. Pseudo node ID-LSP number. If the Pseudo node ID is set to 00 value this indicates that the LSP was generated by the router for itself. Any other value indicates that the LSP was generated by the DIS on behalf of the pseudo node it represents on a broadcast segment.

Useful IS-IS Operational Commands (4 of 8)

- You want to check the LSDB details

```
user@router> show isis database extensive | find TLVs
  TLVs:
    Area address: 49.0001 (3)
    LSP Buffer Size: 1492
    Speaks: IP
    Speaks: IPV6
    IP router id: 10.222.1.2
    IP address: 10.222.1.2
    Hostname: R1
    IP prefix: 10.222.0.0/30, Internal, Metric: default 10, Up
    IP prefix: 10.222.0.8/30, Internal, Metric: default 10, Up
    IP prefix: 10.222.1.2/32, Internal, Metric: default 0, Up
    IP extended prefix: 10.222.0.0/30 metric 10 up
    IP extended prefix: 10.222.0.8/30 metric 10 up
    IP extended prefix: 10.222.1.2/32 metric 0 up
    IPv6 prefix: fdcd:aaaa:bbbb:0:1::2/128 Metric 0 Up
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Displaying Detailed IS-IS Database Information

The **`show isis database extensive`** command provides detailed output for the contents of the IS-IS LSDB. This command is important in IS-IS troubleshooting because it allows you to view detailed information about the router's neighbors, the advertised IP subnets, and other parameters.

The example on the slide shows a number of TLVs which make up the LSPDUs received from an IS-IS peer and is stored in the LSDB. This information can be used to view information about the advertising router, such as the router is in area 49.0001 and that it supports both IP and IPv6 routing. The loopback address is 10.222.1.2 and its host name is R1. Also note that the router is advertising various internal IP subnets in the 10.222/16 range and an IPv6 route of fdcd:aaaa:bbbb:0:1::2/128.

Useful IS-IS Operational Commands (5 of 8)

- For example, find out if a router is advertising a prefix
  - Use the **match** operator for a fast answer

```
user@router> show isis database level 1 R1.00 extensive | match 172.22.138.16/30
   IP prefix: 172.22.138.16/30                Metric:       10 Internal Up
    IP extended prefix: 172.22.138.16/30 metric 10 up

user@router> show isis database level 2 R1.00 extensive | match 192.168.17.0/24
   IP prefix: 192.168.17.0/24                 Metric:        0 Internal Up
    IP extended prefix: 192.168.17.0/24 metric 0 up
```

  - Quick shortcut to check if local or external routes are redistributed into a level

© 2020 Juniper Networks, Inc. All Rights Reserved.

Verify Advertised Prefixes

Even though link-state PDUs can be large and complex objects, using CLI operators like **find** or **match** allows you to control the command output and quickly gather information such as if routes are being advertised into a level.

Useful IS-IS Operational Commands (6 of 8)

- You want to check SPF events

```
user@router> show isis spf log
 IS-IS level 1 SPF log:
Start time          Elapsed (secs) Count Reason
Thu Mar 14 23:55:36        0.000035     3 Reconfig
Thu Mar 14 23:55:37        0.000093     1 Updated LSP R1.00-00
Thu Mar 14 23:57:27        0.000128     1 Topologies changed for adjacency R2 on ae0.0
Thu Mar 14 23:57:42        0.000425     5 New adjacency R2 on ae0.0
---(more)---
```

- You want to check SPF results

```
user@router> show isis route
 IS-IS routing table            Current version: L1: 8 L2: 8
IPv4/IPv6 Routes
----------------
Prefix             L Version   Metric Type Interface       NH   Via
10.222.0.16/30     1       8       20 int  ae0.0           IPV4 R2
                                           ge-0/0/4.301    IPV4 R3
10.222.0.20/30     2       8       20 int  ge-0/0/6.0      IPV4 R4
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Viewing the SPF Log

The **`show isis spf log`** command displays the entries in the log of SPF calculations. Use this command to verify that you have a stable IS-IS network. Multiple recalculations in a short period of time indicate potential instability. Don't forget that many recalculations in a short period of time may indicate duplicate System Ids exist in the IS-IS network.

Displaying IS-IS Routes

The **`show isis route`** command displays the results of the SPF calculations that are passed to the routing table.

Useful IS-IS Operational Commands (7 of 8)

- You want to check IS-IS routes in the routing table

```
user@router> show route protocol isis

inet.0: 33 destinations, 38 routes (33 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.222.0.16/30       [IS-IS/15] 00:22:50, metric 20
                       to 10.222.0.2 via ae0.0
                     > to 10.222.0.10 via ge-0/0/4.301
10.222.0.20/30       [IS-IS/18] 00:27:41, metric 20
                     > to 10.222.0.6 via ge-0/0/6.0
10.222.1.1/32        [IS-IS/15] 00:22:50, metric 10
                     > to 10.222.0.10 via ge-0/0/4.301
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Displaying IS-IS Routes in the Routing Table

Use the **`show route protocol isis`** command to display routes in the routing table that were installed by IS-IS.

Useful IS-IS Operational Commands (8 of 8)

- For difficult problems you can perform a packet capture:
  - `monitor traffic interface` also known as a tcpdump

```
user@router> monitor traffic interface ae0.0 no-resolve detail
Address resolution is OFF.
Listening on ae0.0, capture size 1514 bytes

00:33:52.060730  In IS-IS, length 131
       L1 CSNP, hlen: 33, v: 1, pdu-v: 1, sys-id-len: 6 (0), max-area: 3 (0)
         source-id:    0102.2200.1003.00, PDU length: 131
         start lsp-id: 0000.0000.0000.00-00
         end lsp-id:   ffff.ffff.ffff.ff-ff
           LSP entries TLV #9, length: 96
             lsp-id: 0102.2200.1001.00-00, seq: 0x00000006, lifetime:   356s, chksum: 0x41c3
             lsp-id: 0102.2200.1002.00-00, seq: 0x00000007, lifetime:  1153s, chksum: 0xd429
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Debugging IS-IS: Monitor Traffic

To view the protocol messages received on or sent from a router interface, use the **`monitor traffic interface <interface-name>`** command. This command is essentially a front end to the well known tcpdump utility.

The example on the slide shows an IS-IS Level 1 CSNP packet received on ae0.0 interface. The detail option captures the full 1514 byte frame.

IS-IS Adjacencies: Traceoptions (1 of 5)

- Useful traceoptions when troubleshooting IS-IS adjacencies

```
[edit protocol isis]
user@router# show
traceoptions {
    file isis size 10m files 2;
    flag error detail;
    flag hello send receive detail;
}
```

  - As for OSPF, start with **`error detail`**, then add **`hello send receive`**, if needed
  - The messages are sometimes less explicit than OSPF and require interpretation

© 2020 Juniper Networks, Inc. All Rights Reserved.

Using Traceoptions

If **`monitor traffic interface`** does not provide sufficient information for problem resolution, another option is to use traceoptions; the Junos debugging tool. When using traceoptions, the information that will be captured is controlled using the **`flag`** options. A good approach is to begin by using just the **`flag error`** option. If that does not provide enough information, you can try the **`flag hello send receive.`** The latter option is extremely useful in cases where there are sporadic peer-down events due to a neighbor timing out.

Having the **`flag hello`** enabled on both sides can show if the loss is unidirectional or bidirectional and allows for correlation with transmission or interface errors.

The messages found in a traceoptions file for IS-IS are a little more cryptic than those in OSPF and can require a bit more interpretation.

IS-IS Adjacencies: Traceoptions (2 of 5)

- Level 1 area mismatch

```
show log isis | match ERROR
Mar 12 11:09:54.259879 ERROR: IIH from R2 with no
matching areas, interface ge-1/0/4.0
```

- Can happen if Level 1 has been left enabled on a link to another area
- A Level 2 adjacency on the same link will come up

© 2020 Juniper Networks, Inc. All Rights Reserved.

Level 1 Area Mismatch

This error can be caused by a wrong ISO (NSAP) address under the loopback interface, causing a router to belong to a different area than its neighbor.

It is also possible to see this error if, by mistake, Level 1 has been left enabled on a link to another area. In this case, there is no service impact, as Level 2 will come up.

IS-IS Adjacencies: Traceoptions (3 of 5)

- Network mismatch

```
show log isis | match ERROR
```

```
Dec 6 10:42:46.769421 ERROR: IIH from R2 without
matching addresses, interface ge-1/0/4.0
```

- Indicates a wrong interface configuration: no overlapping subnet means the link cannot be used for transit
- Note: different from OSPF, adjacencies are possible even in case of different netmasks—as long as there is a common subnet that can be used for transit

© 2020 Juniper Networks, Inc. All Rights Reserved.

Network Mismatch

While there is no requirement for IS-IS to have netmasks that match exactly, if subnets are overlapping, an adjacency can be formed. For example, if R1 were to have an IP address of 192.168.1.1/30 and R2 had an IP address of 192.168.1.2/24, IS-IS could form an adjacency as the subnets overlap and an attempt to reach the peer is possible based on the address of each side.

However, if R1 were to have an IP address of 192.168.2.1/30 and R2 had an IP address of 192.168.1.2/24, the subnets do not overlap and no adjacency would be formed.

IS-IS Adjacencies: Traceoptions (4 of 5)

- MTU or interface type problem

```
Dec 6 10:37:50.283525 ISIS packet ignored: no matching interface
```

  - A catch-all error which can be due to several problems:
    - MTU mismatch
    - Interface type mismatch (one side only set to point-to-point)
  - The MTU mismatch causes these errors to be logged on one side only; the other side is stuck in `Initializing` state

```
user@router> show isis adjacency
Interface             System         L State        Hold (secs) SNPA
ge-1/0/4.0            R2             2 Initializing         26  5c:5e:ab:0:1f:7c
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

MTU or Interface Type Problem

This rather generic error message is logged whenever one side detects an MTU or interface type mismatch.

In case of MTU mismatch, the errors are logged only on the side with the bigger MTU. The side with smaller MTU will be stuck in the `Initializing` state.

The error message does not provide explicit information about the cause of the problem, so it is important to recognize this issue from its symptoms. One side is stuck in `Initializing` and the other ignoring hello packets due to `no matching interface`.

IS-IS Adjacencies: Traceoptions (5 of 5)

- No local system-id (that is, no ISO address)

```
show log isis | match ERROR
Dec 6 13:52:06.880845 ERROR: received IIH but have no local sysid
```

  - Causes all adjacencies to be down because the router detects its neighbors' hello packets, but cannot respond
  - Rather uncommon initial configuration issue

© 2020 Juniper Networks, Inc. All Rights Reserved.

No Local System-Id

The message highlighted on the slide indicates the router has not been assigned an ISO address. This is a very uncommon configuration mistake, as it is easy to spot because no adjacencies are formed.

Agenda: IS-IS Troubleshooting

- IS-IS Troubleshooting Overview

→Case Study

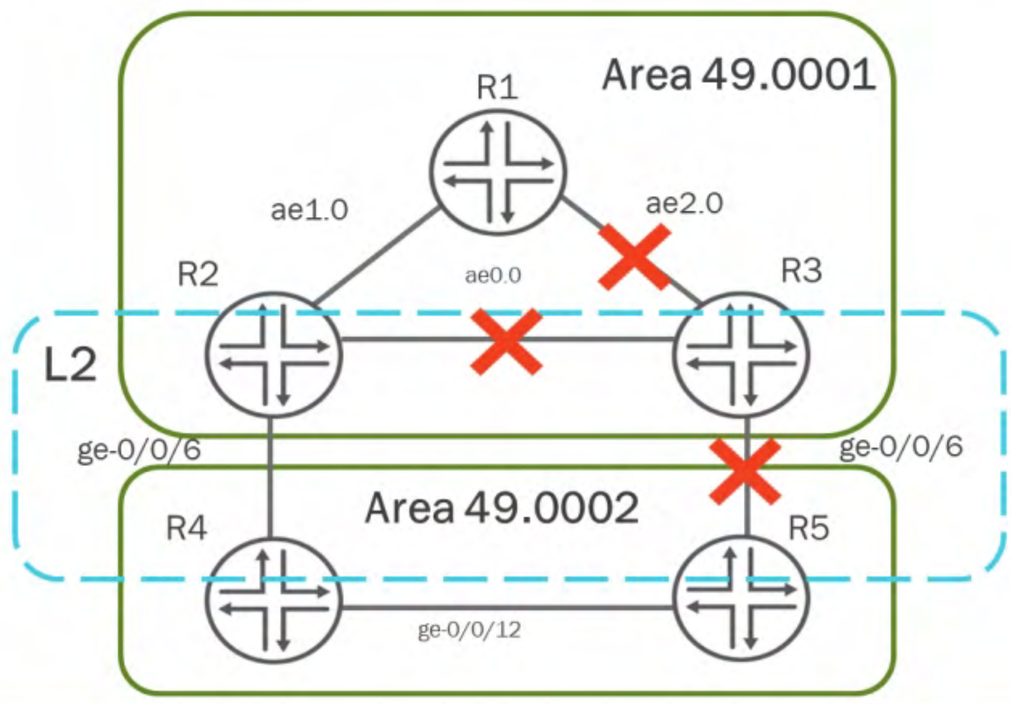
© 2020 Juniper Networks, Inc. All Rights Reserved.

Case Study

The slide highlights the topic we discuss next.

IS-IS Case Study: Adjacency Issues

- Topology, symptoms, and relevant information:
  - You recently put a new router (R3) into service. While you are verifying your network you found that R3 is not able to form adjacencies with neighbors. Physical connections have been verified and they are working fine.



© 2020 Juniper Networks, Inc. All Rights Reserved.

IS-IS Case Study: Adjacency Issues

The slide presents a network topology, reported symptoms, and description of a case related to IS-IS adjacency issues. In this case study you recently installed a new router (R3). You have determined that R3 has not established any IS-IS adjacencies. You have checked all physical connections and ensured that they are working fine.

Troubleshooting R3 Adjacencies (1 of 6)

- Check R3 adjacency state
  - No adjacencies

```
user@R3> show isis adjacency
```

- Check the IS-IS interfaces
  - L1 interfaces are area sensitive

```
user@R3> show isis interface
IS-IS interface database:
Interface             L CirID Level 1 DR        Level 2 DR        L1/L2 Metric
ae0.0                 1   0x1 R3.00             Disabled               10/10
ae2.0                 2   0x1 Point to Point    Disabled               10/10
ge-0/0/6.0            2   0x1 Disabled          Point to Point         10/10
lo0.0                 0   0x1 Passive           Disabled                0/0
```

- Check the R3 area information in L1 LSDB

```
user@R3> show isis database level 1 R3.00 extensive | match "Area address"
    Area address: 49.0002 (3)
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Check the IS-IS Adjacencies

When troubleshooting, the first command normally executed is the **`show isis adjacency`** command. The **`show isis adjacency`** command shows that R3 does not have any adjacencies.

Check R3 IS-IS Interfaces

By inspecting the local router's IS-IS interface settings using the **`show isis interface`** command, you can determine which levels the interfaces are attempting to form adjacencies. The output indicates R3 has interfaces in both Level 1 and Level 2.

Check R3 IS-IS Area

Because Level 1 adjacencies can only be established in the same IS-IS area, check the area settings on R3 and its neighbors. Using the command shown on the slide, you can determine the area address that is stored in the IS-IS database. This command reveals that R3 IS-IS area is 49.0002, which is not the local area defined. Another option to determine this information is to view the configuration file and show the NET address configuration on the router's lo0.0 interface.

Troubleshooting R3 Adjacencies (2 of 6)

- Fix the issue and check the R3 adjacencies again

```
user@R3> show isis adjacency
Interface             System         L State        Hold (secs) SNPA
ae2.0                 R1             1  Up                   20
```

- Check IS-IS statistics
  - The output shows Hello PDU drops

```
user@R3> show isis statistics
IS-IS statistics for R3:
PDU type        Received    Processed        Drops         Sent       Rexmit
LSP                  158          158            0          240            1
IIH                 6231           34          853        16642            0
CSNP                 153          153            0         4770            0
PSNP                  10            9            1           11            0
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Modify the Area Setting

Now fix the area configuration and check the IS-IS adjacencies again. The R3 has established an adjacency with R1. The other R3 interfaces are configured in Level 2 so the mismatched area cannot be a cause of the malfunctioning adjacencies.

Check the IS-IS Statistics

Use the **`show isis statistics`** command to determine if IS-IS PDUs are being sent and received successfully. The output of this command indicates that the IS-IS Hello PDU drop counter is incrementing.

Troubleshooting R3 Adjacencies (3 of 6)

- Enable traceoptions

```
[edit protocols isis]
user@R3# show
traceoptions {
    file isis.log;
    flag hello detail;
    flag error detail;
}
```

- Inspect the log file
  - Authentication failure

```
uer@R3> show log isis.log | find error
Mar 15 04:38:49.800297 ERROR: IIH authentication failure
Mar 15 04:38:49.800642 ERROR: IIH from 0102.2200.1005 on ge-0/0/6.0 failed authentication
```

  - Make sure that authentication settings match at R3 and R5

```
user@R3> show isis adjacency
Interface             System         L State        Hold (secs) SNPA
ae2.0                 R1             1  Up                   20
ge-0/0/6.0            R5             2  Up                   23
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Debugging the IS-IS Operations

To find the reason of the Hello drops, enable traceoptions to debug IS-IS.

Examine the Log

The log file examination reveals that IS-IS does not accept IS-IS Hello PDUs from its neighbor 0102.2200.1005 (R5) due to an authentication failure.

Adjust the authentication settings at both sides and verify the adjacencies again. This time, R3 shows two out of three adjacencies as up. The IS-IS neighbor on the ae0.0 interface is still missing.

Troubleshooting R3 Adjacencies (4 of 6)

- Check the IS-IS statistics again
  - There are still Hello drops

```
user@R3> clear isis statistics

user@R3> show isis statistics
IS-IS statistics for R3:
PDU type        Received    Processed        Drops         Sent       Rexmit
LSP                    0            0            0            0            0
IIH                    6            0            2            6            0
CSNP                   3            3            0            2            0
PSNP                   0            0            0            0            0
---(more)---
```

- Take a closer look at the trace log
  - It appears that R3 sends Hello but receives nothing from R2

```
user@R3> show log isis.log | match ae0.0
Mar 15 04:53:37.415805 ISIS L2 periodic xmit to 01:80:c2:00:00:14 interface ae0.0
Mar 15 04:53:46.084359 ISIS L2 periodic xmit to 01:80:c2:00:00:14 interface ae0.0
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Check the IS-IS Statistics

By monitoring the IS-IS statistics again, there are still Hello PDU drops occurring.

Examine the Log

Looking for ae0.0 entries in the traceoptions log file, you can see that R3 is not receiving Hello PDUs from R2.

Troubleshooting R3 Adjacencies (5 of 6)

- Enable traceoptions at R2 and check the trace log

```
user@R2> show log isis.log
Mar 15 05:05:53.413277 Sending PTP IIH on ae0.0
Mar 15 05:05:53.413405     max area 0, circuit type l1
Mar 15 05:05:53.413476     ptp adjacency tlv length 5
Mar 15 05:05:53.413527     neighbor state down
---(more)---
```

- Match it against the R3 trace log data

```
user@R3> show log isis.log
Mar 15 05:15:53.065303 Sending L2 LAN IIH on ae0.0
Mar 15 05:15:53.401852     max area 0, circuit type l1
Mar 15 05:15:53.401955     hold time 27, priority 64, circuit id R3.00
Mar 15 05:15:53.402020     speaks IP
Mar 15 05:15:53.402069     speaks IPv6
---(more)---
```

  - Can you spot the issue?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Debugging IS-IS at R2

Enabling traceoptions on R2, you can determine if R2 is sending any Hello PDUs. The traceoptions log file shows that R2 does send Hello PDUs, but they are point-to-point IS-IS Hello PDUs whereas the R3 log file shows that R3 is sending Level 2 LAN IS-IS PDUs.

Troubleshooting R3 Adjacencies (6 of 6)

- Solution
  - Make sure that the ae0.0 interface at both R2 and R3 is configured as either point-to-point or broadcast interface.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Solution

To ensure that the remaining neighbor adjacency is up, configure the ae0.0 interface type on both sides as either point-to-point or broadcast.

Summary

- In this content, we:
  - Listed useful commands to troubleshoot and verify different IS-IS problems
  - Troubleshot and isolated different IS-IS issues

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- Useful commands that can be used to troubleshoot and verify correct operation of IS-IS; and
- Troubleshot and isolated different IS-IS issues.

Review Questions

1. Is IPv6 routing enabled by default in IS-IS? What about IPv4?
2. Which address family must be enabled on an interface to run IS-IS?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

Lab: Troubleshooting IS-IS

- Troubleshoot IS-IS issues.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: Troubleshooting IS-IS

The slide provides the objective for this lab.

Answers to Review Questions

1.

Yes, IPv6 routing is enabled by default in IS-IS. IPv4 routing is also enabled by default on IS-IS.

2.

IS-IS requires family **`iso`** on the interfaces participating in protocol communication.

Education Services

Advanced Junos Service Provider Routing

Chapter 10: BGP

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - Describe basic BGP operation
  - List common BGP attributes
  - Explain the route selection process for BGP
  - Describe how to alter the route selection process
  - Configure some advanced options for BGP peers

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- Basic BGP operation;
- Common BGP attributes;
- The route selection process;
- The alteration of the route selection process; and
- The configuration of some advanced options for BGP peers.

Agenda: BGP

→Review of BGP
- BGP Operations
- BGP Path Selection
- Load Balancing Options
- Configuration Options

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review of BGP

The slide lists the topics we will discuss. We discuss the highlighted topic first.

BGP Review

- BGP is the core routing protocol within the Internet
- BGP is a path vector protocol
  - Distance is determined by the number of Autonomous Systems traffic must take to get to a destination.
- BGP supports two different methods of route exchange
  - EBGP used between Autonomous Systems
  - IBGP used between routers inside an Autonomous Systems
- BGP is about control
  - Many attributes available to use by policies to provide this control

© 2020 Juniper Networks, Inc. All Rights Reserved.

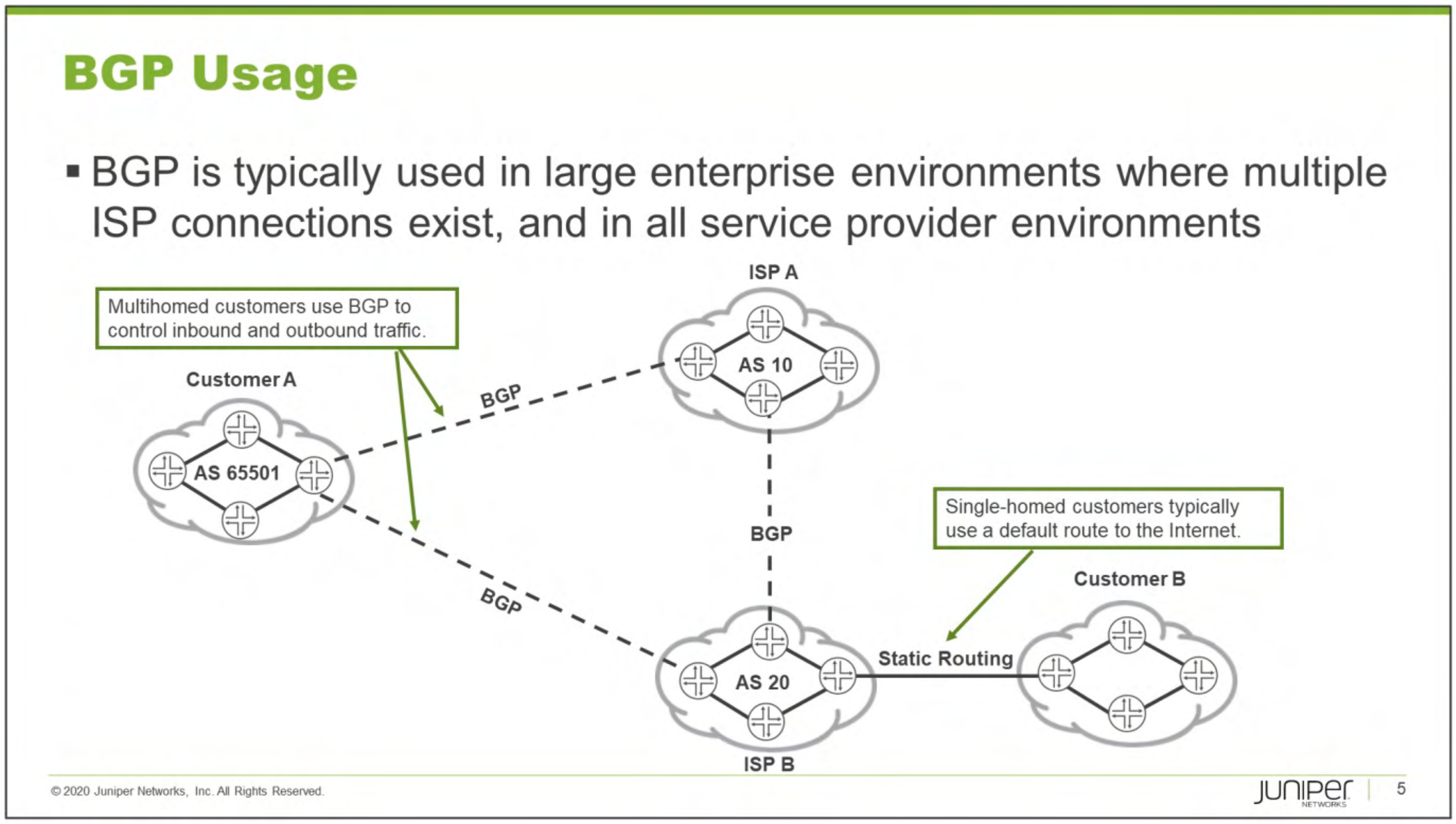
BGP Review

BGP is an Exterior Gateway Protocol, and is the core routing protocol of the Internet. It is used to reliably distribute large quantities of routes between autonomous systems (ASs). It is sometimes referred to as a *path-vector routing protocol* because it uses an AS path as a vector to prevent inter-domain routing loops. The term *path vector*, in relation to BGP, means that BGP routing information includes a series of AS numbers, indicating the path that a route takes through the network. Although BGP is primarily used for inter-AS routing, BGP is also used in large networks for MPLS-based virtual private networks (VPNs) and is used to separate large OSPF domains. BGP is much more scalable and offers a greater amount of control through policy than an interior gateway protocol (IGP).

BGP version 4 (BGP4) is essentially the only exterior gateway protocol (EGP) currently used in the Internet. It is defined in RFC 4271, which made the former standard of more than 10 years, RFC 1771, obsolete

BGP supports two different types of exchanges of routing information. Exchanges between ASs are known as *external BGP* (or *EBGP) sessions* and handle inter-AS routing. Exchanges within an AS are known as *internal BGP* (or *IBGP*) sessions, and handle intra-AS routing.

While IGPs are all about speed and use metrics to determine the best path, BGP uses attributes to identify and control traffic. BGP is designed to provide granular control over the hundred of thousands of routes that are used in the Internet today.



BGP Usage

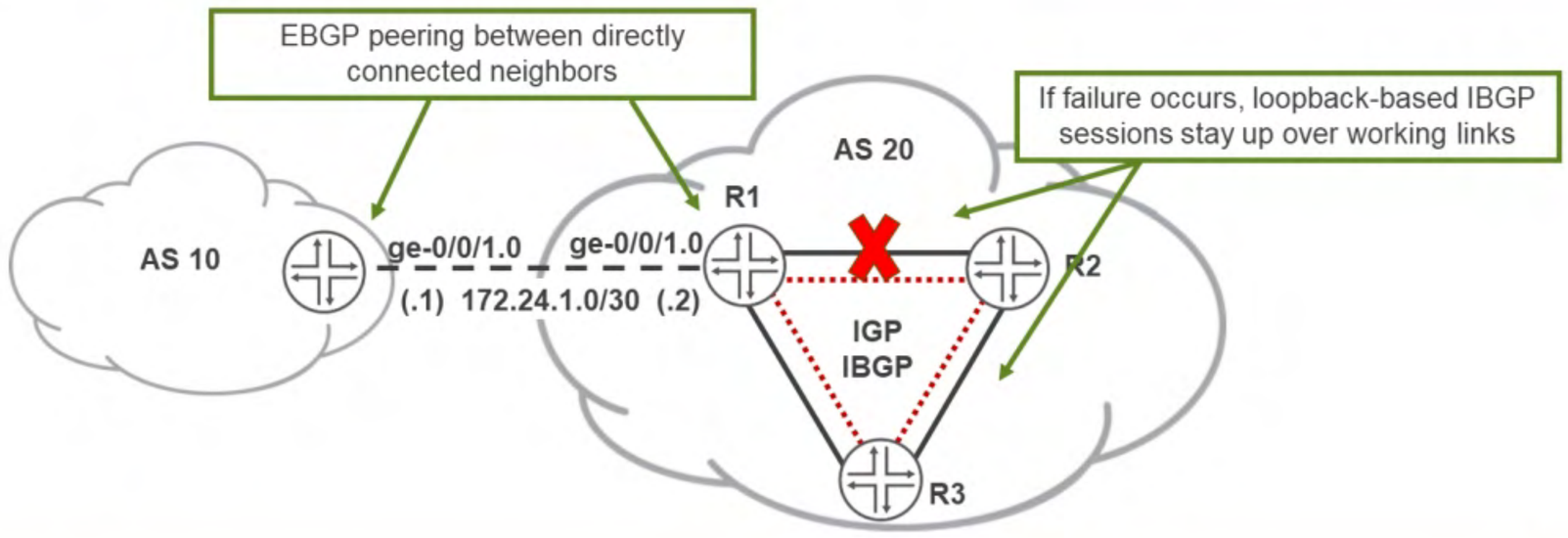
Networks with a single upstream connection receive little benefit from running a dynamic routing protocol with their Internet service provider (ISP). These customers typically use a static default route to send all external traffic toward the Internet. Their provider also typically uses a static route to direct traffic destined for the customer's addresses to the customer. Normally, a single-homed network uses addresses assigned by the provider from the provider's aggregate. Because these addresses are assigned to the provider and can only be used by the customer while they are a customer of the provider, they are known as *nonportable* addresses. Using these addresses allows the provider to announce a single aggregate route for many customer networks, reducing global routing table growth. Currently, the Internet routing table contains hundreds of thousands of routes, which highlights the need for a scalable and robust protocol such as BGP.

BGP is normally used when a network has multiple upstream connections, either to a single ISP or to multiple ISPs. BGP's policy controls provide the ability to optimize inbound and outbound traffic flows based on a network's technical and business constraints. Although BGP can detect and route around failures in redundant environments, BGP sessions within the same AS do not typically react as quickly as an IGP, and they often rely on the IGP used in the AS to remain operational when failures occur.

Networks that are multihomed to a single ISP likely use nonportable addresses assigned by the provider. Networks that are multihomed to multiple ISPs likely use portable addresses assigned directly by the regional address registry.

BGP Peering

- EBGP sessions are usually established using the IP addresses of the physically connected interfaces
- IBGP sessions are usually established between loopback addresses
  - Uses IGP to maintain sessions regardless of physical topology



© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Peers

An EBGP peer connection is between a device in one AS and another device in a different AS. The connection between the two ASs consists of a physical connection and a BGP connection. The physical connection is a shared Data Link Layer subnetwork between the two ASs. On this shared subnetwork, each AS has at least one border gateway belonging to that AS. The BGP connection exists between BGP speakers in each of the ASs. EBGP will prepend the local AS number as it forwards routes to its peers as well as change the BGP next-hop to be its peering address. EBGP uses the AS path as its loop-prevention mechanism.

An IBGP connection is typically established between internal routers which may or may not be directly connected. Because the IBGP connection typically exists between remotely connected routers, there is a requirement for an IGP within the AS. BGP's TCP session is established using regular routing tables. IBGP, by default, will not change any information in the BGP update message. It has no intrinsic loop prevention mechanism, As such an IBGP router will not forward an IBGP routing update. This means for all IBGP routers to receive BGP updates they must be connected to all other IBGP routers.

EBGP normally uses Interface Peering

The EBGP connection typically is established between immediately connected devices located in two different ASs because the time-to-live (TTL) value of the EBGP packets is equal to 1, by default. The session is established using the IP addresses assigned to the interfaces on that subnet as the session endpoints. By establishing the EBGP session using the IP address assigned to the interfaces on the shared subnet, you gain many advantages. One of these advantages is that you prevent either AS from needing to maintain any routing information about the other AS (besides what it received through BGP). You also ensure that all traffic flows over this particular shared subnet.

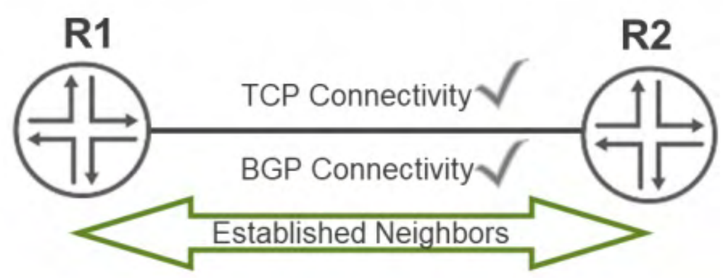
IBGP normally uses Loopback Peering

You maintain only one IBGP session between each internal peer. The IGP is used to maintain reachability between the loopback addresses regardless of the physical topology, allowing the IBGP sessions to stay up even when the physical topology changes. The physical topology is relevant in one respect: each router along the path between BGP speakers must have enough information to make consistent routing decisions about packet forwarding. In many cases, this requirement means that all routers along all possible physical paths between BGP speakers must run BGP; however, in some networks this requirement is not necessary.

BGP Message Types

- BGP uses 5 different messages to establish and maintain BGP peering sessions
  - All BGP messages use a common header

BGP Message Types	
Open	Keepalive
Update	Notification
Refresh	



© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Message Types

BGP processes a message only after the entire message is received. The maximum message size is 4096 bytes the smallest BGP message is a header without any data, which would be 19 bytes. The following list details the BGP message types:

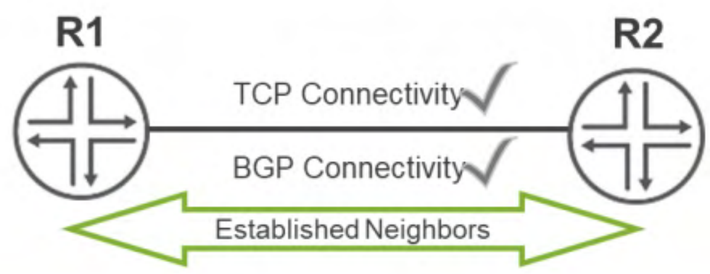
- *Open message*
- *Update message*:
- *Keepalive message*
- *Notification message*
- *Refresh*

Each BGP message uses the same fixed size header, which is 19 bytes. BGP keepalive messages do not include any data portion following the header.

BGP Peering Sessions

- BGP peering sessions are manually defined and rely on TCP connections
  - No automatic neighbor discovery

BGP Neighbor States	
TCP Connectivity	**BGP Connectivity**
Idle	OpenSent
Connect	OpenConfirm
Active	Established



© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Peering Sessions

Unlike other dynamic protocols, BGP requires that you manually define the neighbors with which you want the local device to peer. Because BGP peers must be manually defined, no automatic neighbor discovery exists as with other protocols.

BGP uses TCP as its transport protocol (port 179). TCP provides a full-duplex, connection-oriented, reliable, byte-stream service to BGP. BGP considers a connection between two peers to be idle until a TCP connection is established between them. With the TCP connection established, the endpoints are assured of a reliable connection. The following list describes the various BGP neighbor states:

- *Idle state*: The Idle state is the initial state when all incoming BGP connections are refused. A start event is required for the local system to initialize BGP resources and prepare for a transport connection with the other BGP peer.
- *Connect state*: In the Connect state, BGP is waiting for the transport protocol connection to be completed. If the transport protocol connection succeeds, the local system sends an OPEN message and transitions to the OpenSent state. If the transport protocol connection fails, the local system restarts the ConnectRetryTimer, listens for a connection initiated by the remote BGP peer, and changes its state to Active.
- *Active state*: In the Active state, BGP is trying to acquire a peer by initiating a transport protocol connection. If the transport protocol connection succeeds, the local system sends an OPEN message to its peer and transitions to the OpenSent state. If the local system's BGP state remains in the Active state, you should check physical connectivity as well as the configuration on both peers.

Continued on the next page.

BGP Peering Sessions (contd.)

The open messages are sent once the TCP three-way handshake is complete. The open message initiates the BGP session and contains details about the BGP neighbor and information about supported and negotiated options.

- *OpenSent state*: In the OpenSent state, BGP waits for an OPEN message from its peer. When an OPEN message is received, it is checked and verified to ensure that no errors exist. If an error is detected, the system transitions back to the Idle state. If no errors are detected, BGP sends a Keepalive message.
- *OpenConfirm state*: In the OpenConfirm state, BGP waits for a KEEPALIVE or NOTIFICATION message. If no KEEPALIVE message is received before the negotiated hold timer expires, the local system sends a NOTIFICATION message stating that the hold timer has expired and changes its state to Idle. Likewise, if the local system receives a NOTIFICATION message, it changes its state to Idle. If the local system receives a KEEPALIVE message, it changes its state to Established.
- *Established state*: In the Established state, BGP can exchange UPDATE, NOTIFICATION, and KEEPALIVE messages with its peer. When the local system receives an UPDATE or KEEPALIVE message, and when the negotiated hold timer value is nonzero, it restarts its hold timer. If the negotiated hold timer reaches zero, the local system sends out a KEEPALIVE message and restarts the hold timer.

BGP Update Messages

- BGP update messages include path advertisements and their associated attributes:
  - NLRI
  - Origin
  - AS Path
  - BGP next-hop
  - Additional attributes include: Local preference, MED, and communities
  - Can also list withdrawn routes that are no longer reachable

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Update Messages.

Once BGP has achieved the established state, update messages can be exchanged reliably. All update messages contain required information which includes the Network Layer Reachability Information (also called the NLRI), an origin attribute, the AS Path attribute and the BGP protocol next-hop.

The primary purpose of BGP is not to find the *shortest* path to a given destination; rather, its purpose is to find the *best* path. Each AS determines the best path to a prefix by taking into account its own outbound routing preferences, the inbound routing preferences of the route's originator (as updated by ASs along the path between the source and destination ASs), and some information that is collected about the path itself. All this information is contained in *path attributes* that describe the path to a prefix. The path attributes contain the information that BGP uses to implement the routing policies of source, destination, and transit ASs

Other BGP Messages

- Keepalives
  - Used to maintain BGP sessions
  - Configured using the `holdtime` option
  - `Holdtime` is 3 times the keepalive interval
  - Default `holdtime` is 90 seconds
- Notifications
  - Sent when an error is detected with the BGP session such as a hold timer expiring, neighbor capabilities change
- Route Refresh
  - Used to ask a BGP peer to resend all routes of a particular address family

© 2020 Juniper Networks, Inc. All Rights Reserved.

Other BGP Messages

The following three messages are also used by BGP:

- *Keepalive message*: BGP does *not* use keepalives at the Transport Layer. Instead, peers exchange keepalives to ensure that the hold timer does not expire. The hold time is three times the interval at which keepalive messages are sent.
- *Notification message*: BGP uses notification messages to signal when something is wrong with the BGP session. A notification is sent when an unsupported option is sent in an open message and when a peer fails to send an update or keepalive. When an error is detected, the BGP session is closed.
- *Refresh*: Normally a BGP speaker cannot be made to readvertise routes that have already been sent and acknowledged (using TCP). The route refresh message supports *soft clearing* of BGP sessions by allowing a peer to readvertise routes that have already been sent. This soft clearing has some very specific uses when working with MPLS-based VPNs and adding new customer sites to existing customer VPN structures.

Agenda: BGP

- Review of BGP
- →BGP Operations
- BGP Path Selection
- Load Balancing Options
- Configuration Options

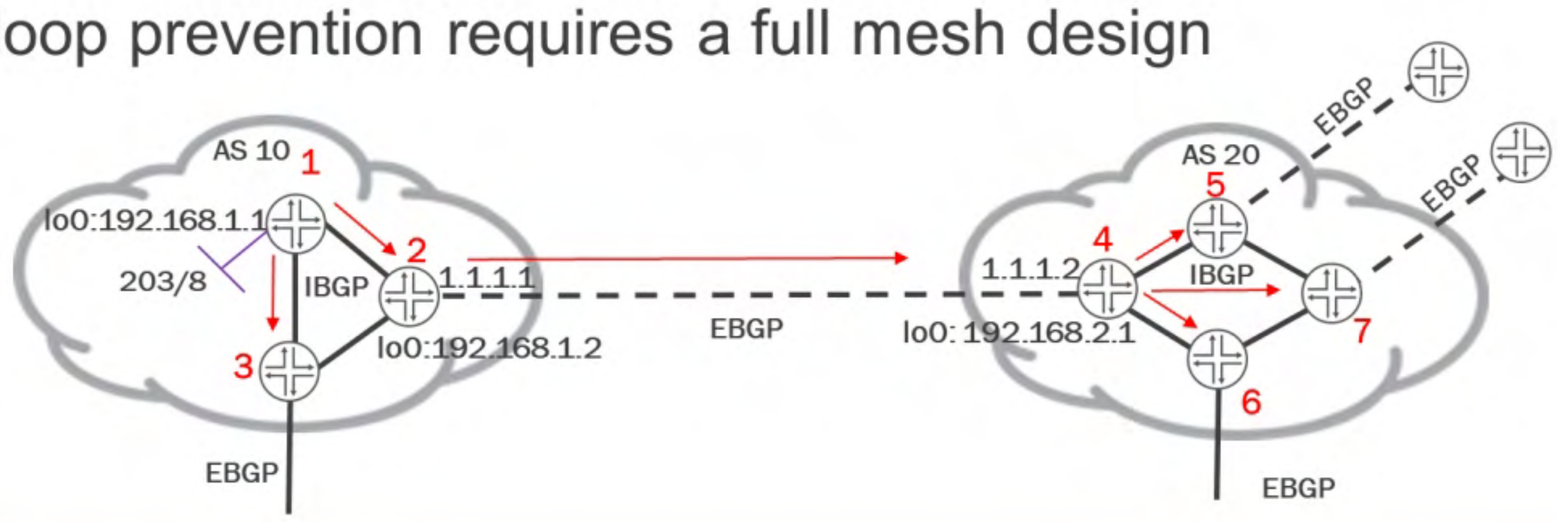
© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Operations

The slide highlights the topic we discuss next.

Default BGP Advertisement Rules

- EBGP/IBGP routers advertise EBGP learned routes to EBGP and IBGP Peers
- EBGP routers advertise IBGP routes to EBGP peers
- EBGP/IBGP routers do not advertise IBGP learned routes to IBGP peers
- IBGP loop prevention requires a full mesh design



© 2020 Juniper Networks, Inc. All Rights Reserved.

Default BGP Advertisement Rules

By default, only active BGP routes are advertised. The slide illustrates the default BGP advertisement rules. The rules are as follows:

1. IBGP peers advertise routes received from EBGP peers to other IBGP peers.
2. EBGP peers advertise routes learned from IBGP or EBGP peers to other EBGP peers.
3. IBGP peers do not advertise routes received from IBGP peers to other IBGP peers.

The purpose of the advertisement rules is to prevent routing loops on a BGP network.

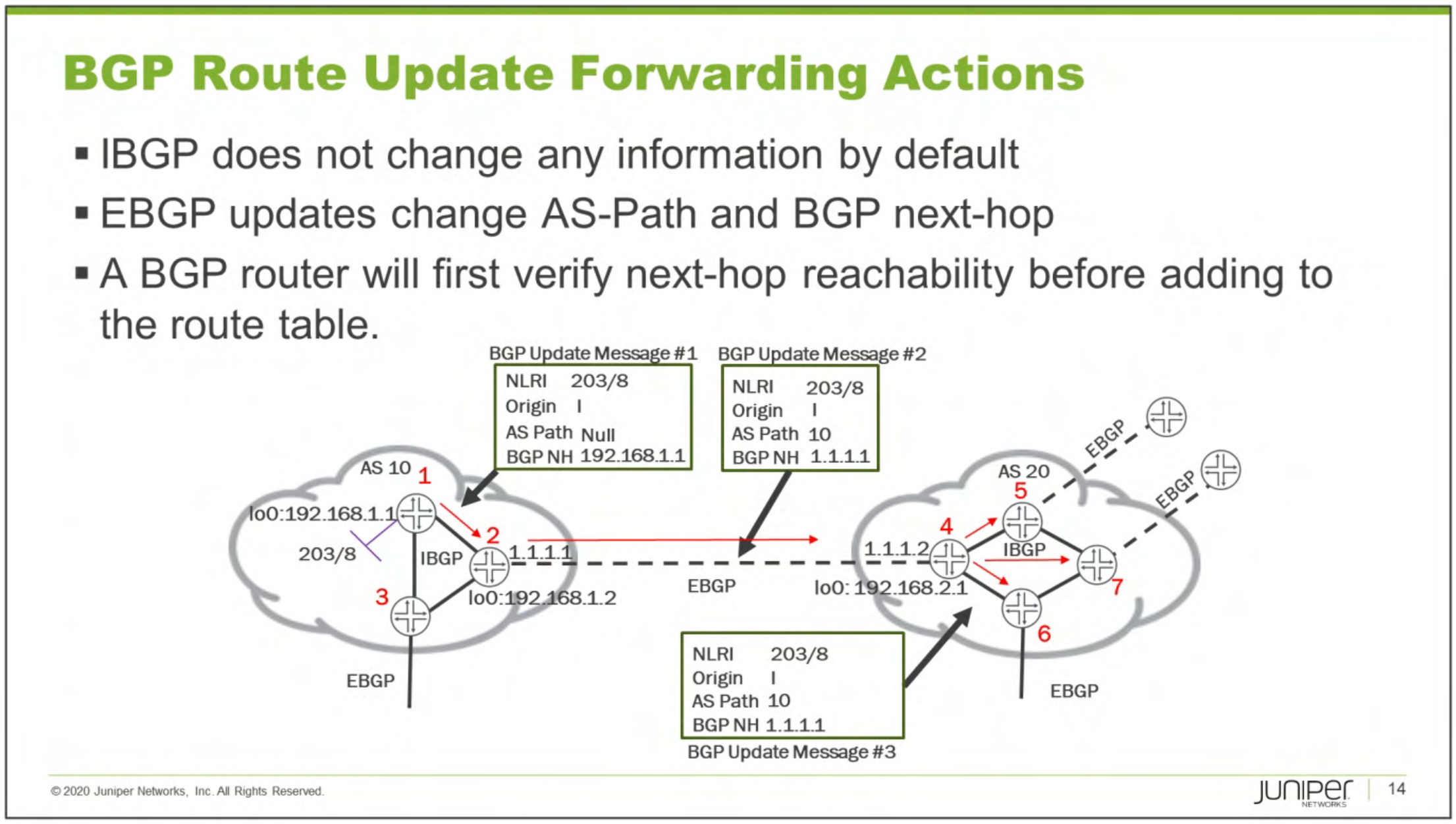
IBGP Route Propagation

IBGP speakers send routes to their IBGP peers that they received from EBGP peers and routes that they originated themselves. IBGP speakers never send routes to IBGP peers that they learned from other IBGP peers. For all IBGP speakers in an AS to have consistent routing information, a full mesh of IBGP sessions must exist between all BGP speakers. Without this full mesh, some BGP speakers might not receive all the required routing information.

In the example on the slide, a full mesh of IBGP sessions does not exist. Router 4 receives the announcement through an EBGP session. Because it is the best route it has for that prefix, it propagates the route to its IBGP peers Router 5 and Router 6. if they determine that route to be its best path for the prefix; it cannot send the route to its IBGP peer Router 7. Because it received the route through IBGP, it cannot send the route to any IBGP peers. Therefore, R7 does not receive or install a route for the prefix advertised from AS 10. This situation can be alleviated by adding an IBGP session between Router 4 and Router 7. (It is irrelevant whether the two routers are directly connected.)

If IBGP routers readvertised IBGP routes to other IBGP peers, a loop would form. Because the AS path is not updated by each router, but rather only when the associated prefix is advertised to an EBGP peer, the AS path cannot be used to detect loops for BGP routes advertised within an AS. For this reason, BGP enforces advertisement rules that require the full-mesh peering of IBGP routers to ensure consistent routing information on all IBGP routers within the AS.

Using route reflectors or confederations can also alleviate this situation, both of which can reduce or alleviate the full-mesh requirement.



BGP Route Update Forwarding Actions

When a route is introduced into BGP as shown on the slide at Router 1 it will be forwarded to the devices BGP peers.

Since Router 1 in an internal BGP router, the 203/8 prefix will be included in the NLRI of the update packet. In addition the Origin (how the route was learned) will be added to the update. Since the update is internal, there will not be a change to the AS-Path. An empty AS-Path attribute is called a “Null” path. Finally in this example since the 203/8 was an originally a static route with no next-hop, the BGP next-hop is set to the IBGP peering address of the redistributing router.

When Router 3 receives the update, the first thing that is checked is the BGP next-hop. If it is reachable from the inet.0 table, the route can be added to the routing table of Router 3. If it is determined to be the best route based on the BGP Route selection criteria, then it will be forwarded based on the rules just discussed. Remembering the rules of BGP route advertisements, this update will be forwarded to router 4.

The update will follow the rules for EBGP where the AS-Path will be pre-pended with the AS number of the advertising router, and the BGP next-hop will be updated to the BGP peer address which is its interface address.

When router 4 receives the update, it will check to see if it can resolve the BGP next-hop (which it can as it is directly connected) and ensure there is no loop by checking the AS Path and checking for its own AS number. Since this is not an issue the route can be added to the route table and if the route is selected as the best route it will be forwarded to any BGP peers (both IBGP and EBGP peers.

A new update message will be created to send to IBGP peers. In this case neither the AS number nor the BGP next-hop is changed. In the example on the screen, an IBGP update is sent to Router 6. Router 6 will see if it can resolve the BGP next-hop. Depending on your BGP configuration, the answer might be no. If this is the case then the route will be marked as hidden and not be usable in the inet.0 table for forwarding traffic. There are several ways to resolve this issue such as **`next-hop self`**, or marking the BGP link on Router 4 as **`passive`** in the IGP. These options are discussed next.

Hidden Routes

- Hidden routes are routes that the device cannot use
- Reasons routes might not be usable:
  - Unresolvable next-hop
  - Import policy
  - Aggregate or generate route that does not have any contributing routes
- Number one reason: Unresolvable next hop

```
user@router> show route hidden extensive
inet.0: 24 destinations, 24 routes (23 active, 0 holddown, 1 hidden)
 BGP     Preference: 170/-101
                 Next hop type: Unusable, Next hop index: 0
                 Address: 0x9f2fec4
                 Next-hop reference count: 6
                 State: <Hidden Int Ext>
                 Local AS: 65001 Peer AS: 65001
                 Age: 28
                 Validation State: unverified
                 Task: BGP_65001.172.16.1.2
                 AS path: 65020 I
                 …
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Hidden Routes

One might expect all routes received from a BGP peer would be installed in the RIB-LOCAL table and be visible using the **`show route protocol bgp`** command. This is not always true. There are several reasons why:

- An import policy rejects the route.
- The next hop cannot be resolved using the indirect next hop resolution rule. Because routing protocols such as internal BGP (IBGP) can send routing information about indirectly connected routes, Junos OS relies on routes from intra-AS routing protocols (OSPF, IS-IS, RIP, and static) to resolve the best directly connected next hop.
- A damping policy suppresses the route.
- The AS path contains illegal or invalid confederation attributes.
- The next hop address is the address of the local routing device.
- The AS path contains illegal or invalid transitive attributes.
- The AS path is empty. This only applies to EBGP. For IBGP, an empty AS path is normal.
- The AS path contains a zero.
- The next hop address is a multicast address.
- The next hop address is an IPv6 link-local address.
- The route prefix or the route next hop is a martian address.
- The LDP (Label Distribution Protocol) session fails. The received routes are not installed in the routing table until the peer router reestablishes the LDP session.

Unresolvable Next-Hop

The number one reason for hidden BGP routes is an unresolvable next-hop. The BGP Update message contains a protocol next-hop IP address. If the router cannot resolve this address using its routing table, the route cannot be used and is not installed in the routing table. The number of hidden routes is always shown in the output of the **`show route`** command. To view the reason why routes are hidden, issue the **`show route hidden extensive`** command.

BGP Next-Hop Solutions

- BGP next-hop solutions:
  - Next-hop self
    - Use a policy to alter the next-hop value
    - Change the BGP next hop to be the address of the IBGP peer
    - Apply as export to IBGP
  - IGP passive interface
    - IGP advertises external interface prefixes to IBGP peers, no adjacency formed
    - Adds external interface prefixes to the IGP routing tables
  - Redistribute interface (direct) routes
    - Possible but not desirable
  - Create static routes throughout the network
    - Possible not plausible

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Next-Hop Solutions

Numerous ways exist to solve this BGP next-hop reachability problem. However, only two are considered as best practices in a networking environment.

The most commonly used (and recommended) solution is **`next-hop self`**. With this solution, when a BGP router advertises an EBGP-learned route to an IBGP peer, it alters the BGP next-hop attribute. The next-hop attribute's IP address of the remote EBGP peer is replaced with the IP address of the BGP router itself. Because the IBGP session was most likely established using the peer's loopback address, this new BGP next-hop value is reachable, and the advertised BGP route can be used. We create **`next-hop self`** by using a policy to match specific routes with an action of changing the next-hop attribute value. The Junos OS then applies this policy as an export policy to any IBGP peers.

With IGP passive, the IGP is configured on the inter-AS link and advertises the interface addresses, but forms no adjacency (it is *passive*). Both methods inject the interface addresses into the local routing table for the IGP to use.

An IGP passive interface allows the local IGP to advertise the subnet on a particular interface without forming an adjacency at the IGP level to the remote EBGP peer, which has the advantage of not using a policy, but it requires explicit configuration for each interface and subnet address that you want to advertise.

Agenda: BGP

- Review of BGP
- BGP Operations
- →BGP Path Selection
- Load Balancing Options
- Configuration Options

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Path Selection

The slide highlights the topic we discuss next.

Review: Active BGP Route Selection Process

- Once BGP verifies next-hop reachability and that no loops exist, it selects the active route as follows:

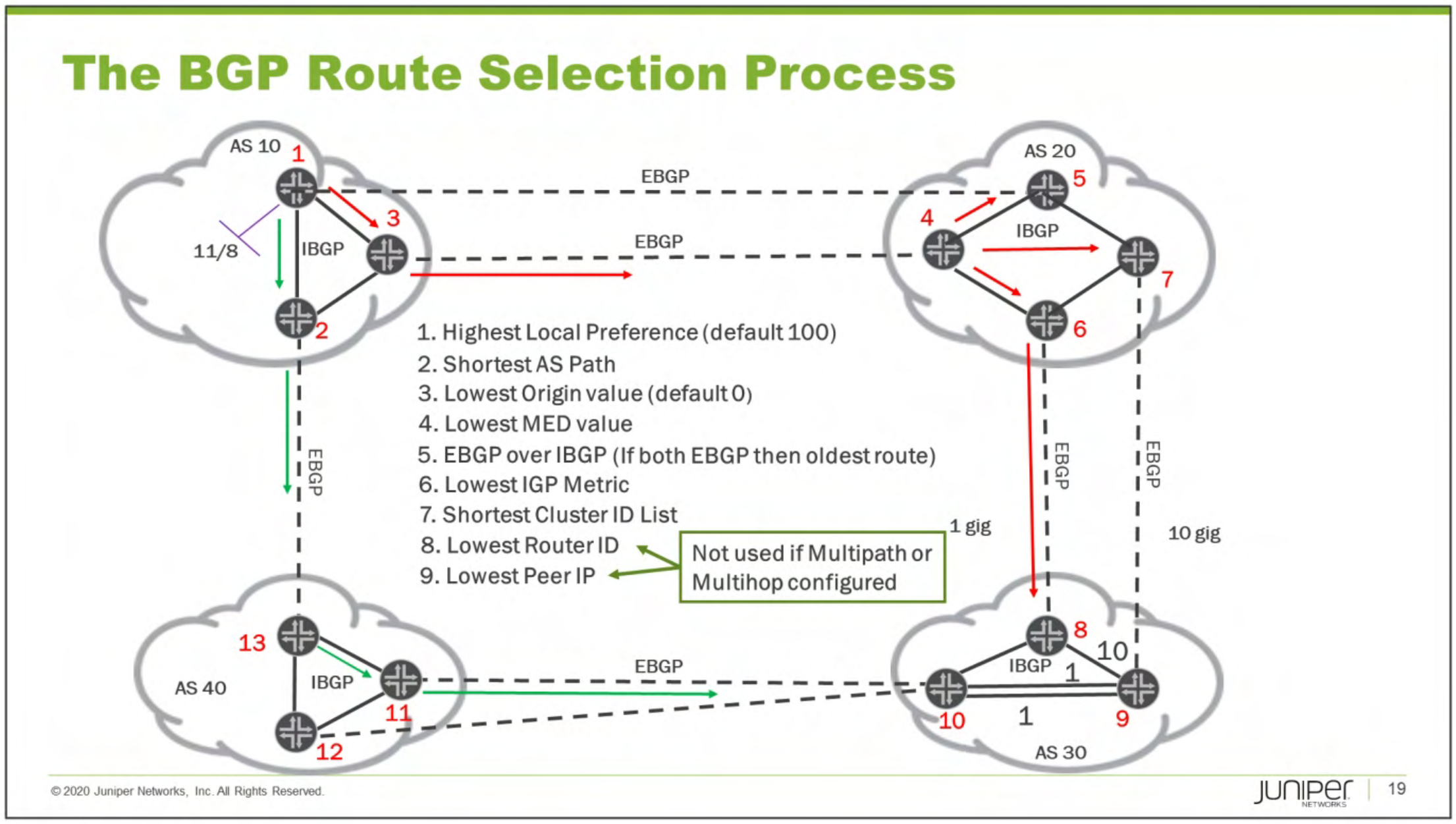
BGP Route Selection Summary	
1. Prefer the path with the higher local-preference	6. Prefer path whose next-hop is resolved by IGP route with lowest IGP metric
2. Prefer the route with the shortest AS-path length	7. Prefer routes with the shortest cluster list length
3. Prefer the route with the lower origin code	8. Prefer routes from the peer with the lowest router ID
4. Prefer the path with the lowest MED metric	9. Prefer routes from the peer with the lowest peer IP address
5. Prefer routes learned from an EBGP peer over an IBGP peer, If only EBGP routes, prefer the current active route	

© 2020 Juniper Networks, Inc. All Rights Reserved.

Summary of BGP Active Route Selection

Before the router installs a BGP route, it must ensure that the BGP next-hop attribute is reachable and that no routing loops exist. If the BGP next hop cannot be resolved or if a loop is detected, the route is not evaluated through the BGP route selection process or installed in the route table.

Before the Junos OS installs a BGP route in the route table, the route preference is evaluated. Recall that the route preference can be changed through policy, so the route preference can differ for the same prefix learned through different BGP paths. If the route preference for a BGP prefix learned through different BGP paths differs, the BGP route with the lower route preference is selected. Note that this evaluation occurs prior to the BGP selection process outlined on the slide.



The BGP Route Selection Process

When a BGP route is installed in the routing table, it must go through a path selection process if multiple routes exist to the same destination prefix and the route preference is the same. The BGP path selection process proceeds in the following order:

1. The router compares routes for the *highest* local preference (the only choice based on a higher, rather than lower, value).
2. The router evaluates the AS-path attribute next, where a shorter path is preferred. This attribute is often a common tiebreaker for routes. (To understand why this might not work evaluate paths from perspective of Router 8)
3. The router evaluates the origin code. The lowest origin code is preferred: (I [IGP] < E [EGP] < ? [Incomplete]).
4. If any of the remaining routes are advertised from the same neighboring AS, the router checks the multiple exit discriminator (MED) attributes for the lowest value. The absence of a MED value is interpreted as a MED of 0. (To understand why this might not work, evaluate perspective from Routers 8 and 9)
5. If multiple routes remain, the router prefers any routes learned through an EBGP peer over routes learned through an IBGP peer. If all remaining routes were learned through EBGP choose the route that has been in the routing table the longest. This will result in the most stable route being used. (To understand why this might not work, evaluate from the perspective of Router 8 first then use Router 10)
6. If the remaining routes were learned through IBGP, use the path with the lowest IGP cost to the IBGP peer. (To understand why this might not work, evaluate from the perspective of Router 9 and use a cost of 1 to Router 8.)
7. The router then examines the cluster-list attribute for the shortest length. The cluster list is similar in function to an AS path.
8. Prefer the path from the peer with the lowest router ID. For any path with an originator ID attribute, substitute the originator ID for the router ID during router ID comparison. (To understand why this might not work, evaluate from the perspective of Router 9)
9. The router prefers routes from the router with the lowest peer IP address. (To understand why this might not work, evaluate from the perspective of Router 9)

Agenda: BGP

- Review of BGP
- BGP Operations
- BGP Path Selection
- →Load Balancing Options
- Configuration Options

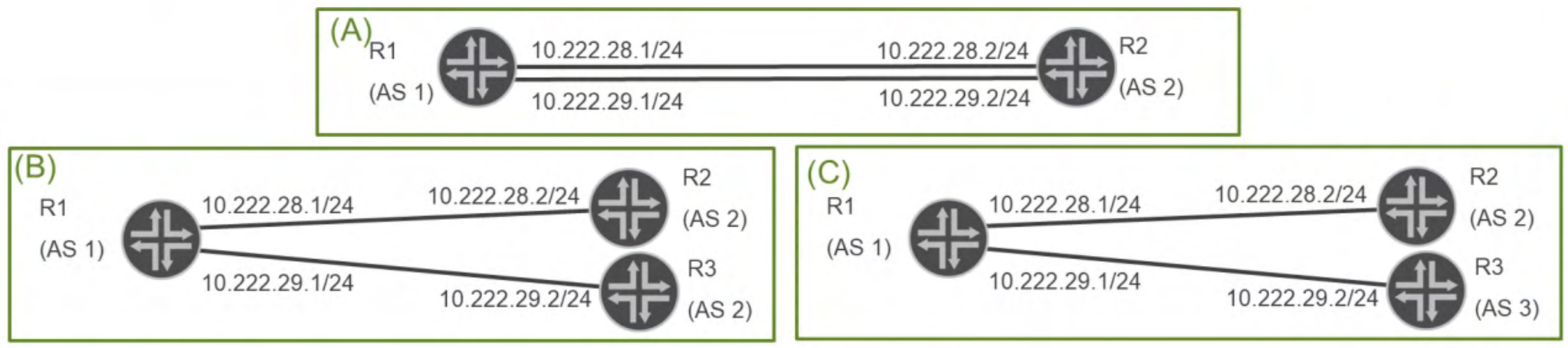
© 2020 Juniper Networks, Inc. All Rights Reserved.

Load Balancing Options

The slide highlights the topic we discuss next.

BGP Multipath (1 of 3)

- BGP can ignore both RID and peer ID comparisons when `multipath` is configured within BGP
  - Can use with:
    - Two different peering sessions to the same router (A)
    - Two different peering sessions to different routers in the same AS (B)
    - Two different peering sessions to different routers in different ASs using `multipath multiple-as` (C)



© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Multipath (1 of 3)

When you configure **`multipath`** on a BGP router, the selection algorithm ignores both the router ID and the peer ID selection criteria. Should multiple copies of a route reach those portions of the route selection process, BGP installs all copies into the local routing table. Each version is listed in the table with only one of them marked as active. This active route is the version of the route that would have been selected by the algorithm had the **`multipath`** option not been configured. However, the next-hop values for the nonactive routes are also listed as valid next hops for the active route. This listing allows the Junos default load-balancing options to be used.

The Junos OS also utilizes the link bandwidth extended community to unequally load-balance traffic in conjunction with the **`multipath`** command. If used, data packet forwarding is performed in a proportional manner to the bandwidth advertised in the extended community.

The **`multipath`** command allows multiple copies of a route from the same remote router. It also allows multiple copies of a route from two different routers in the same AS (either a local or remote AS). If you are connected to two different ASs and want to load balance the traffic, then you must use the **`multipath multiple-as`** option. The multipath option is used for resiliency and redundancy.

BGP Multipath (2 of 3)

- Without multipath, routes from each peer contain a single next hop

```
[edit protocols bgp group ext-peers]
type external;
peer-as 2;
neighbor 10.222.28.2;
neighbor 10.222.29.2;
}
user@R1> show bgp summary
[...]
Peer         AS  InPkt  OutPkt  OutQ  Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
10.222.28.2  2     53      53     0      0       22:48 4/4/4/0                0/0/0/0
10.222.29.2  2     54      51     0      0       22:48 0/4/4/0                0/0/0/0
```

Multipath not configured

Routes from lower Peer ID selected as Active

```
user@R1> show route protocol bgp
inet.0: 8 destinations, 12 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.20.4/30      *[BGP/170] 00:46:13, localpref 100
                       AS path: 2 I, validation-state: unverified
                     > to 10.222.28.2 via ge-1/0/4.0
                     [BGP/170] 00:46:13, localpref 100
                       AS path: 2 I, validation-state: unverified
                     > to 10.222.29.2 via ge-1/0/5.0
...
```

Single next hop per active route

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Multipath (2 of 3)

The local routing table of the R1 router is shown on the slide. We see the one of the four routes advertised from AS 2: 172.16.20.4/30, This listing of the prefix contains two versions of the route. One route is from the R2 router (10.222.28.2), and this route is marked active (indicated by the asterisk). The other version of the route is from the R3 router (10.222.29.2), and it is not marked as active.

BGP Multipath (3 of 3)

- With multipath, the active peer route contains more than one next hop

```
[edit protocols bgp group ext-peers]
type external;
peer-as 2;
multi-path;
neighbor 10.222.28.2;
neighbor 10.222.29.2;
}
user@R1> show bgp summary
[...]
Peer         AS   InPkt   OutPkt   OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
10.222.28.2   2      53       53      0       0       22:48 4/4/4/0                0/0/0/0
10.222.29.2   2      54       51      0       0       22:48 4/4/4/0                0/0/0/0
```

Multipath configured

Routes from both peers selected as Active

```
user@R1> show route protocol bgp
inet.0: 8 destinations, 12 routes (8 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

172.16.20.4/30     *[BGP/170] 00:49:55, localpref 100
                      AS path: 2 I, validation-state: unverified
                    > to 10.222.28.2 via ge-1/0/4.0
                      to 10.222.29.2 via ge-1/0/5.0
                    [BGP/170] 00:49:55, localpref 100
                      AS path: 2 I, validation-state: unverified
                    > to 10.222.29.2 via ge-1/0/5.0
...
```

Two next hops per active route. However, by default, the forwarding table maintains a single next hop per route.

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Multipath (3 of 3)

The configuration of the R1 router now contains the **`multipath`** command within the peer group for AS 2. After committing the configuration, we examine the contents of the local routing table. We still see the four routes advertised from AS 2, and each listing of the prefix still contains two versions of the route. As before, the routes from the R2 router are marked as active while the routes from the R3 router are marked as inactive.

The effect of the **`multipath`** command on the routes from AS 2 is that the next hop for the routes from R3 (10.222.29.2) are now added to the version of the route from R2. The next-hop information for the inactive route version does not change in this environment.

Because each active route now has two next hops in the routing table, the default Junos load-balancing process can be used. Each route has a single next hop selected, and this single next hop is placed into the forwarding table. All traffic for each route uses just that single next hop. The overall benefit of this system is the total amount of traffic sent from AS 1 to AS 2 can now be load-balanced over the two inter-AS links. In our example, the route shown has two possible paths and has chosen 10.222.28.2 as its next-hop, As more routes are announced between the AS networks, the prefixes are hashed and distributed among the possible next-hops.

You can also see the effects of using multipath in the output of the **`show bgp summary`** command. The route information from both R2 and R3 now appears as 4/4/4/0. The routes from R2 are active while the next-hop values from R3 are also being used to forward user traffic.

Multihop Peering

- EBGP sessions can peer with nonphysical addresses

```
[edit protocols bgp group ext-peers]
type external;
local-address 192.168.3.4;
neighbor 172.16.128.1 {
   multihop ttl 1;
}
[edit routing-options]
static {
   route 172.16.128.1 next-hop [ 10.10.1.1 10.10.2.1 ];
}
```

A TTL value of 1 accommodates peering to a loopback address on a directly connected peer—higher values are needed for peers that are not directly connected

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Multihop Peering

The default for an EBGP connection is to peer over a single physical hop using the physical interface address of the peer. In some cases, it is advantageous to alter this default, one-hop, physical peering EBGP behavior. One such case is when multiple physical links connect two routers that are to be EBGP peers. In this case, if one of the point-to-point links fails, reachability on the alternate link still exists. You must take three extra configuration steps to accomplish a single BGP peering session across these multiple physical links.

First, each router must establish the peering session with the loopback address of the remote router. You can configure this session using the **`local-address`** command, which alters the peer address header information in the BGP packets. Second, use the **`multihop`** command to alter the default use of the neighbor's physical interface address. In addition, you can also specify a TTL value in the BGP packets to control how far they propagate. On the slide, we use a TTL value of 1 to ensure that the session cannot be established across any other *backdoor* links in the network. Third, each router must have IP routing capability to the remote router's loopback address. As the slide shows, we often accomplish this capability by using a static route to *map* the loopback address to the interface physical addresses.

Note that when multihop is configured, the Junos OS sets the TTL value to 64, by default. You can specify a TTL value explicitly when wanted to help limit the scope of the EBGP session. Note that a TTL value of 1 is sufficient to enable an EBGP session to the loopback address of a directly connected neighbor because the IP TTL is decremented for egress traffic only, and this traffic will be considered as destined for the local Routing Engine (RE).

Multiple Hops without Per-Flow Load Balancing

- Both multihop and multipath create routes with multiple next hops in the routing table
  - Use a routing policy to forward traffic on both next hops

```
user@router> show route 172.16.20.4/30 terse
inet.0: 10 destinations, 10 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination        P Prf   Metric 1   Metric 2  Next hop         AS path
* 172.16.20.4/30     B 170        100             10.10.2.2        65002 I
                                                 >10.10.1.2
```

Two viable next hops, but only one is being installed in the forwarding table.

```
user@router> show route forwarding-table matching 172.16.20.4/30
Routing table: inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
172.16.20.4/30     user     0                    indr 1048575     5
                                 10.10.1.2       ucst   663     3 ge-1/0/4.0
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Routes with Multiple Next Hops but no Per-Flow Load Balancing

Within the context of a BGP network, both the **multihop** and **multipath** options can result in a route being installed in the local routing table with multiple next hops. As we previously discussed, this route allows the Junos OS to perform per-prefix load balancing as the total amount of traffic sent towards the destinations is spread across the multiple next hops. However, each route selects a single next hop for forwarding traffic, which is installed into the forwarding table on the Packet Forwarding Engine.

The slide shows the BGP route of 172.16.20.4/30, which is active in the routing table. This route has two possible next hops which can be used to forward traffic—10.10.1.2 and 10.10.2.2. It has currently selected the 10.10.1.2 next hop, which we verify in the forwarding table with the **show route forwarding-table** command. The router output shows this single next hop in the table with a type set to **ucst**, for unicast transmission.

Multiple Hops with Per-Flow Load Balancing

```
user@router> show configuration policy-options policy-statement load-balance
then {
    load-balance per-packet;
}
user@router> show configuration routing-options forwarding-table
export load-balance;

user@router> show route 172.16.20.4/30 terse
inet.0: 10 destinations, 11 routes (10 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A V Destination        P Prf   Metric 1   Metric 2  Next hop        AS path
* ? 172.16.20.4/30     B 170        100                            2 I
  unverified                                        10.10.1.1
                                                   >10.10.2.1

user@router> show route forwarding-table matching 172.16.20.4/30
Routing table: inet
Internet:
Destination        Type RtRef Next hop          Type Index NhRef Netif
172.16.20.4/30     user     0                   indr 1048575     5
                                                ulst 1048574     4
                              10.10.1.1         ucst   727     3 ge-1/0/4.0
                              10.10.2.1         ucst   732     3 ge-1/0/5.0
```

Load Balancing Policy

Policy applied to forwarding-table

Route table shows selected next-hop

Two next hops in the forwarding table

© 2020 Juniper Networks, Inc. All Rights Reserved.

Multiple Hops with Per-Flow Load Balancing

You can alter the default behavior of the Junos OS to install a single next hop per route in the forwarding table with a routing policy. The policy should contain the action of **`then load-balance per-packet`** and be applied as an export policy to the forwarding table within the `[edit routing-options]` configuration hierarchy.

After committing the configuration, we see the same 172.16.20.4/30 route in the routing table of the local router. The same protocol information is displayed and again, a single next hop has been selected. This selection mechanism is not affected by our load-balancing policy, so we cannot verify if it is working by examining the routing table. Instead, a look at the forwarding table shows the outcome of our policy. Both the 10.10.1.2 and the 10.10.2.2 next hops are listed as valid outbound interfaces for the 172.16.20.4/30 route. Traffic destined for this route is now forwarded across both available next hops using a microflow hashing algorithm. The default inputs to the microflow hash are the incoming router interface, the source IP address, and the destination IP address. You can modify the inputs to the hashing algorithm at the `[edit forwarding-options hash-key family inet]` configuration hierarchy. Specifying the **`layer-4`** command at this configuration hierarchy incorporates Layer 4 source and destination port information into the hash key.

Agenda: BGP

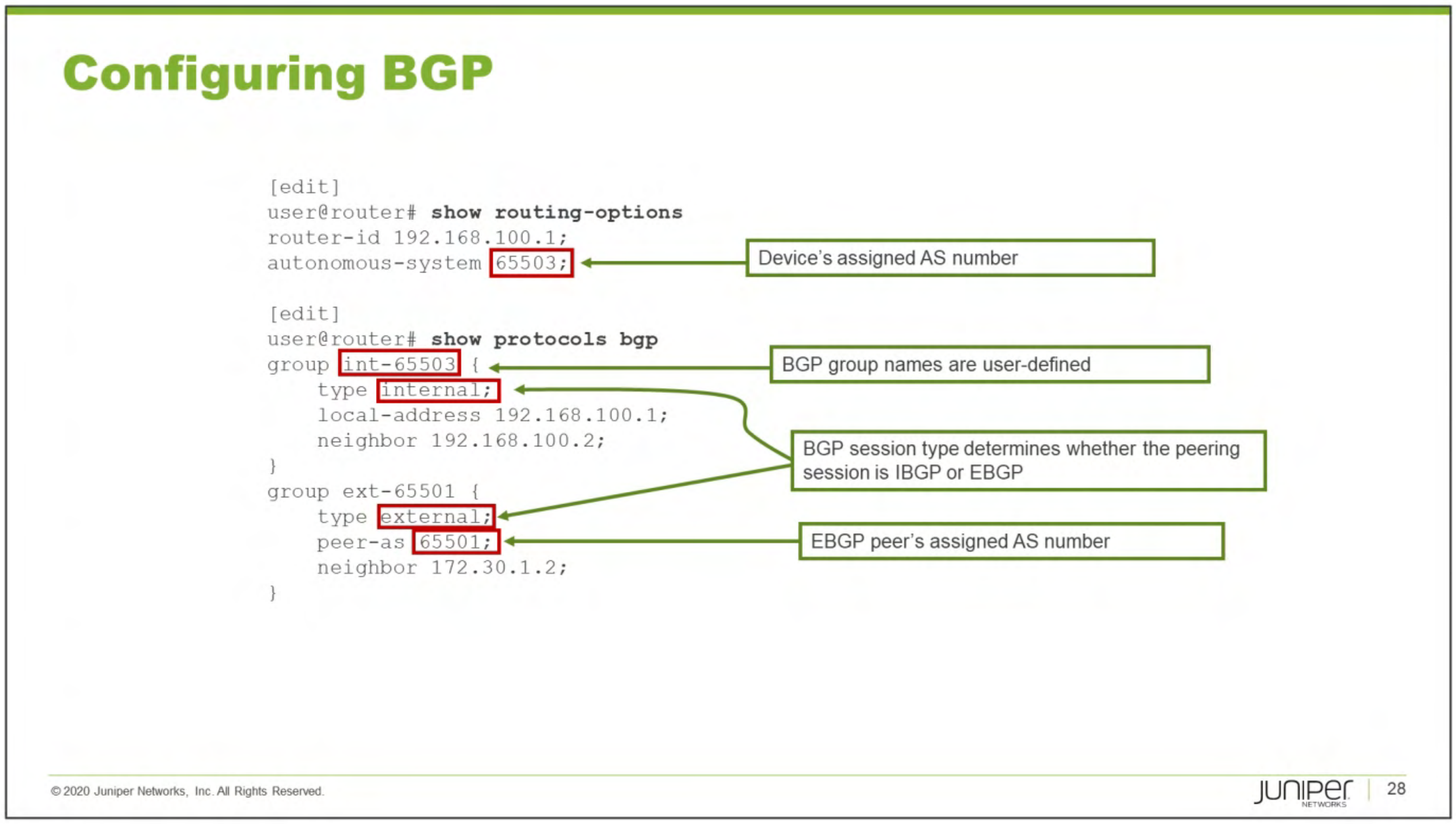
- Review of BGP
- BGP Operations
- BGP Path Selection
- Load Balancing Options

→Configuration Options

© 2020 Juniper Networks, Inc. All Rights Reserved.

Configuration Options

The slide highlights the topic we discuss next.



Configuring BGP

The slide illustrates a sample configuration.

In this configuration example, we see some parameters defined under the `[edit routing-options]` and `[edit protocols bgp]` hierarchies. Under the `[edit routing-options]` hierarchy, we defined the system's router ID (RID) and the local AS number. Optionally, you can configure the system's local AS number under the global BGP hierarchy for a specific BGP group, or, for a specific BGP neighbor, use the **`local-as`** configuration option. When the AS number is configured at multiple hierarchy levels, the AS number specified at the most specific hierarchy level is used. The ability to specify different AS numbers at different hierarchy levels can be quite useful, especially when merging networks with different AS numbers.

Because we are using loopback-based peering for the internal BGP group, we must reference loopback addresses in the related BGP configuration. In this case, the `neighbor` address is the remote peer's loopback address. The `local-address` is the local device's loopback address. If the local address is not specified, the system uses the interface address of the egress interface used to reach the referenced peer address. Because the peer is expecting to form an IBGP peering session using the 192.168.100.1 address, you must specify that address as the **`local-address`** in the configuration.

As mentioned on the slide, the session type determines whether the peering session is IBGP or EBGP. You specify an **`external`** session type for EBGP and an **`internal`** session type for IBGP. If you omit the session type, you must specify the **`peer-as`** number, which can be a remote AS number or the local AS number. If the specified AS number does not match the AS number defined on the router, BGP assumes the session type is external. If the specified AS number does match the AS number defined on the router, BGP assumes the session type is internal. The software notifies you if you did not include the required details.

Configuring BGP Authentication

- BGP protocol exchanges can be MD5 authenticated

```
[edit]
user@router# show protocols bgp
authentication-key "$9$hrHyeWNdsJGiLxGik.zFcyl"; ## SECRET-DATA
group int-65503 {
    type internal;
    local-address 192.168.100.1;
    authentication-key "$9$LkW7dsaZjP5F245Fn/0OX7-"; ## SECRET-DATA
    neighbor 192.168.100.2;
}
group ext-65501 {
    type external;
    peer-as 65501;
    neighbor 172.30.1.2 {
        authentication-key "$9$m5z6p0IreW9AeWLxwsP5Q"; ## SECRET-DATA
    }
}
```

Global Level

Group Level

Neighbor Level

Be very careful when configuring authentication at the global level because this affects *all* peers, including EBGP peers that might use a different key from your internal peers (or none at all).

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Authentication

All BGP protocol exchanges can be authenticated to guarantee that only trusted routing devices participate in the AS's routing. By default, authentication is disabled. You can configure MD5 authentication. The MD5 algorithm creates an encoded checksum that is included in the transmitted packet. The receiving routing device uses an authentication key (password) to verify the packet's MD5 checksum.

Configuring BGP Authentication

- Hitless key rollover is an option

```
[edit security]
user@router# show
authentication-key-chains {
    key-chain key-chain-name {
        key 1 {
            secret secret-data;
            start-time yyyy-mm-dd.hh:mm:ss;
        }
        key 2 {
            secret secret-data;
            start-time yyyy-mm-dd.hh:mm:ss;
        }
    }
}
[edit protocols bgp]
user@router# show
group int-65503 {
    type internal;
    local-address 192.168.100.1;
    authentication-key-chain key-chain-name;
    neighbor 192.168.100.2;
}
```

The range of valid key identifier values is from 0 through 63 and each `key` must use a unique value. Furthermore, each `key` *must* contain a `secret` and `start-time` value. The `start-time` values are based on the configured time zone and, again, must be unique within the same keychain.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Hitless Key Rollover

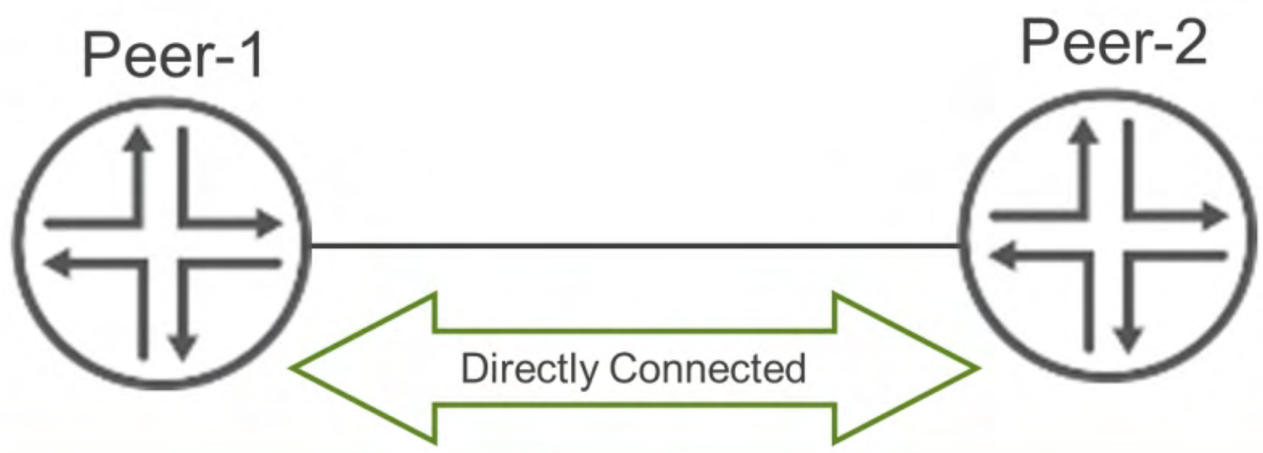
Hitless authentication key rollover allows users to choose the algorithm through which authentication is established. The user associates a keychain and an authentication algorithm with a BGP neighbor session. The keychain can include multiple keys. Each key contains an identifier and a secret. The key is also configured with a unique start time and an end time.

To configure the authentication key, include the `key-chain` statement at the `[edit security authentication-key-chains]` hierarchy level, and specify the `key` option to create a key chain consisting of several authentication keys. You then reference this keychain name under the appropriate BGP level. In the example, the keychain is configured at the group level.

As time is used to change keys, it is important that devices participating in key-chains are synchronized using NTP to prevent the changing of keys causing an outage.

Another Option for Secure Peering

- BGP Generalized TTL Security Mechanism
  - Used in BGP single-hop
  - Requires that BGP session is only established with a directly connected router
  - Drops any BGP packets that do not have a maximum value TTL
  - Helps protect against DoS attacks



© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP GTSM

As BGP MD5 was realized to be largely underutilized, another method was developed for securing a BGP session and ensuring that is was not hijacked by a third party. This method is BGP Generalized Security Mechanism (GTSM).

When using GTSM, a BGP router sets the TTL of BGP packets to 255. The receiving router then checks if the incoming packets have a TTL of 255 and drops the packets if they don't. Because routers decrement the TTL when forwarding packets, and 255 is the maximum value, it's impossible for an attacker several hops away to send packets that arrive with a TTL of 255. This feature was designed to replace the need to exchange passwords or perform complex calculations.

GTSM is also designed to protect a router's IP-based control plane from DoS attacks. Many attacks focused on CPU load and line-card overload can be prevented by implementing GTSM on all Exterior Border Gateway Protocol speaking routers. GTSM is based on the fact that the vast majority of control plane peering is established between adjacent routers; that is, the Exterior Border Gateway Protocol peers are either between connecting interfaces or between loopback interfaces. Since TTL spoofing is considered nearly impossible, a mechanism based on an expected TTL value provides a simple and reasonably robust defense from infrastructure attacks based on forged control plane traffic.

An example of GTSM configuration is found on the next page.

GTSM Configuration

- Example of GTSM Configuration

Peer–1

```
user@BGP-Peer-1# show firewall
filter ttl-security {
   term gtsm {
      from {
         source-address {
            10.1.2.1/32;
         }
         protocol tcp;
         ttl-except 255;
         port 179;
      }
      then {
         discard;
      }
   }
   term else {
      then {
         accept;
```

Peer-1

```
user@BGP-Peer-1# show interfaces
ge-1/0/0 {
   unit 0 {
      family inet {
         address 10.1.2.2/30;
            filter {
               input gtsm;
```

Peer-2

```
user@BGP-Peer-1#  show protocols
bgp {
   group toAS2 {
      type external;
      peer-as 2;
      ttl 255;
      neighbor 10.1.2.2;
   }
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Example of GTSM Configuration

On one side of the BGP peering a firewall is created that identifies the neighbor and ensures the BGP packets arrives with a TTL of 255. This firewall filter is applied on the incoming interface.

The other BGP peer sets the **`ttl`** option to 255 so all packets are sent with the maximum TTL In this environment only BGP traffic from a directly connected neighbor will be accepted.

Preserving State using Graceful Restart

- A BGP router can negotiate with peers to maintain its announced routes during a restart
  - Neighbors continue forwarding traffic to the router
  - Neighbors do not withdraw the routes of the restarting router
- End-of-RIB markers sent for each NLRI
  - Notifies the neighbor that all current routing information was sent
  - Local router defers path selection algorithm until the marker is received
- Configured globally within the `[edit routing-options]` hierarchy
- Disabled locally per protocol
- Connected devices must also support Graceful Restart

© 2020 Juniper Networks, Inc. All Rights Reserved.

Preserving State using Graceful Restart

When a BGP router supports graceful restart, it can restart its routing process without causing a topology change event in the network. Thus, no routes are withdrawn or announced based on the restart event, which can avoid the network converging on a new topology.

During the establishment of the peering session, the two routers negotiate the ability to support graceful restart. The restart mechanism is only used when the feature is supported by both peers. When the local router restarts its routing process its peer continues to forward traffic to it. In addition, the loss of keepalive messages from the local router does not cause the peer to withdraw the active routes from the restarting router.

The restarting router signals its peers that it has restarted by setting the restart state bit in the capability announcement within the open message. The signal alerts the peer to retain the restarting router's routes and not run the path selection algorithm. In addition, the restarting router might also set the forwarding state bit in the capability announcement as well to alert the peer that it might still forward user traffic through the restarting router.

After the restarting router returns to service and the session is established again, the peers exchange their active routes with each other. When the last route is sent, an *end-of-RIB* marker is advertised to inform the peer that the process is complete. This marker is simply an update message with no withdrawn or advertised routes. The receipt of the end-of-RIB marker allows each router to run the path selection algorithm and select new active routes, if necessary.

Configuration Options

The graceful restart capability within the Junos OS is activated under the `[edit routing-options]` configuration hierarchy using the `graceful-restart` option.

The `graceful-restart` syntax is also a configuration hierarchy within BGP, which contains three options specific to the operation within the protocol. Those options include the following:

- **`disable`**: This option stops the local router from participating in any graceful restart function.
- **`restart-time`**: This negotiable timer sets the amount of time that can elapse for the peering session to reestablish. The default value for this timer is 120 seconds, and its range is between 1 and 600 seconds.
- **`stale-routes-time`**: This timer specifies the amount of time that routes from the restarting peer can be used before they are withdrawn. The default value for this timer is 300 seconds, and its range is between 1 and 600 seconds.

Peer Configuration Options (1 of 3)

- **`passive`** keeps BGP from sending an open message

```
[edit protocols bgp]
group ext-peers {
    type external;
    peer-as 2;
    neighbor 10.10.10.1 {
        passive;
    }
}
```

- **`allow`** accepts open messages from any peer within the configured IP address range

```
[edit protocols bgp]
group ext-peers {
    type external;
    peer-as 65000;
    allow 10.10/16;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

`passive` Option

By default, a local router initiates a BGP open message to the remote router to establish the session. The **`passive`** command stops this default action, and no open message is sent. The IP address and AS of the remote peer are still configured, but the remote router must initiate the BGP session.

`allow` Option

The related option of **`allow`** also stops the sending of a BGP open message to the remote router. In addition, the **`allow`** command also relaxes the requirement of explicitly configuring the remote router's IP address by allowing you to define a subnet range for connections. BGP processes any open message received from an IP address within the configured range and initiates a session with that remote router.

Peer Configuration Options (2 of 3)

- **prefix-limit** allows a specified amount of prefixes to be received
- Configured under family xxxx to limit prefixes

```
[edit protocols bgp]
group ext-peers {
    type external;
    peer-as 2;
    family inet {
        unicast {
            prefix-limit {
                maximum 25000;
                teardown 80 idle-timeout 10;
```

- **hold-time** alters the keepalive time used to maintain the BGP session
  - Keepalive value is 1/3 of configured hold-time value

```
[edit protocols bgp]
group ext-peers {
    type external;
    hold-time 45;
    peer-as 2;
    neighbor 10.10.10.1;
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Limiting the Number of Prefixes Accepted

By default, each BGP router accepts any number of routes sent from each of its peers. You might want to alter this default for either security or memory reasons. You can use the **prefix-limit** command to set a limit on the maximum number of routes received from any individual peer. Use the **maximum** option to set a number of routes that if exceeded will result in a message with peer address, address family and instance name being logged without tearing down the session.

Including the **teardown** option results the peering session is terminated when a BGP peer sends more routes than allowed, After the session is torn down, it is reestablished in a short time unless you include the **idle-timeout** statement. The **idle-timeout** value (entered in minutes) keeps the peering relationship from rebuilding for up to 40 hours. Using an **idle-timeout** value of **forever** will keep it down indefinitely. This value requires you to intervene manually to restart the peering session.

The value that immediately follows the **teardown** option is a percentage. This represents the percentage of the maximum configured earlier where the system will start to generate system log messages

Altering the Session Hold Time

When two BGP peers establish their peering session, they negotiate the hold time for that relationship. By default, the Junos OS uses a value of 90 seconds to negotiate for the hold time of the session. The **hold-time** command allows you to alter this value from as short as 20 seconds to as long as 65,535 seconds (18 hours, 12 minutes, and 15 seconds).

Peer Configuration Options (3 of 3)

- By default, Junos OS does not advertise the routes learned from an External BGP peer to another EBGP peer if that peers AS number appears in the AS-path
- This forwarding suppression is not the default for all vendors
- The **advertise-peer-as** statement overrides the default action on Junos BGP routers

```
[edit protocols bgp]
group ext-peers {
    advertise-peer-as;
    peer-as 2;
    neighbor 10.10.10.1;
}
```

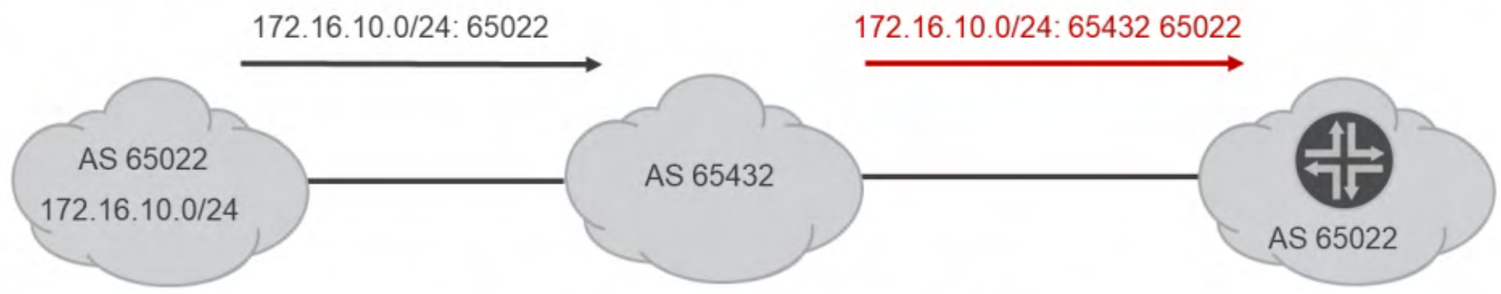
© 2020 Juniper Networks, Inc. All Rights Reserved.

Disabling Suppression of Route Advertisements

By default, the Junos OS does not advertise the routes learned from an EBGP peer to another EBGP peer if that peers AS number appears in the AS-path. You can modify the default behavior with the **advertise-peer-as** command.

Modifying AS Path: `loops`

- Required to accept routes with your own AS in the AS-path




```
user@AS65022> show route receive-protocol bgp 10.222.4.1

inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

[edit]
user@AS65022# set routing-options autonomous-system 65022 loops 2

user@AS65022> show route receive-protocol bgp 10.222.4.1

inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 172.16.10.0/24          10.222.4.1                              65432 65022 I
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Allowing AS Paths Loops

The Junos OS allows you to configure your router to accept an AS path loop in certain situations. The slide once again shows AS 65432 providing transit service to AS 65022. As before, the 172.16.10.0/24 route is not accepted by the router in AS 65022 because of an AS path loop.

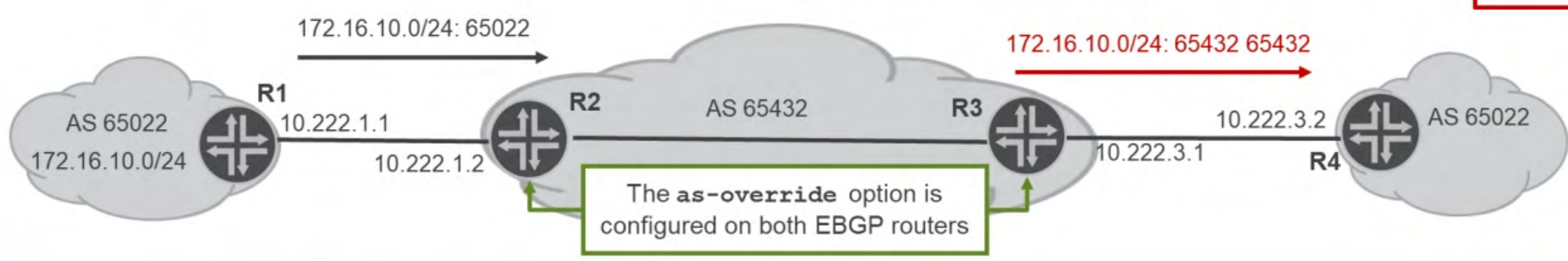
The configuration option used in this example is performed on the router in AS 65022 itself. The optional keyword **`loops`** is appended to the **`autonomous-system`** command within the `[edit routing-options]` configuration hierarchy. This keyword allows the local AS number to appear multiple times in the path. The route can then be received by the router in AS 65022.

This scenario also requires the EBGP peer in AS 65432 to be configured with the **`advertise-peer-as`** command. Otherwise, routes learned from one instance of AS 65022 will not be advertised to the second instance of AS 65022.

Modifying AS Path: as-override

- An alternative to advertise-peer-as
- Changes the peers AS number in the AS path to match the AS number of the sending AS

```
user@R2> show route receive-protocol bgp 10.222.1.1

inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 172.16.10.0/24          10.222.1.1                              65022 I

user@R3> show route advertising-protocol bgp 10.222.3.2
[no output]

user@R4> show route receive-protocol bgp 10.222.3.1
inet.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

[edit]
user@R3# set protocols bgp group AS-65022 as-override

user@R3> show route advertising-protocol bgp 10.222.3.2
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 172.16.10.0/24          Self                                    65432 I

user@AS65022> show route receive-protocol bgp 10.222.3.1
inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 172.16.10.0/24          10.222.3.1                              65432 65432 I
```



© 2020 Juniper Networks, Inc. All Rights Reserved.

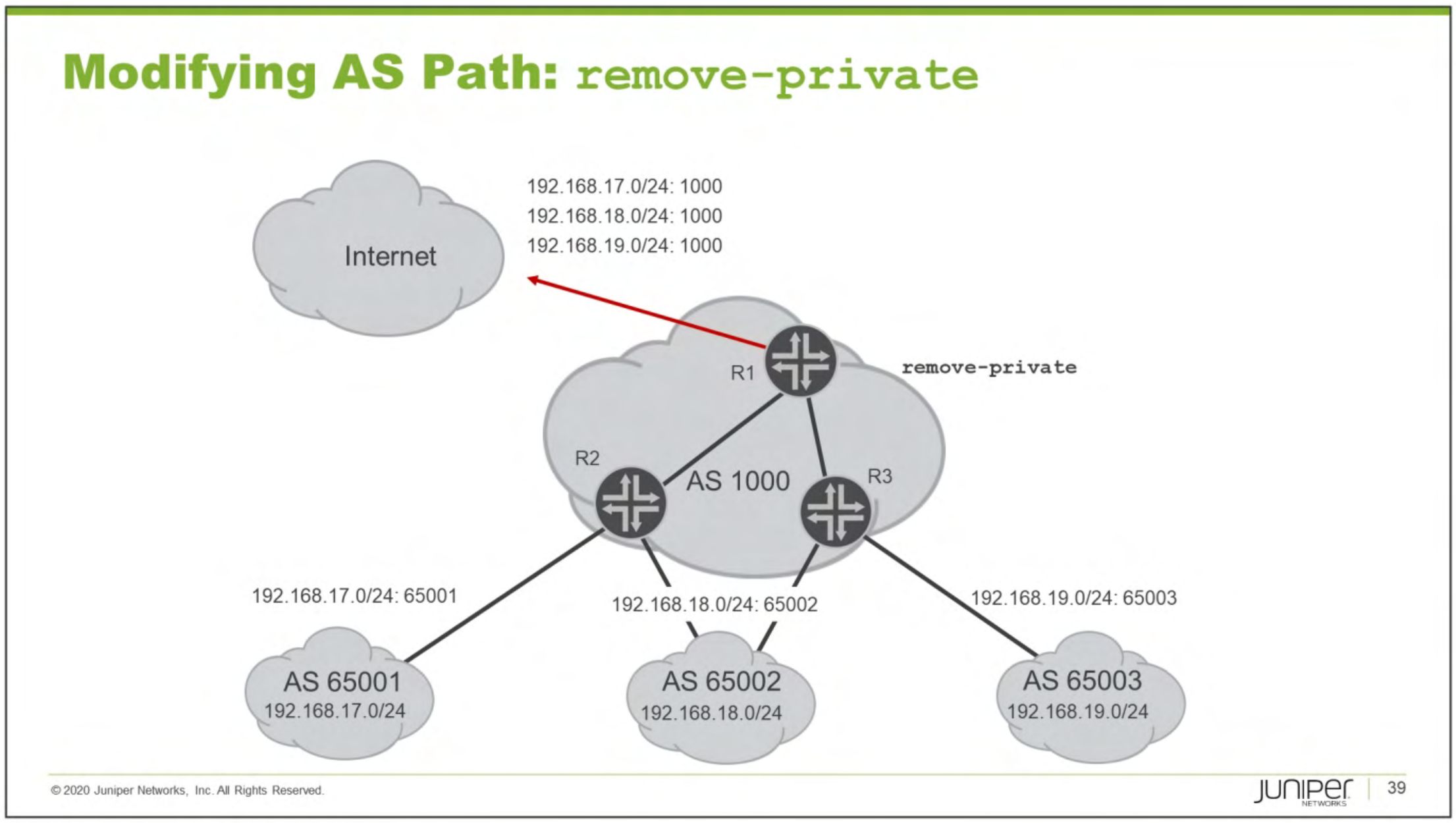
Overriding the Default Prepend Action

In certain situations, the default mechanics of the AS path prepend mechanism might cause routes to not be received by a peer. One such situation is displayed on the slide.

AS 65432 is providing transit service to AS 65022, which peers at two different locations. For reasons that we do not discuss in this chapter, the two portions of AS 65022 do not have any other peering links between them. In fact, these two sections of the network rely on AS 65432 for reachability information to the other half of the AS. By default, the R3 router does not advertise 172.16.10.0/24 route to AS 65022. After all, AS 65022 is already in the AS path.

Assume both EBGP peers in AS 65432 alter their configuration with their peers in AS 65022 to include the **as-override** command. This command allows the routers in AS 65432 to examine the AS path before advertising the route and look for any instances of AS 65022 already in the path. Should it find any, they are replaced with the peer's own AS, AS 65432 in this case. The R3 peer then performs the default prepend action before advertising the route to R4. The R4 router in AS 65022 now receives the route without an AS path loop and installs it in its local routing table.

GR helper mode is enabled by default on all Junos devices.



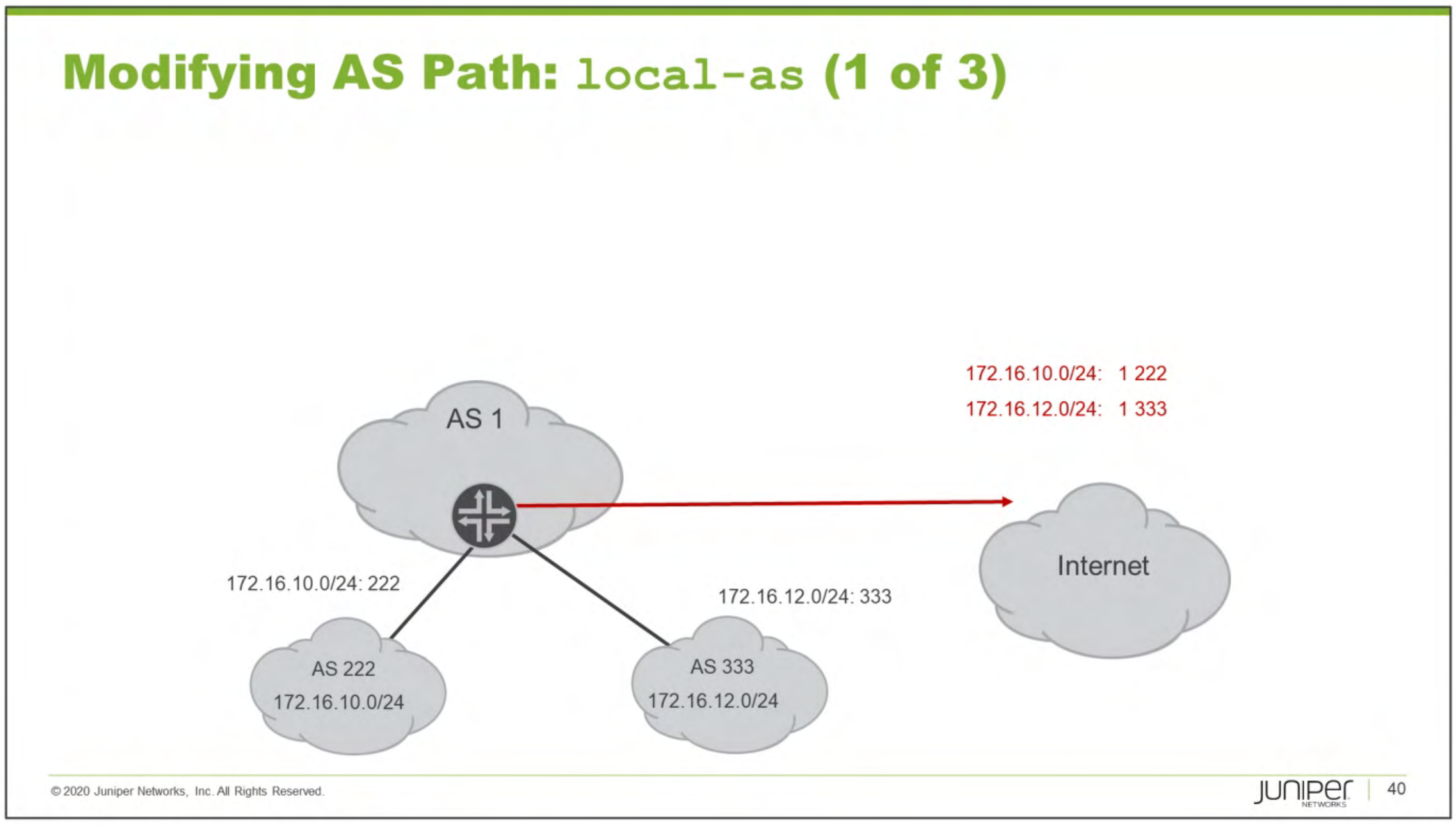
Eliminating Private AS Numbers

In spite of the wording in the BGP RFC, many vendors include configuration options in their BGP implementations that remove information from the AS path, which is technically not allowed. This removal, however, only operates on specific information in the AS path attribute, and it does not apply to making arbitrary changes to the actual AS path. Typically, the information removed was placed there by the AS itself or by other routers within the administrative control of the AS. Thus, one AS is not trampling on the path information another AS has put into the AS path attribute.

One example of this type of configuration option is the **`remove-private`** configuration statement. This keyword allows an ISP to remove private AS numbers from paths received from BGP customers when those customers are using private AS numbers. Because the customers are effectively within the administrative scope of the ISP, the provider is allowed to remove the private AS numbers from the path.

On the slide, AS 1000 has three different customers connected using BGP. The customers are using AS 65001, AS 65002, and AS 65003 for the BGP peer communications. Within AS 1000, each of the BGP routers sees the private AS numbers within the path.

The **`remove-private`** option is enabled on the R1 router in AS 1000. As BGP advertises the customer routes out of AS 1000, the private AS numbers are removed from the AS path attribute. In this case, BGP views all customer networks as having originated within AS 1000.

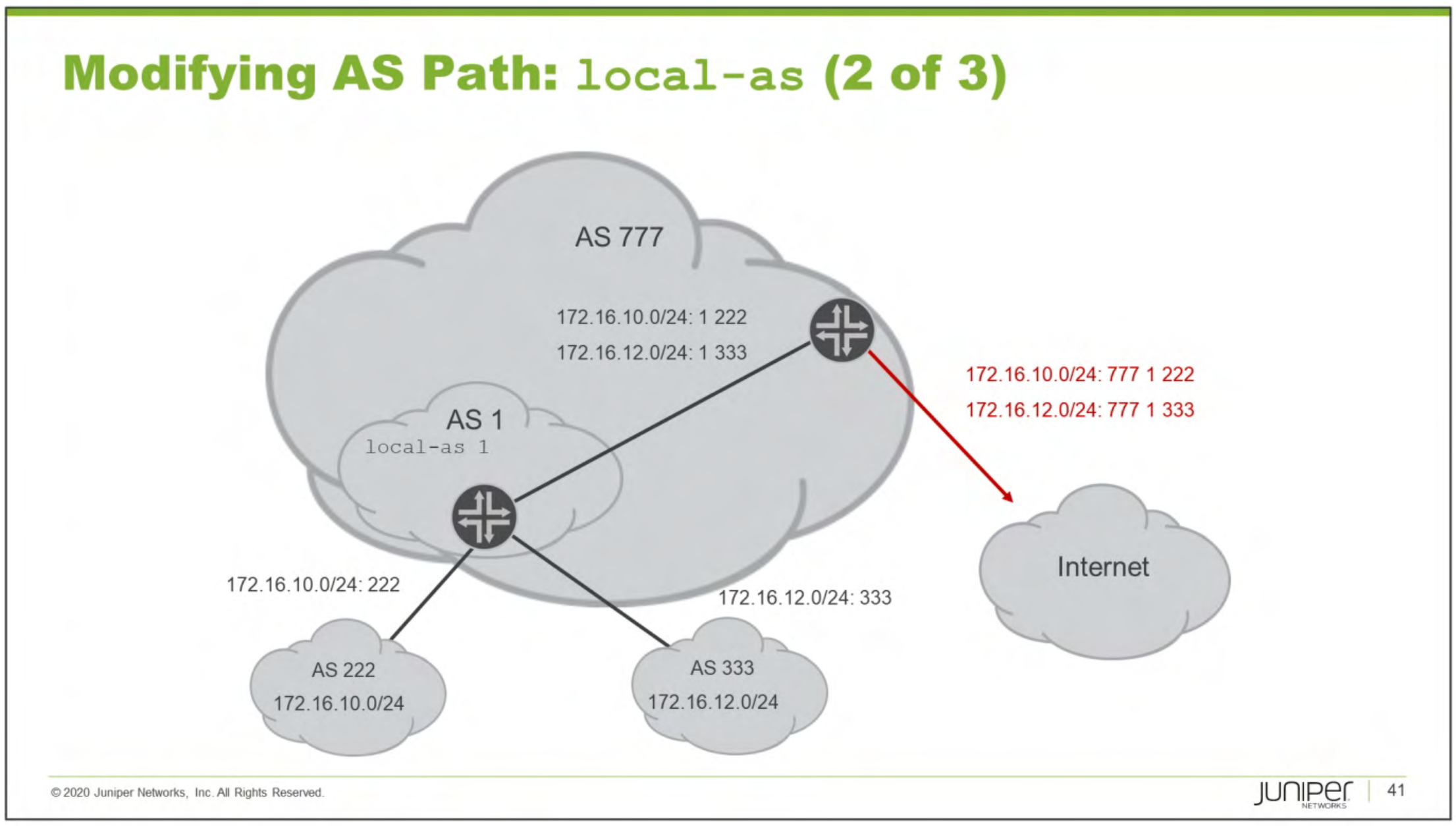


Modifying the AS Path Attribute, Part 1

Another option for removing AS path attribute information is the **`local-as`** configuration statement. The purpose of the **`local-as`** keyword is to aid an ISP in migrating BGP customers to a new AS number. The following slides display an example of how you can use the **`local-as`** option.

Consider the normal BGP AS path operation first. AS 1 has two customers for which it provides service: AS 222 and AS 333. As AS 1 announces these routes from AS 222 and AS 333 on to the Internet, AS 1 injects its own AS number (1) into the AS path attribute, as the BGP RFC expects.

So far, nothing is unusual about the AS path operation.



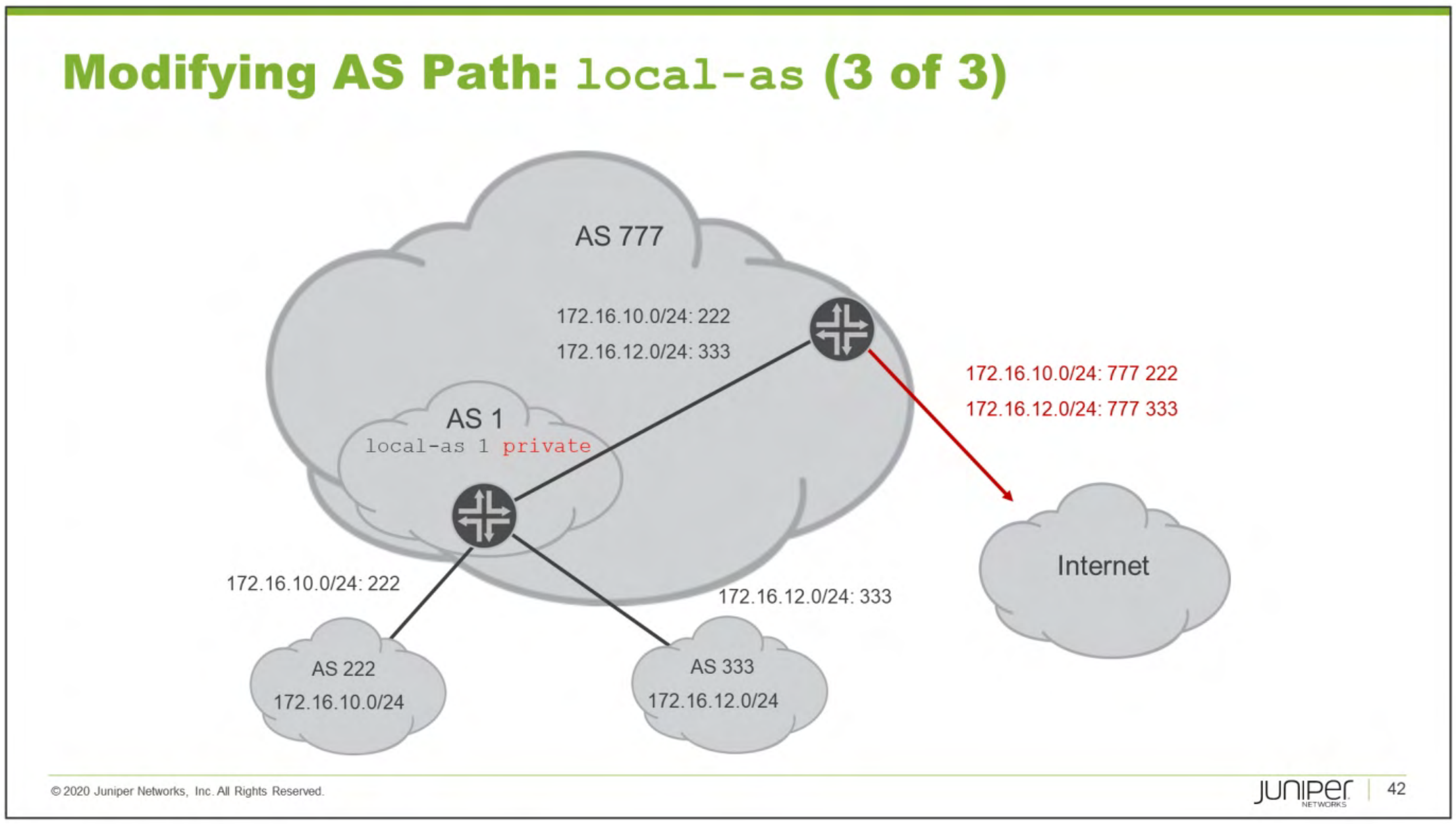
Modifying the AS Path Attribute, Part 2

Next, consider what happens if AS 1 merges with AS 777. This situation is shown on the slide.

Suppose the resulting merged organization decides to use AS 777 as the official AS to represent both networks on the Internet. To ease in the migration of the customer BGP configurations from AS 1 to AS 777, the edge routers can use the **`local-as 1`** configuration option.

The effect of this option is that the customer routes within AS 777 see *both* AS 1 and the customer AS numbers (AS 222 and AS 333). As AS 777 advertises those routes to the Internet, it prepends its own value of AS 777 onto each of the routes.

Therefore, even though AS 1 has been merged into AS 777, AS 1 still shows up on the paths sent to the Internet.



Modifying the AS Path Attribute, Part 3

You can use an optional parameter with the **`local-as`** configuration statement that actually does remove AS path information from the BGP AS path attribute. This optional parameter restricts knowledge of AS 1 to the edge router connected to the customer (AS 222 and AS 333) only. This situation is shown on the slide.

On the slide, the edge router in the formerly intact AS 1 is configured with the option of **`local-as 1 private`**. As the edge router advertises the customer routes into AS 777, AS 1 information is removed from the AS path attribute, as shown on the slide. At this point, the AS 777 routers, as well as the Internet, have no knowledge of AS 1.

The **`local-as 1 private`** statement now has indeed removed AS path information. Again, this example applies to a type of special case and should not be used arbitrarily to attempt to change AS path information received from another AS.

Other options are the following:

- **`local-as`** ***`autonomous-system`*** **`alias`**: A BGP peer considers any local AS to which it is assigned as equivalent to the primary AS number configured for the routing device. When you use the **`alias`** option, only the AS (global or local) used to establish the BGP session is prepended in the AS path sent to the BGP neighbor.
- **`local-as loops`** ***`number`***: Specify the maximum number of times that the local AS number can appear in an AS path received from a BGP peer. For number, include a value from 1 through 10.

Summary

- In this content, we:
  - Described basic BGP operation
  - Listed common BGP attributes
  - Explained the route selection process for BGP
  - Described how to alter the route selection process
  - Configured some advanced options for BGP peers

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- Basic BGP operations;
- Listed common BGP attributes;
- Explained the route selection process for BGP;
- Described how to alter the route selection process; and
- Configured some advanced options for BGP peers.

Review Questions

1. What are the advantages of loopback peering for IBGP sessions?
2. What are the default BGP route advertising rules?
3. How does multipath affect route selection?
4. What are valid ways for BGP to resolve a next hop?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

3.

4.

Lab: Configuring BGP

- Configure and monitor IBGP and EBGP.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: Configuring BGP

The slide provides the objective for this lab.

Answers to Review Questions

1.

Using loopback addresses for peering sessions between IBGP neighbors maintains reachability even when the physical topology changes as long as a viable path exists.

2.

EBGP-learned routes are advertised to both EBGP and IBGP peers. IBGP-learned routes are advertised to EBGP peers. However, IBGP-learned routes are not advertised to other IBGP peers to prevent routing loops.

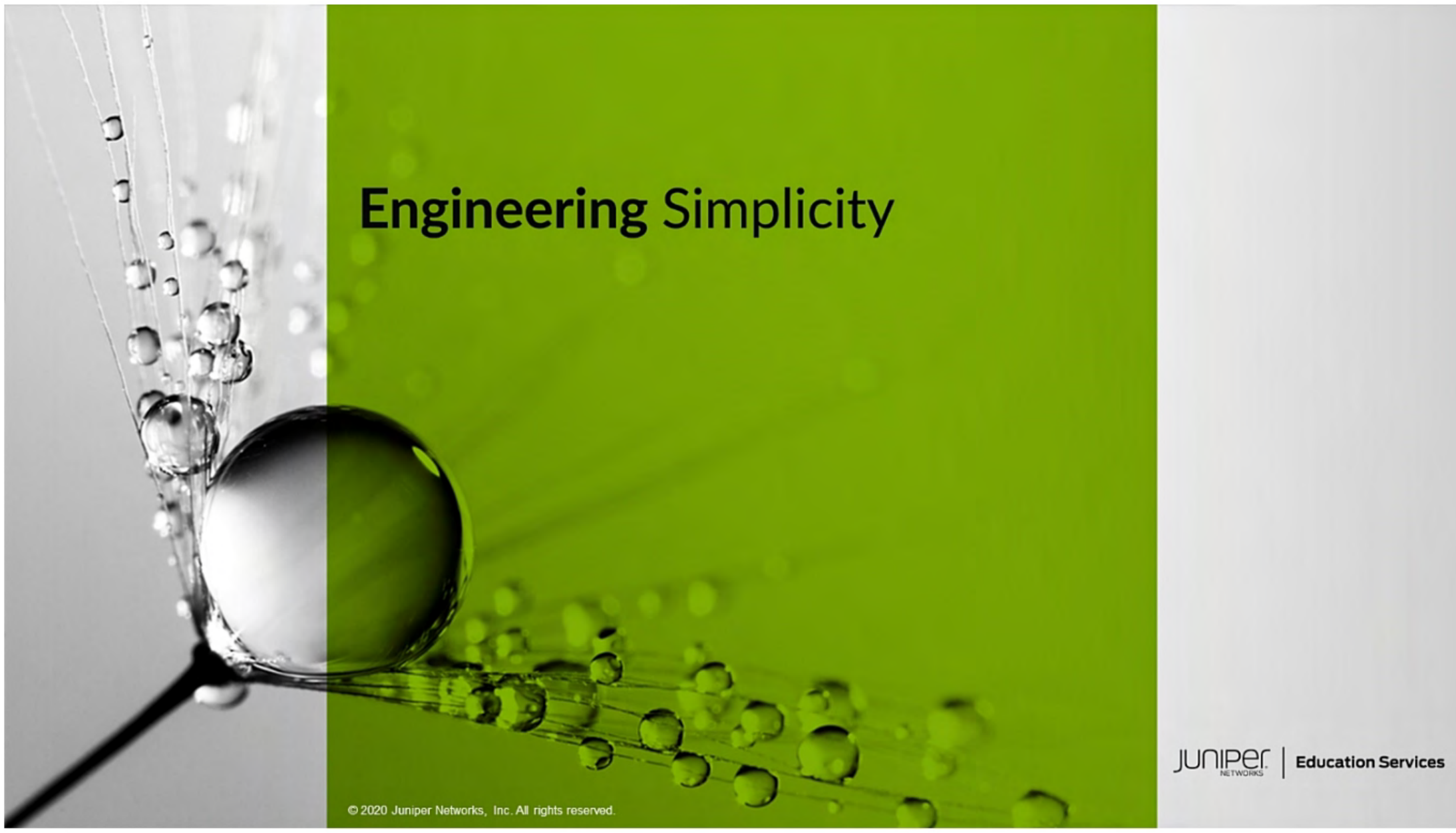
3.

When you configure multipath on a BGP router, the route selection algorithm ignores both the router ID and the peer ID selection criteria.

4.

There are multiple approaches that can solve the BGP next hop issue:

- Next-hop self;
- IGP passive interfaces;
- Export direct routes into IGP; and
- Static routes


Engineering Simplicity
© 2020 Juniper Networks, Inc. All rights reserved.
Juniper Networks
Education Services

Education Services

Advanced Junos Service Provider Routing

Chapter 11: BGP Attributes and Policy—Part 1

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - Explain how policies function in BGP
  - Describe BGP attributes and explain how these attributes can be used to manipulate traffic
  - Show how Regex can be used in policies to manipulate AS-path

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- Explain how policies function in BGP;
- Describe BGP attributes and explain how those attributes can be used to manipulate traffic; and
- Show how Regex can be used in policies to manipulate AS-path.

Agenda: BGP Attributes and Policy—Part 1

→BGP Policy Overview
- BGP Attributes Overview
- Local Preference
- AS Path

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Policy Overview

The slide lists the topics we will discuss. We discuss the highlighted topic first.

BGP Policy

- BGP behavior can be influenced by policy
  - BGP attributes can be matched or changed using import or export policies
  - Can differentiate between IBGP or EBGP routes
- BGP stores routes in three main RIB memory tables
  - RIB-IN: Stores all received routes
  - RIB-LOCAL: Stores routes the local router uses to forward traffic
  - RIB-OUT: Stores all advertised routes
- Only active BGP routes in the local routing table are advertised to peers
  - Single best BGP path is advertised
  - An overshadowed BGP route can be advertised if the `advertise-inactive` option is configured
  - More than one BGP route can be advertised using the add-path option

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP and Policy Dependency

BGP is a very policy-oriented protocol, in the sense that import and export policies largely determine BGP behavior. The router can use all the BGP attributes. You can adjust these attributes to influence the behavior of the local router as well as other routers receiving the route.

You can use each of the BGP attributes as a match criterion for a policy, and you can modify each attribute using a policy action. You can further decide to act on a BGP attribute based on whether it is an EBGP or IBGP route. To understand better where BGP import and export policies are applied to BGP routes, we detail the process of how a router uses BGP routes.

BGP RIB Tables

BGP uses three different storage tables, known as routing information bases (RIBs), as *databases* to maintain routing knowledge. A separate RIB-IN exists for each established BGP peer to store all routes received from that peer. A RIB-LOCAL is where BGP stores routes used for traffic forwarding. A separate RIB-OUT exists for each established BGP peer to store routes to be advertised to that peer.

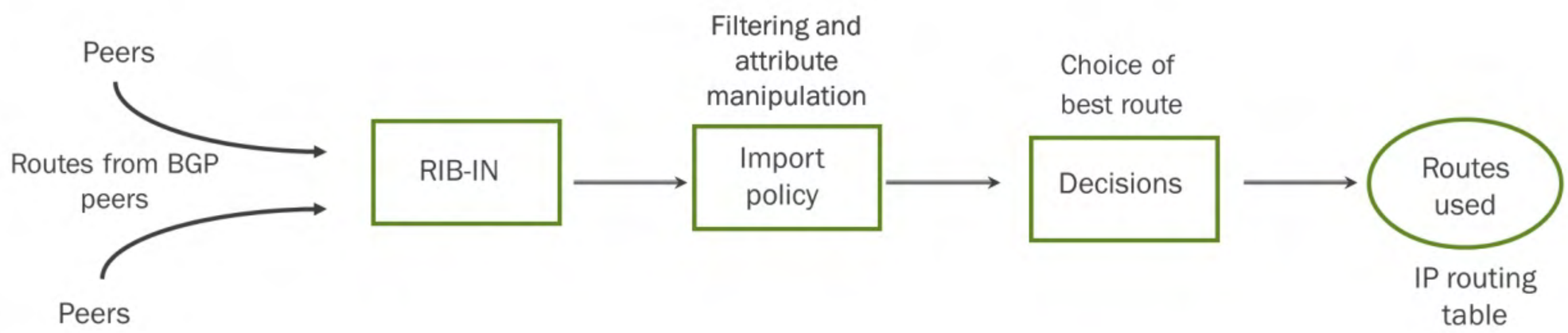
BGP and Active Routes

Only active BGP routes in the routing table can be moved into the RIB-OUT tables and are advertised to BGP peers. In addition, only the single best BGP path to each separate IP route destination is placed in the RIB-LOCAL and RIB-OUT tables.

At times, it is possible that the best BGP path is not advertised to a peer because of the local router's routing table rules. For example, if the router knows about a particular route through both IS-IS and BGP, the IS-IS route will be active in the local routing table because of the default Junos OS route preference values. Therefore, the BGP version of that route will not be sent to any peers because BGP advertises only active routes (routes used by BGP). To override this default action, you can use the **`advertise-inactive`** command. This command always forces the advertisement of the single best BGP path to any destination, regardless of whether the route is currently active in the local routing table.

BGP Import Policy

- Import policies are enforced between the RIB-IN and RIB-LOCAL tables



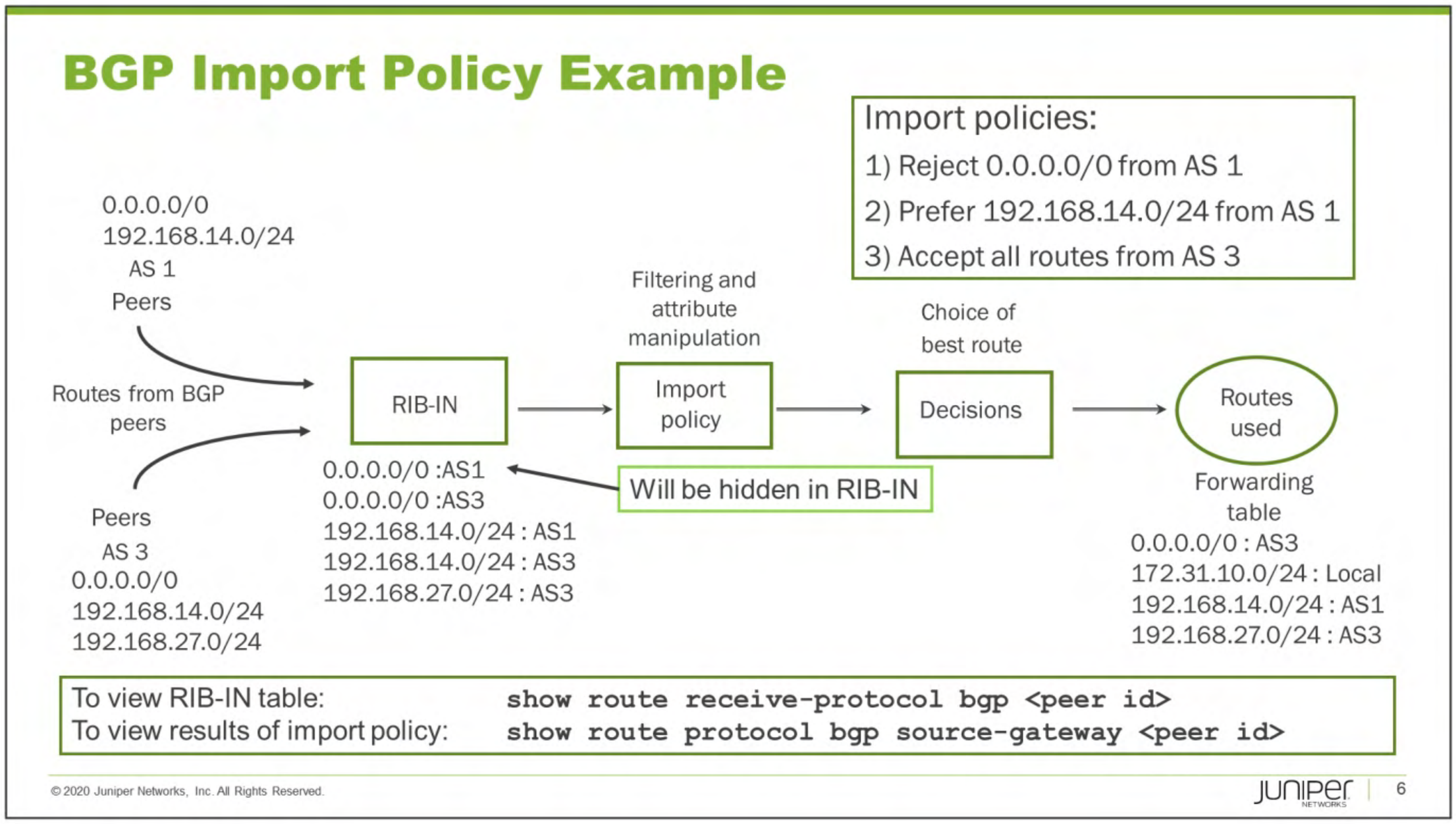
© 2020 Juniper Networks, Inc. All Rights Reserved.

Import Policies and BGP

BGP stores the information about routes received from BGP peers in the RIB-IN table. No policies are applied yet; all BGP routing information that is received is stored in this table except routes that fail AS path or cluster-ID sanity checks, as well as VPN routes that do not have a matching target community.

As BGP moves the routes that it received from peers to the RIB-LOCAL table, the Junos routing policy framework can apply import policies. These policies can reject (filter) routes or can change attributes and affect what the BGP route selection process uses to pick the best route.

After the BGP import policy or policies (if any are configured and applied) has filtered and manipulated the BGP attributes, the BGP decision process chooses the best route to use and installs that route into the IP routing table. Note that even if no routing policies are configured, the default (and unseen) BGP import policy is always applied.



Sample BGP Import Policy

The slide shows an example of some BGP import policies in action.

Some policies are configured that reject the default route (0/0) if received from AS 1, prefer the AS 1 version of 192.168.14.0/24, and accept all other routes from AS 3. These three policies, which might be three separate policies or a single policy with three terms, are configured as import policies for BGP.

Import policy is evaluated as the BGP process attempts to move routes from the RIB-IN table to the BGP decision process, where the active route is selected.

The result of this policy example is that the forwarding table correctly reflects the user's desire to forward through AS 1 for traffic matching the 192.168.14.0/24 prefix and AS 3 for traffic matching the 0/0 default route. Note that the routing table (not shown) will contain two entries for the 192.168.14.0/24 prefix, because the 192.168.14.0/24 prefix coming from AS 3 is not filtered in this example. The AS 1 version of this prefix is preferred and is installed in the forwarding table as a result. The following list summarizes the overall effects of this policy example:

- The 0.0.0.0/0:AS1 route is *rejected*.
- The 192.168.14.0/24 route from AS3 is *accepted but not preferred*.
- The 192.168.14.0/24 route from AS1 is *accepted and preferred*.
- All other AS3 routes are *accepted*.

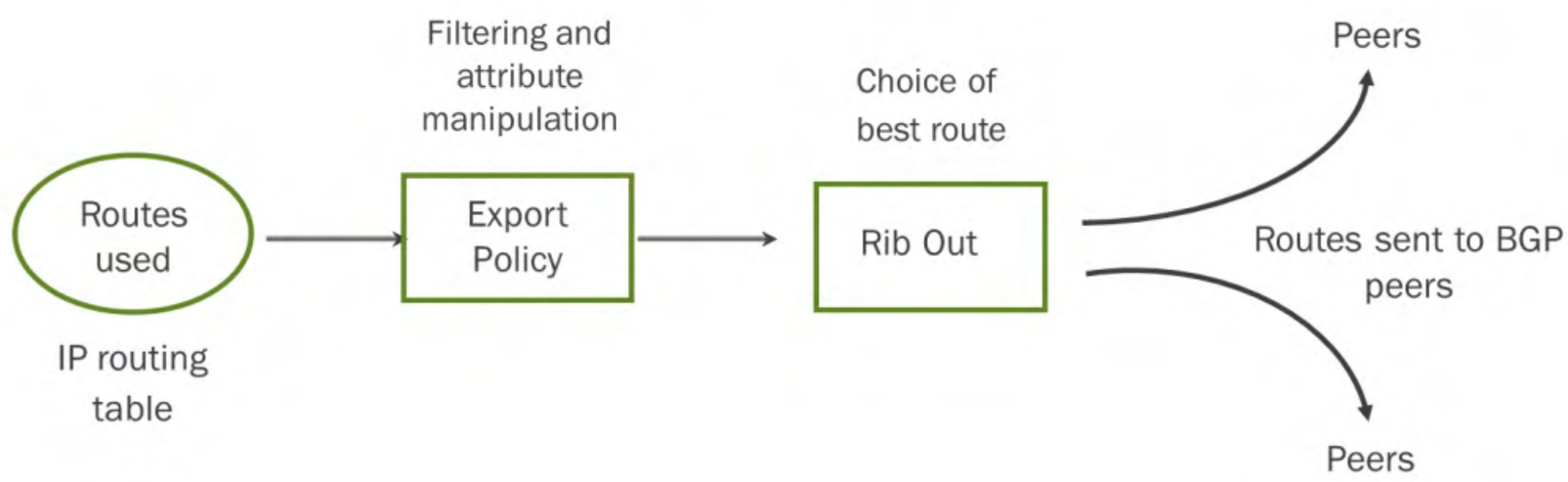
While import policies are processed after the routes have been accepted into the RIB-IN table, if the policy drops the route, the route will be hidden in the RIB-IN table. You can view the hidden routes in the RIB-IN table using the Junos operational command: `show route receive-protocol bgp <peer-ip> hidden`.

To view the routes in the RIB-IN table use the Junos operational command: `show route receive-protocol bgp <peer-ip>`.

To view the results of the import policy you need to look in the inet.0 routing table. To view the BGP routes received from a peer that are in the route table use the Junos operational command: `show route protocol bgp source-gateway <peer-ip>`.

BGP Export Policy

- Export policies are enforced between the RIB-LOCAL and RIB-OUT tables



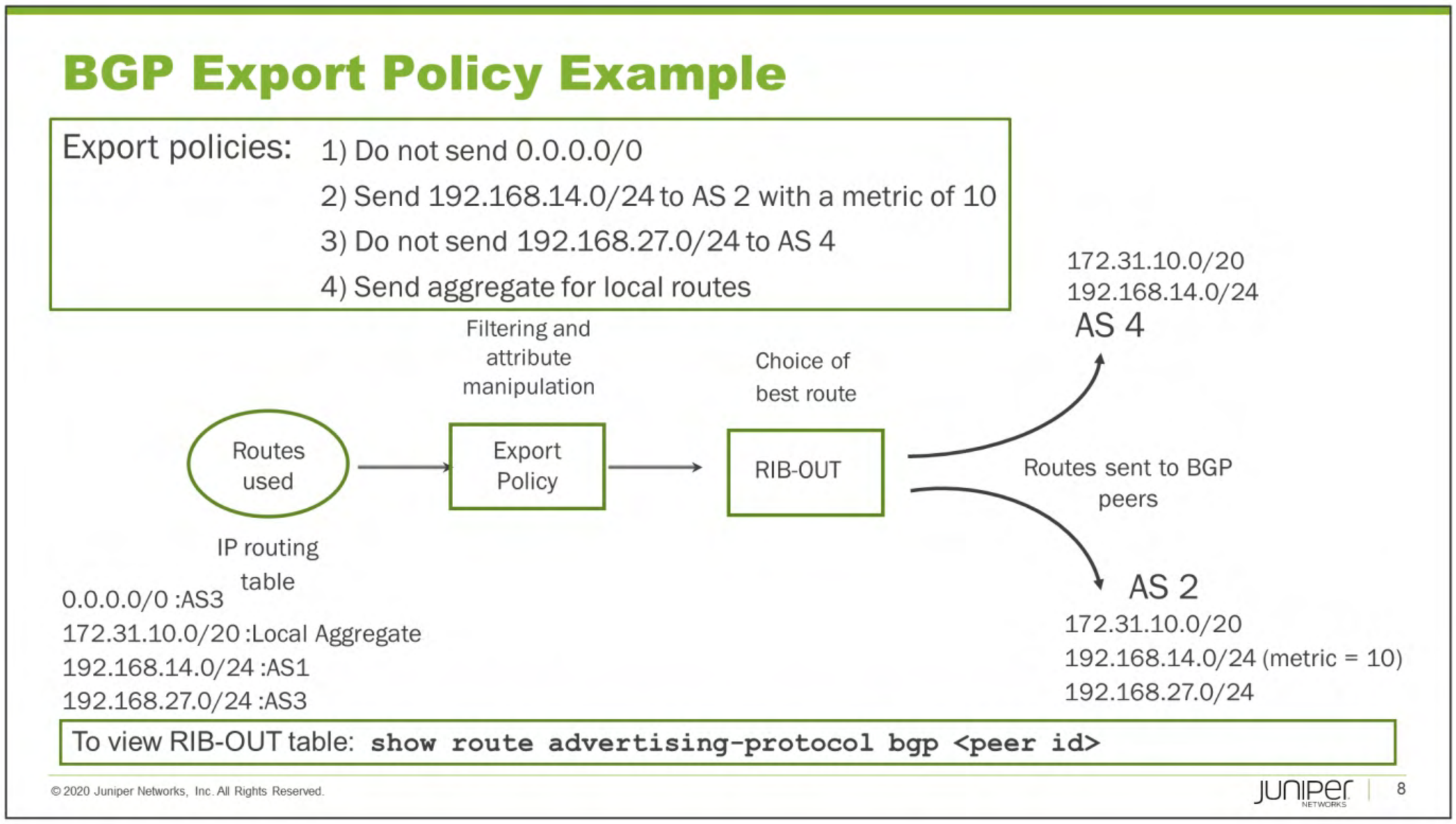
© 2020 Juniper Networks, Inc. All Rights Reserved.

Export Policies and BGP

BGP stores the information about routes to be advertised to BGP peers in the RIB-OUT table.

As BGP moves the routes from the RIB-LOCAL table to the RIB-OUT table, the Junos routing policy framework can apply export policies. These policies can reject (filter) routes and affect which BGP routes are advertised to BGP peers or can change the BGP attributes of advertised routes.

In addition, the Junos OS can inject new routing information into the BGP routing process at this point.



Sample BGP Export Policy

The slide shows an example of some BGP export policies in action.

Some policies are configured that reject the default (0/0) route, do not send some routes to AS 4, alter a BGP attribute on routes sent to AS 2, and inject new route information into the BGP process. These four policies, which might be four separate policies or one policy with four terms, are then configured as export policies within BGP.

As the BGP routing process attempts to move those routes from the RIB-LOCAL table to the RIB-OUT tables, the Junos OS applies the export policies. The net result of this policy application is shown on the slide and summarized as follows:

- The 0.0.0.0/0:AS3 route is *rejected and is not advertised*.
- The 192.168.27.0/24:AS3 route is only *sent to AS 2*.
- The 192.168.14.0/24 route is *sent to both AS 2 (with a metric of 10) and AS 4*.
- The 172.31.10.0/20 aggregate is *sent to both AS 2 and AS 4*.

To view the routes in the RIB-OUT table use the Junos operational command: `show route advertising-protocol bgp <peer-id>`.

Agenda: BGP Attributes and Policy—Part 1

- BGP Policy Overview
- →BGP Attributes Overview
- Local Preference
- AS Path

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Attributes Overview

The slide highlights the topic we discuss next.

BGP Attributes

- Attributes are very important to the operation of the protocol
- Used to filter out undesirable advertisements from a neighbor or can be altered in an attempt to influence a routing decision of a neighbor
- Attributes must fall into one of four categories:
  - Well-known mandatory
  - Well-known discretionary
  - Optional transitive
  - Optional nontransitive

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Attributes

A BGP prefix and its set of attributes is defined as network layer reachability information (NLRI). BGP attributes are a group of parameters that characterize a BGP NLRI. Thus a BGP prefix is defined by both its IP address and its set of BGP attributes.

Attributes set BGP apart from other protocols. They give a network operator the flexibility to design a robust network and the tools to engineer a scalable routing policy.

Use in Routing Policy

Attributes, combined with the extensive features of Junos routing policy, provide the ability to build a scalable set of rules that dictate whether a network wants to filter an incoming route or influence the path of a neighbor. Junos policy can match and manipulate the more commonly used attributes, as we will see in upcoming slides.

BGP Attribute Categories

To ease the introduction of new BGP features, the BGP designers introduced a set of categories for router vendors to follow. All BGP attributes must fall into one of the four categories listed on the slide. This framework allows for new BGP features like 32-bit ASs or multiprotocol extensions with IP version 6 (IPv6), MPLS, or multicast backwards compatibility without affecting running networks. The next slide describes these categories in more detail.

BGP Attribute Categories (1 of 2)

- **Well-known mandatory:** Must be supported by all BGP speakers and must be present in all BGP updates that contain a route
- **Well-known discretionary**: Must be supported by all BGP speakers and might or might not be present in a BGP update

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Attribute Categories: Part 1—Well-Known Mandatory

An attribute defined as well-known mandatory must be on every BGP update and passed along to all peers. If an update message does not contain an attribute in this category, it causes the peering to reset and send a missing well-known attribute message.

Well-Known Discretionary

An attribute defined as well-known discretionary must be understood by every BGP speaker, but it does not have to be in every update. If an update message contains an attribute in this category and is not understood, the connection is reset and an unrecognized well-known attribute message is sent.

BGP Attribute Categories (2 of 2)

- **Optional transitive**: Optional attribute that might not be understood by all speakers and is expected to transit even if it is not understood by the local speaker
- **Optional nontransitive:** Optional attribute that might not be understood by all speakers and is expected to be quietly ignored and not passed along to other BGP peers

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Attribute Categories: Part 2—Optional Transitive

An attribute defined as optional transitive does not have to be understood by every BGP speaker, but it needs to be passed to all peers. If an unrecognized attribute such as AS4_PATH is received, it can safely be ignored, but it MUST be sent to all neighbors.

Optional Nontransitive

An attribute defined as optional nontransitive does not have to be understood by every BGP speaker, and it must not be passed to any peer. If an unrecognized attribute such as MP_REACH_NLRI is received it can safely be ignored, but it MUST NOT be sent to all neighbors.

Common BGP Attributes

- Common BGP attributes

Attribute Name	Attribute Type	Used in Route Decision
Local Preference	Well-known discretionary	Yes
AS Path	Well-known mandatory	Yes
Origin	Well-known mandatory	Yes
Multiple Exit Discriminator (MED)	Optional nontransitive	Yes
Next Hop	Well-known mandatory	No
Attribute Name	Attribute Type	Used in Route Decision
Local Preference	Well-known discretionary	Yes

© 2020 Juniper Networks, Inc. All Rights Reserved.

Common BGP Route Attributes

The slide lists six common BGP attributes. In the following pages, we describe the uses of these attributes and the categories in which they fall.

Agenda: BGP Attributes and Policy—Part 1

- BGP Policy Overview
- BGP Attributes Overview
- →Local Preference
- AS Path

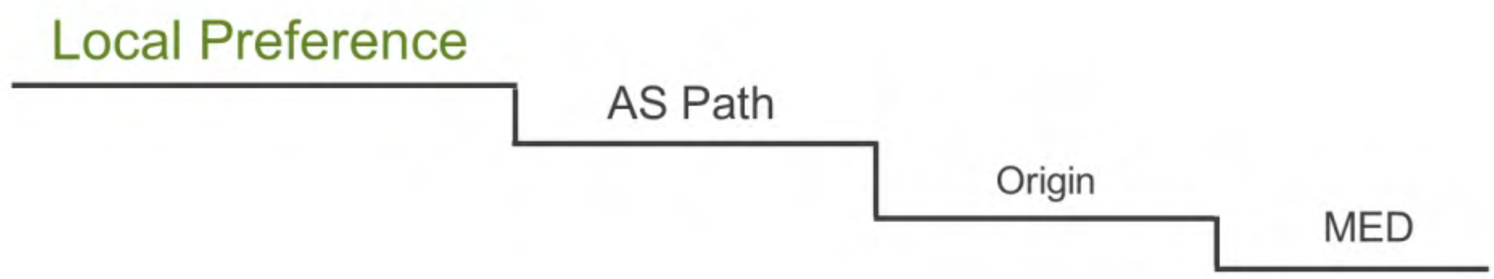
© 2020 Juniper Networks, Inc. All Rights Reserved.

Local Preference

The slide highlights the topic we discuss next.

The Power of Local Preference

- Local preference is the first BGP attribute used to favor one route over another
- A route with higher local preference always wins—regardless of the AS-path length



© 2020 Juniper Networks, Inc. All Rights Reserved.

Local-Preference Power

The BGP attribute of local preference is the highest tiebreaker in the BGP path selection process. If a BGP next hop is reachable, and BGP knows multiple routes, BGP always chooses the route with the highest local preference. Thus, local preference is the first BGP attribute that favors one path over another.

Highest Local Preference Wins

Because of the position of the BGP local preference, neither the AS-path length, nor the origin code, nor the MED value matter. The route with the highest local-preference value is always chosen as the exit point of the AS—the end.

Local Preference Is Not Route Preference

- Local preference is a priority applied *within* BGP itself
  - If you set no local preference for a route, a default value of 100 is used
  - Set local preference in `[edit protocols bgp]`
  - Change local preference with **`local-preference`** in a routing policy
- Route preference is a priority applied to routing protocols themselves
  - For example, the BGP *preference* is 170
    - Set preference in `[edit protocols bgp]`
    - Change preference with `preference` in a routing policy
- Make sure to change local preference, not route preference!

© 2020 Juniper Networks, Inc. All Rights Reserved.

Local-Preference BGP Attribute

Do not confuse the BGP attribute of local preference with the Junos OS concept of route preference. The Junos OS route preference is local to an individual router and assists the routing table in choosing the active route among many possible paths.

The BGP local-preference attribute is used only within BGP itself. The BGP routers transmit the local preference within an AS. If you do not explicitly configure a value for local preference, the default value used on BGP routes is 100. You can change this value on a per-peer basis using the **`local-preference`** command within the `[edit protocols bgp]` configuration hierarchy. In addition, you can alter the value using a policy on a per-route basis. The policy action also uses the **`local-preference`** command to alter the attribute value.

Default Local Preference = 100, BGP Preference = 170

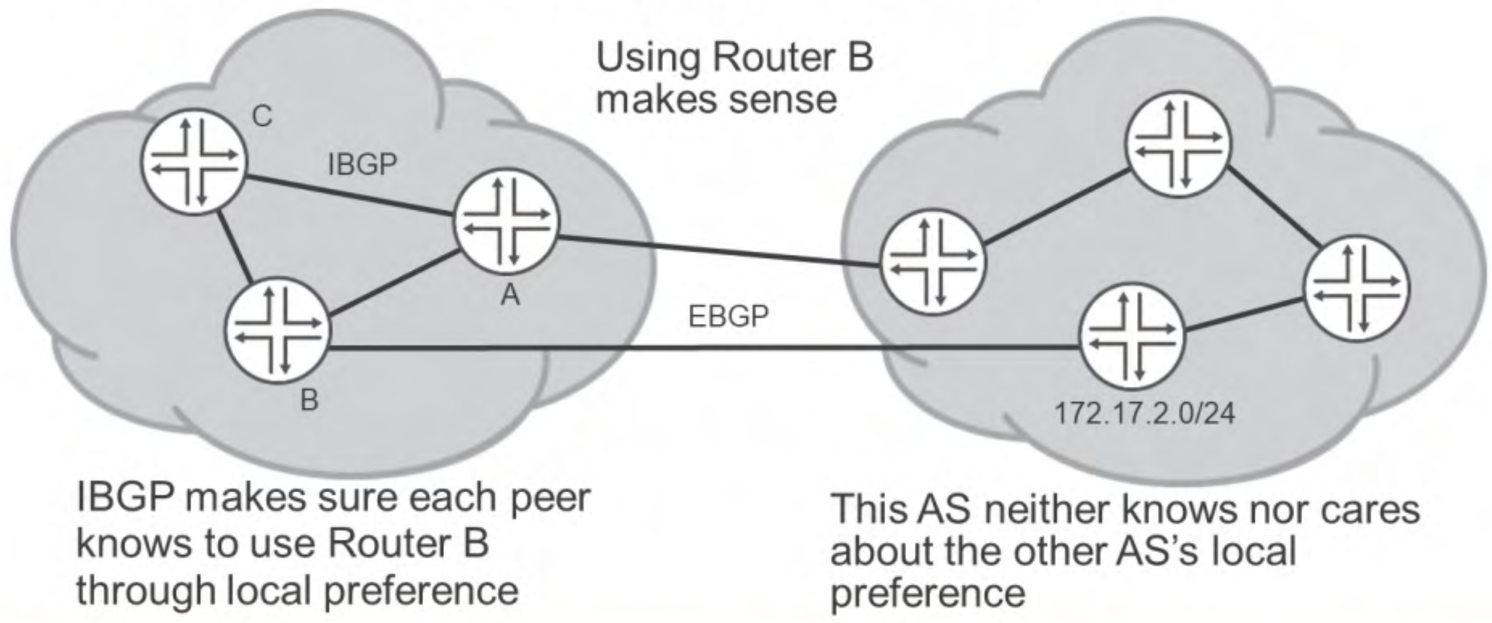
The default local preference applied to a BGP route is 100. However, the default route preference applied to BGP itself is 170. The Junos preference applies to the entire routing protocol. Preference is also set in the `[edit protocols bgp]` configuration hierarchy, but as `preference`, not `local-preference` (which applies only to BGP).

Pay Attention

When it comes to routing policy configurations, make sure to change local preference on the BGP routes, not the preference of the BGP protocol as a whole!

Local-Preference Notes

- Exchanged by IBGP peers only
- Usually used to set the exit point from an AS
- IBGP propagates information throughout the AS
- Which router should C use to reach 172.17.2.0?



© 2020 Juniper Networks, Inc. All Rights Reserved.

IBGP Peers Exchange Local Preference

Only IBGP peers exchange the values of the local-preference BGP attribute. EBGP peers never see a local preference set on route information sent between AS networks.

Sets Exit Point

You typically use the local preference to set the preferred *exit point* for a particular route from an AS. This use can be important when several links exist between two ASs.

IBGP Distributes Local-Preference Information

Once the local preference for a route is set, IBGP propagates that information throughout the entire AS.

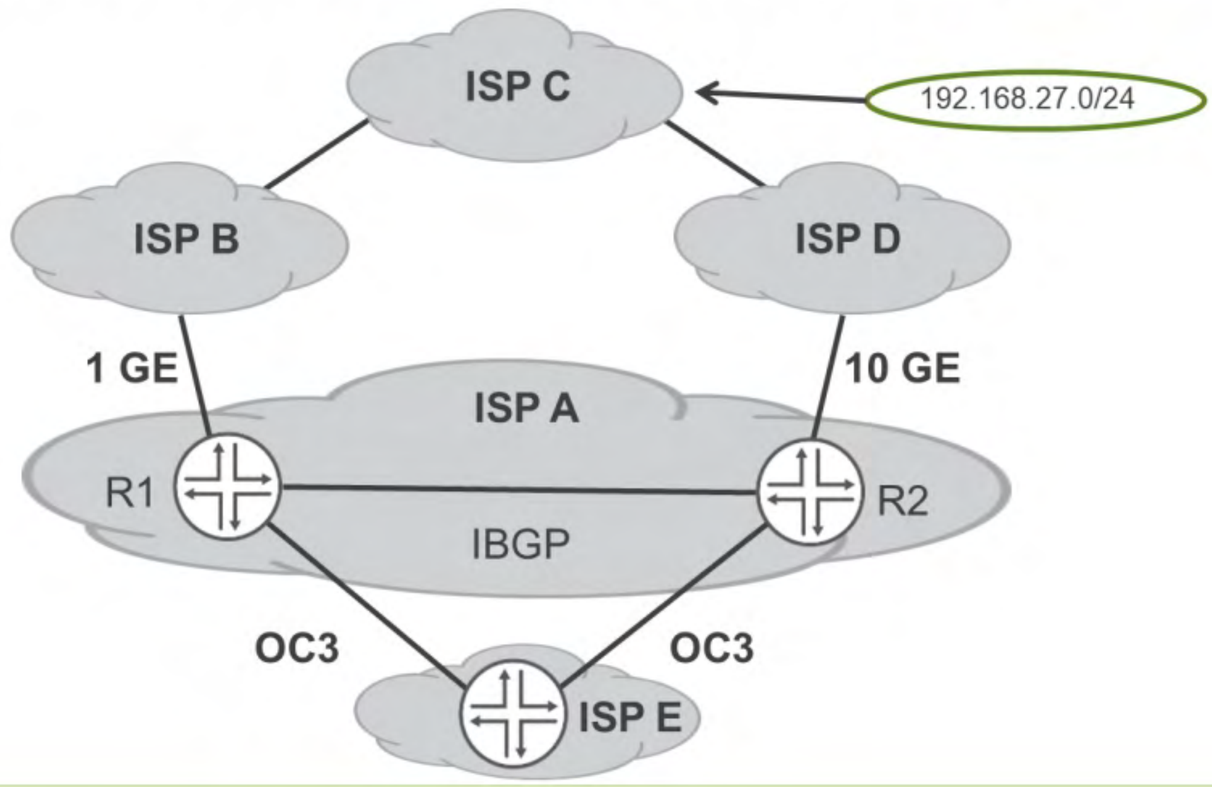
How to Get to 172.17.2.0/24?

The slide shows the concept of how local preference affects traffic leaving an AS. The administrators of the AS on the left know that Router B should always be used to reach the 172.17.2.0/24 network in the AS on the right. Therefore, the administrators of the AS on the left can configure their edge routers to set the local-preference value higher on the copy of the route received on Router B than the copy received on Router A. IBGP makes sure that every router in the AS knows that the preferred exit point for this route is Router B. Note that the AS on the right neither knows nor cares about the value of the local preference assigned to the route by Router A or Router B.

The AS on the left still has failover capability because it receives the route from the AS on the right through both routers. Although Router B is the primary exit point for the route, user traffic can use Router A to reach the AS on the right in the event of a failure.

Local-Preference Scope

- Local preference is not propagated to ISP E
- ISP E must rely on other BGP attributes to make a path determination



© 2020 Juniper Networks, Inc. All Rights Reserved.

Local AS Only

The slide points out the local nature of local preference. Consider another AS called ISP E linked by two lower-speed, but equal, links to ISP A. Which link should ISP E use to reach 192.168.27.0/24 in ISP B?

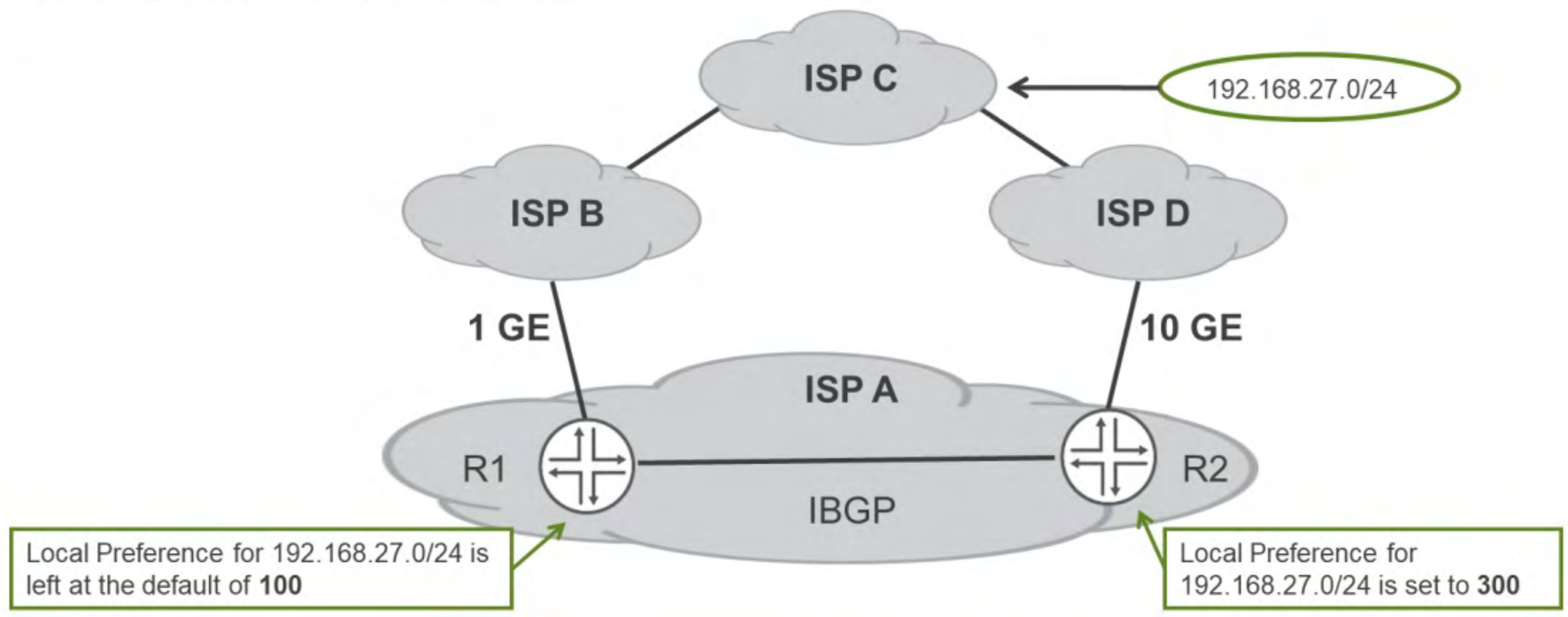
Because EBGP does not propagate the local-preference values used inside ISP A, the ISP E AS has no knowledge of the local-preference routing decisions made within ISP A with regard to the 192.168.27.0/24 route. ISP A, of course, still wants traffic from ISP E to flow towards R2 to avoid shuttling all this traffic across its backbone.

Another Method Needed

However, ISP A cannot accomplish this goal with local preference. ISP A must use other BGP attributes instead of local preference, such as AS path or MED, to influence ISP E's flow of traffic to ISP A. This is because of the strictly local nature of the local-preference attribute.

Local-Preference Case Study 1

- ISP A wants to use the 10 Gigabit Ethernet link for outbound traffic but have the 1 Gigabit Ethernet link available for inbound traffic and backup outbound traffic



© 2020 Juniper Networks, Inc. All Rights Reserved.

Example of Local-Preference Use

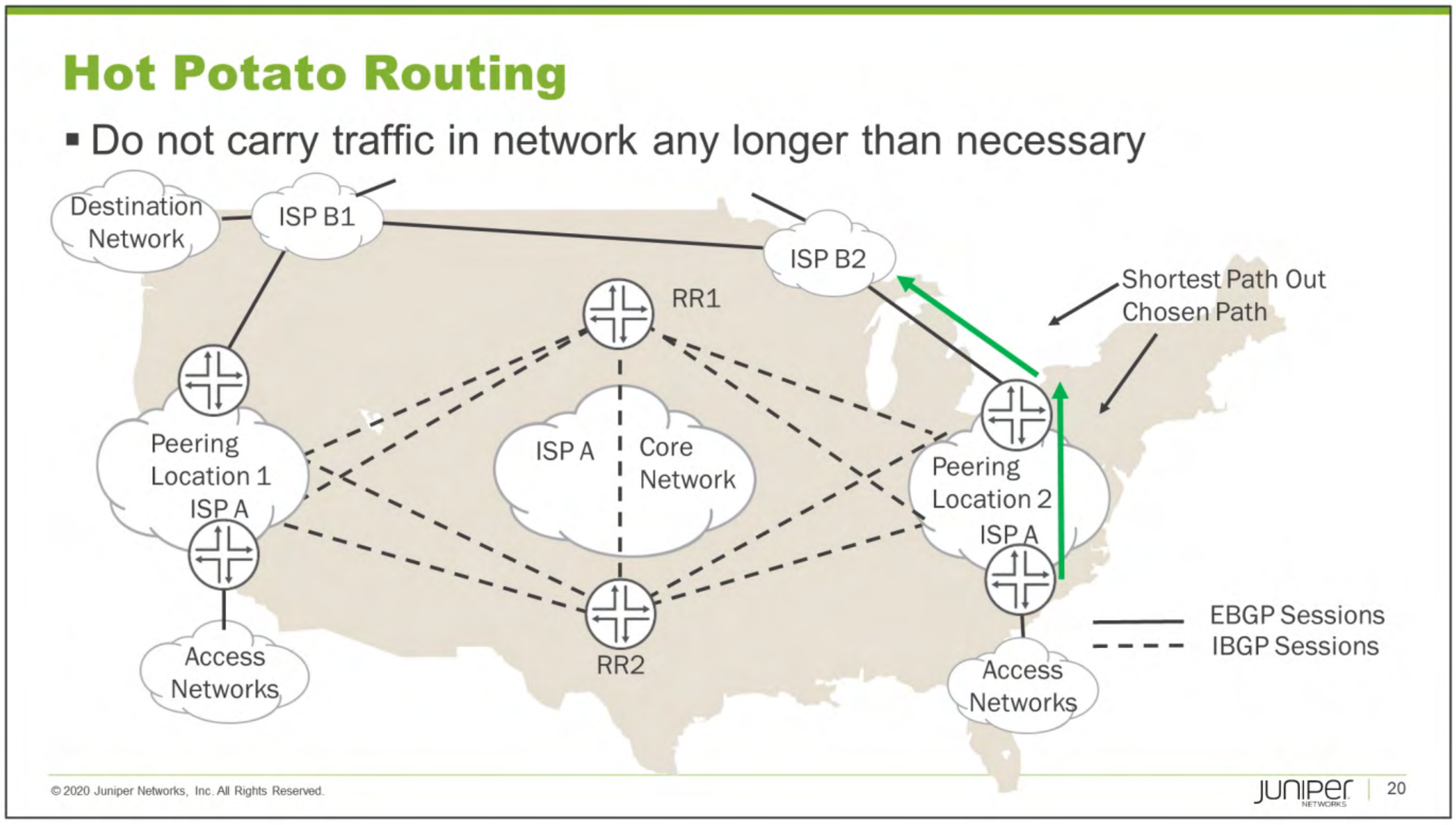
The slide shows an example of local-preference use within an AS.

The network administrators in ISP A decided that the routers in ISP A should use R2 to reach the 192.168.27.0/24 network in ISP C. This decision is because of the greater bandwidth capacity available on that 10 Gigabit Ethernet link.

To perform this router assignment, the ISP A network administrators leave the default local-preference of 100 on the route to 192.168.27.0/24 received from ISP B on R!, and then set the local preference to 300 on the route to 192.168.27.0/24 received from ISP D on R2.

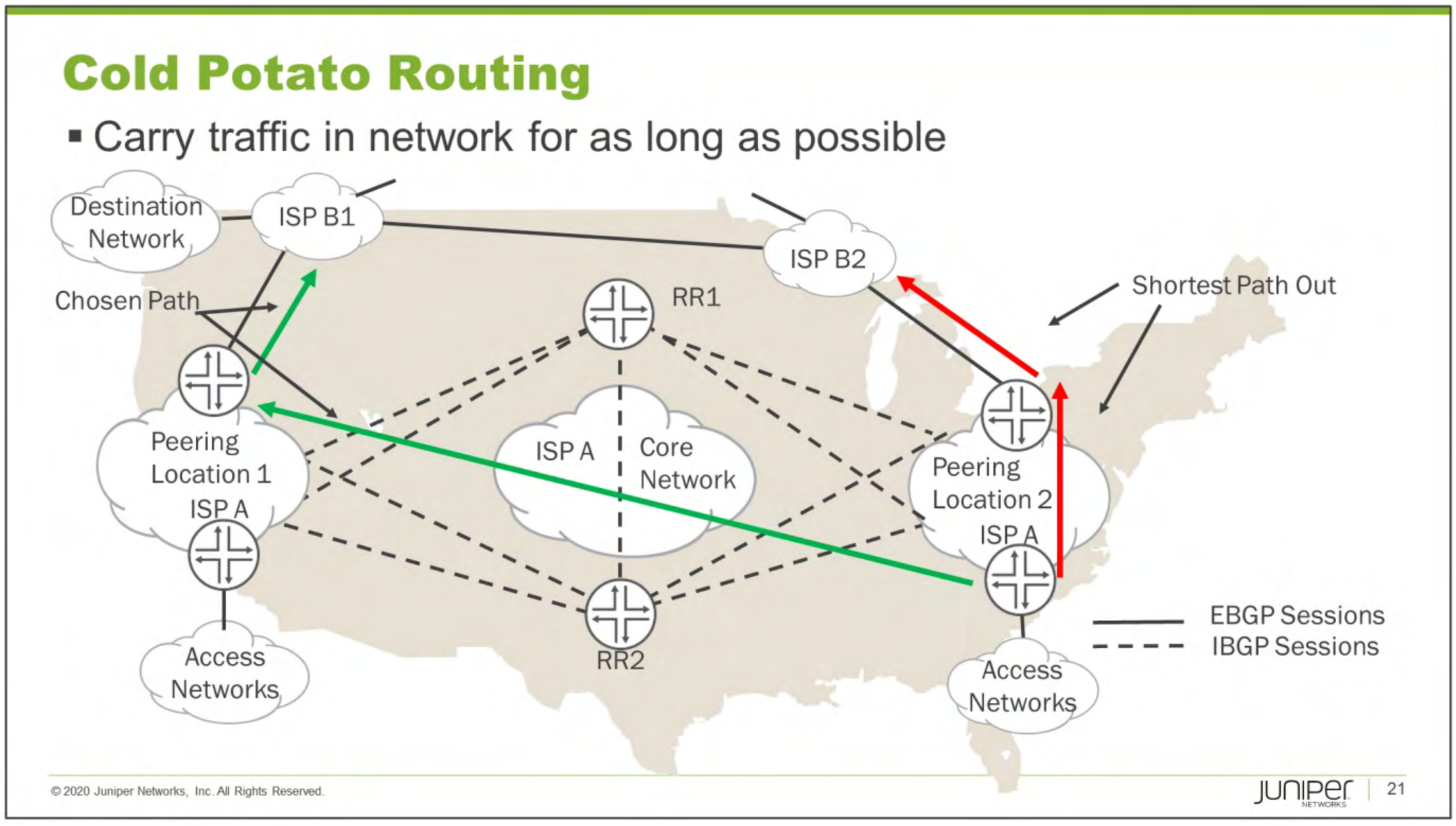
In contrast to almost every other metric associated with routing protocols, the highest local preference is better. Thus, for this route, the exit point of the AS is through R2. IBGP makes these values known to all other routers in ISP A.

The link on R1 can still be used for inbound traffic flows from ISP C and for outbound failover traffic if the 10 Gigabit Ethernet link is not usable because of a router or link failure.



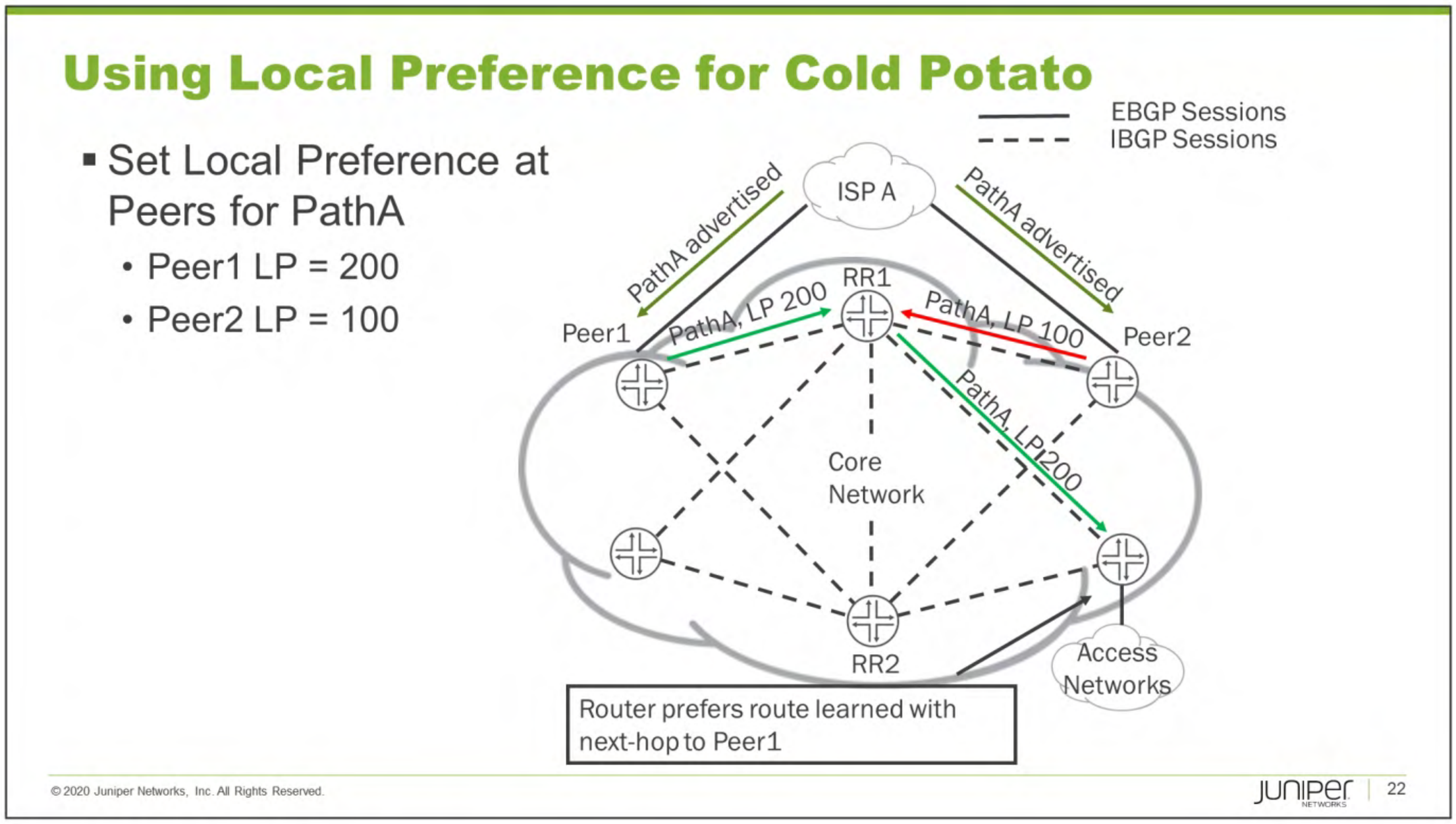
Hot Potato Routing

The concept of hot potato routing is simple - only carry traffic on your own network as long as necessary. In the example, the path through ISP B2 represents the shortest path within ISP A's network for traffic needing to reach the destination network when traffic comes into peering location 2. Transporting customer traffic across the core should only occur if the shortest path is unavailable.



Cold Potato Routing

Cold potato routing is the exact opposite of hot potato routing. In this case, the service provider needs to control the traffic for as long as possible. This could be due to contractual or SLA agreements. In this example traffic entering Peering location 2 would be kept in network and passed to the destination network across the core and delivered to ISP B1.



Using Local Preference for Cold Potato Routing

Referring to the slide, local preference is used to manipulate the routes learned so Peer1 is always the first choice because local preference is set higher than the other peering router. In this case, the access networks shown will choose the cold potato path to reach the destination represented by PathA. Furthermore the Route Reflector will actually hide the second route by not advertising it. This hiding of the network topology is not always optimal.

Route Reflectors in this scenario will be discussed next.

BGP Add-Path (1 of 2)

- Route reflectors hide information
  - Only the "best" route is advertised to clients
  - Backup routes are not sent
  - Clients unable to make their own routing decisions
- The BGP add-path feature allows multiple advertisements
  - Route reflector will send preconfigured number of routes
  - Client can make its own decision on best route
  - Client must be configured to receive multiple routes
  - Downside: Multiple copies of a route means more control plane memory usage

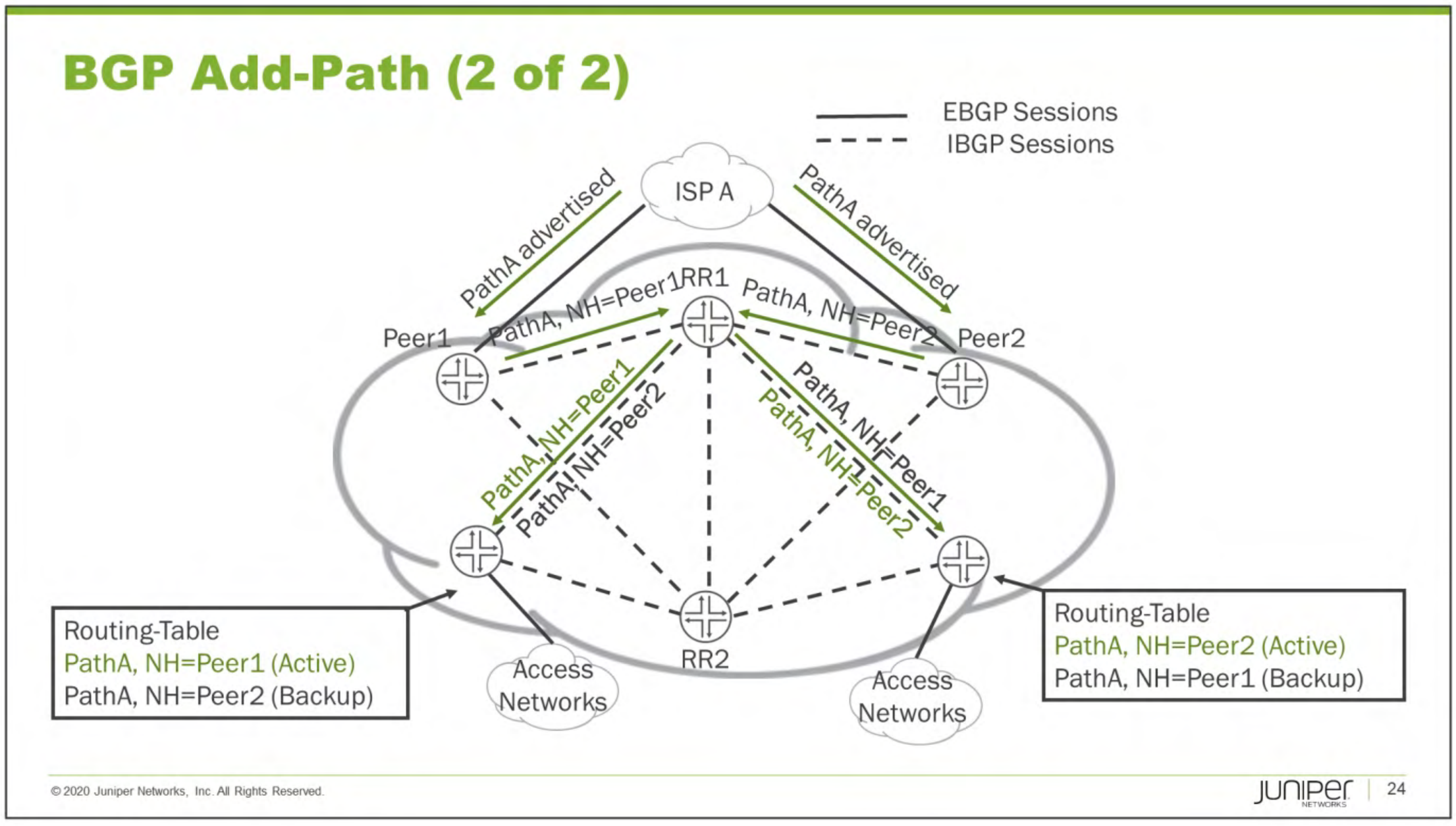
© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Reflectors Hide Information

In their normal state, route reflectors reflect the active route to their clients, meaning network detail can be lost. Often this is not a problem since the "best" route is typically the route a network would want to take to reach a destination. However, that best route can be a different calculation on a route reflector than on a client. Sending multiple routes to a client allows the client to decide which is the "best" route for the client to take.

The BGP `add-path` Feature Allows Multiple Route Advertisements

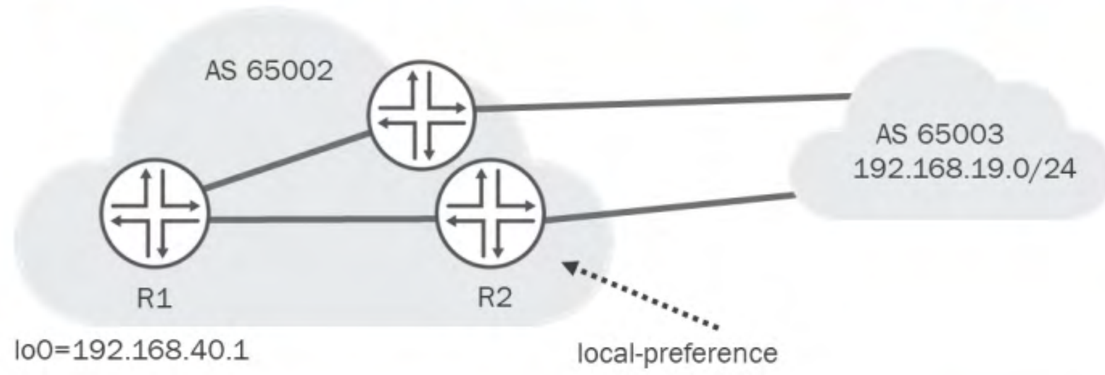
If the `add-path` feature is enabled, the route reflector will advertise the configured number of BGP routes with the same prefix to its clients. Clients will receive multiple routes and can choose the best one based on their own criteria in the network. The downside to this feature is each client router now carries a larger RIB table. add-path requires configuration of both the source peer to advertise the multiple routes and the client to be able to receive the additional routes.



BGP `add-path`

In this slide you can see both clients of the Route Reflector are receiving routes from the Route Reflector. This provides a couple of advantages. Using local preference or even the IGP metric, traffic can be directed out the desired exit point and in the case of active path failure, the backup path is known and can be immediately placed into service.

Modifying Local Preference using BGP




```
user@R2> show route advertising-protocol bgp 192.168.40.1
inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
Restart Complete
  Prefix                  Nexthop              MED     Lclpref    AS path
* 192.168.19.0/24         Self                 0       100        65003 I

[edit]
user@R2# set protocols bgp group int-peers local-preference 300

user@R2> show route advertising-protocol bgp 192.168.40.1
inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
Restart Complete
  Prefix                  Nexthop              MED     Lclpref    AS path
* 192.168.19.0/24         Self                 0       300        65003 I
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Modifying the Local Preference

The Junos OS provides a configuration option within BGP that alters the local preference attribute value for all advertised routes. You can use the **`local-preference`** command at either the global, group, or peer level in the BGP configuration. All advertised routes will inherit this value for the local preference attribute.

The exception to this rule are any routes whose attributes are modified by an applied routing policy. These routes abide by the action defined in the policy and take precedence over the configured value. In other words, the configuration option is applied before the routing policy is applied for all outbound BGP routes.

Modifying Local Preference using Policy

- Change the local preference of routes with an AS path of 124 44 13 to 200 (instead of the default 100)

```
[edit policy-options]
policy-statement check-the-path {
   term got-path {
      from as-path look-for-path;
      then {
         local-preference 200;
         accept;
      }
   }
}
as-path look-for-path "124 44 13";
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Changing Local Preference with a Policy

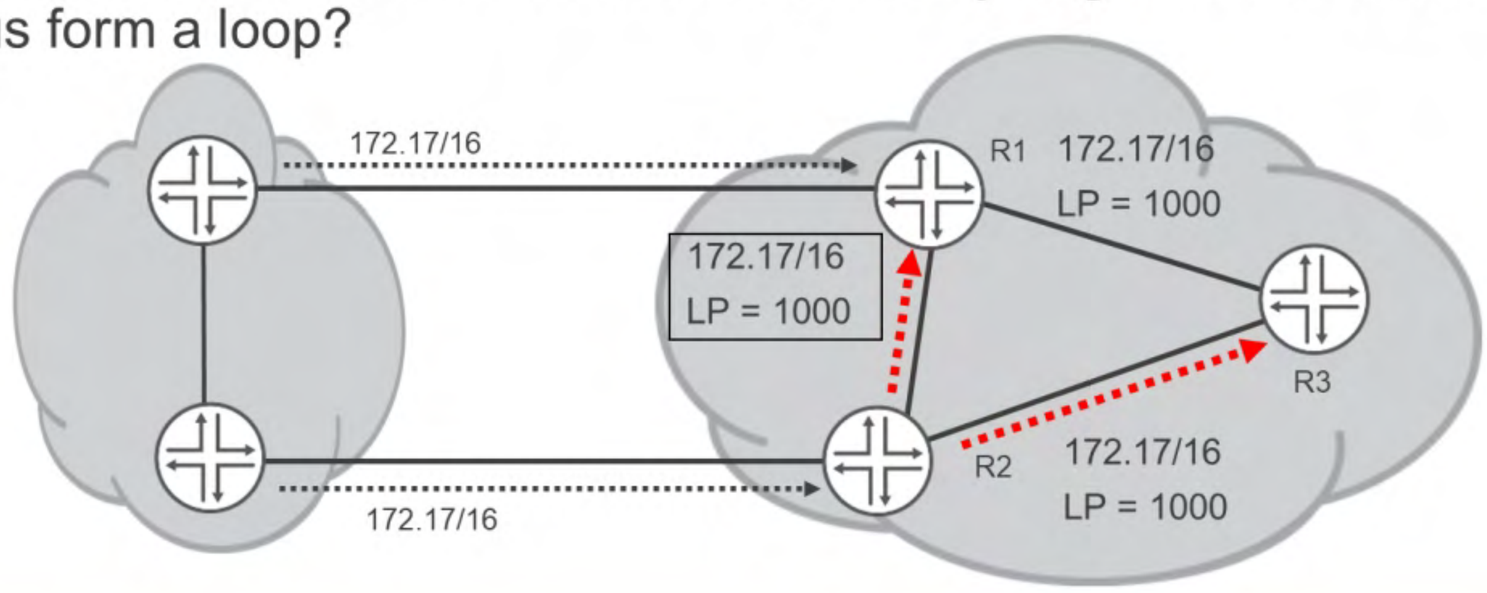
If a local preference has not been changed from its default, Junos considers other BGP attributes, such as the AS path, origin, MED, etc. to determine the best route. This requires additional processing. For this reason, policies can be used to evaluate the route and set the local preference to ensure that the no other attribute controls the route priority.

The example on the slide alters the local-preference value for all received routes that match the AS path of `look-for-path`. Within the term `got-path`, we specified an action of **`local-preference 200`**. This action changes the local-preference attribute value for those routes to 200.

This routing policy does not affect any other received BGP routes.

Creating Routing Loops?

- Because local preference overrides all other attributes, can a routing loop be created in your network?
  - Both R1 and R2 set local preference to 1000 to all IBGP peers (including each other) using **`export`** policy
  - Does R1 and R2 see each other as the best way to get to the remote network and thus form a loop?



© 2020 Juniper Networks, Inc. All Rights Reserved.

Local Preference and Routing Loops

The case study on the slide provides an interesting look at the operation of BGP in general, and how BGP uses local preference specifically.

Considering that the local-preference attribute overrides all other BGP attributes in the path selection process, it might be possible to create a routing loop in BGP. Routing loops are especially bad because the destination is technically reachable, but some or even all routers within an AS cannot find the proper path to the destination.

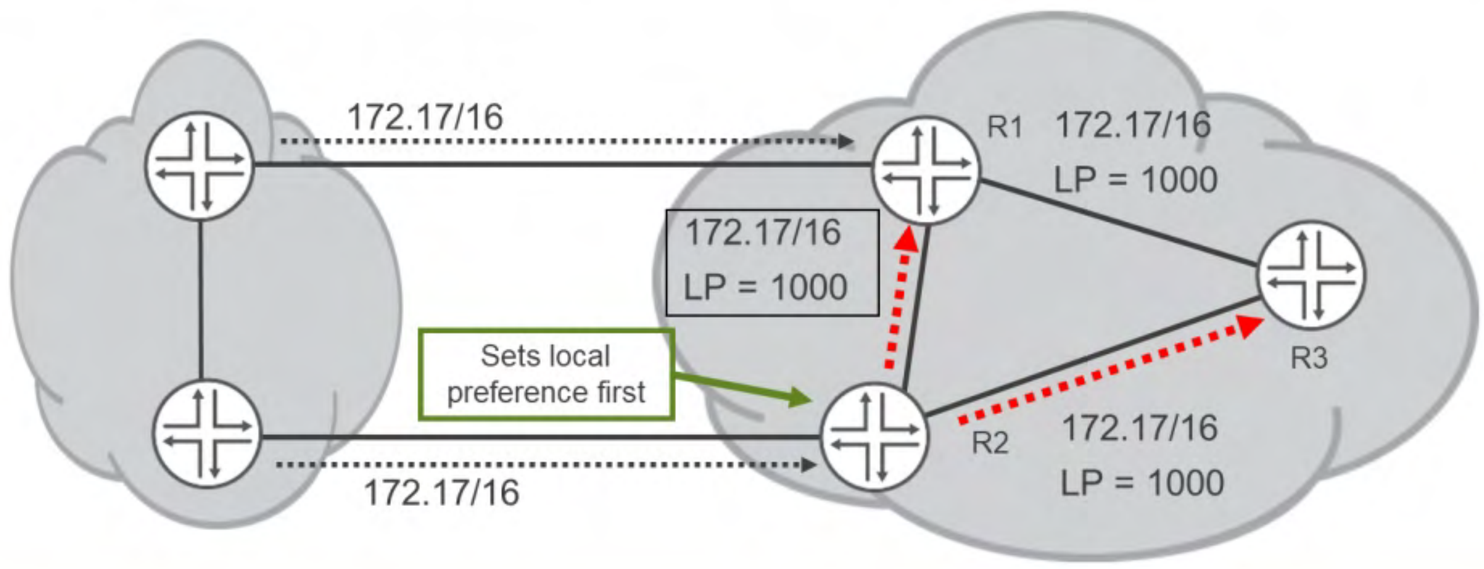
In the example on the slide, both R1 and R2 have an *export* policy in place that sets the local preference to 1000 on EBGP routes before these routes are readvertised to IBGP peers. Note that R1 and R2 have an IBGP peering relationship with each other, which means that both routers also advertise the route to each other.

Theoretically, both R1 and R2 could see the preferred value of 1000 in the copy of the route they receive from each other. This information makes each router determine that the IBGP version is better than the *local* (EBGP) copy of the route! This determination occurs because the local preference on the EBGP version of the route that is stored within the router is still set to 100. As a result, R1 and R2 see *each other* as the best route to 172.17.0.0/16, and a routing loop occurs as the two routers shuttle traffic back and forth.

Note that this situation is easily avoided by setting a route's local preference at ingress using an `import` policy. In this case, the local copy of the route as received from an EBGP peer will correctly reflect the local preference modification.

No Loop in This Case

- IBGP sessions implement split horizon, so only R1 or R2 sends out local-preference values
  - Assume R2 sets local preference to 1000 first
    - R1 installs the R2 routes as active and does not advertise those routes *back* to R2
    - A routing loop is not formed in this particular case



© 2020 Juniper Networks, Inc. All Rights Reserved.

Split Horizon for Local Preference

It turns out that BGP is smart in this case. The following chain of events assumes that R2 received the EBGP route of 172.17.0.0/16 first and advertises the route with a local preference of 1000 to its IBGP peers. (If R1 accomplishes this first, the roles are easily reversed, but the point is the same.)

First, R2 receives the EBGP route of 172.17.0.0/16 from its peer in the other AS. Because R2 detects this route as the current best, R2 installs the route in its routing table.

Next, R2 alters the local preference to 1000 and advertises this route to its IBGP peers with this local-preference value. At this point, both R1 and R3 receive the 172.17.0.0/16 route through IBGP.

R1 then also receives the EBGP route for 172.17.0.0/16 from the other AS. At this point, R1 determines that it already has a route to 172.17.0.0/16 from R2, with a local preference of 1000. The presence of this route to R2 overrides the EBGP-received route.

Because the current version of the route on R1 is from R2, and because IBGP implements split horizon, the routing loop never forms. R1 just sends traffic for 172.17.0.0/16 to R2.

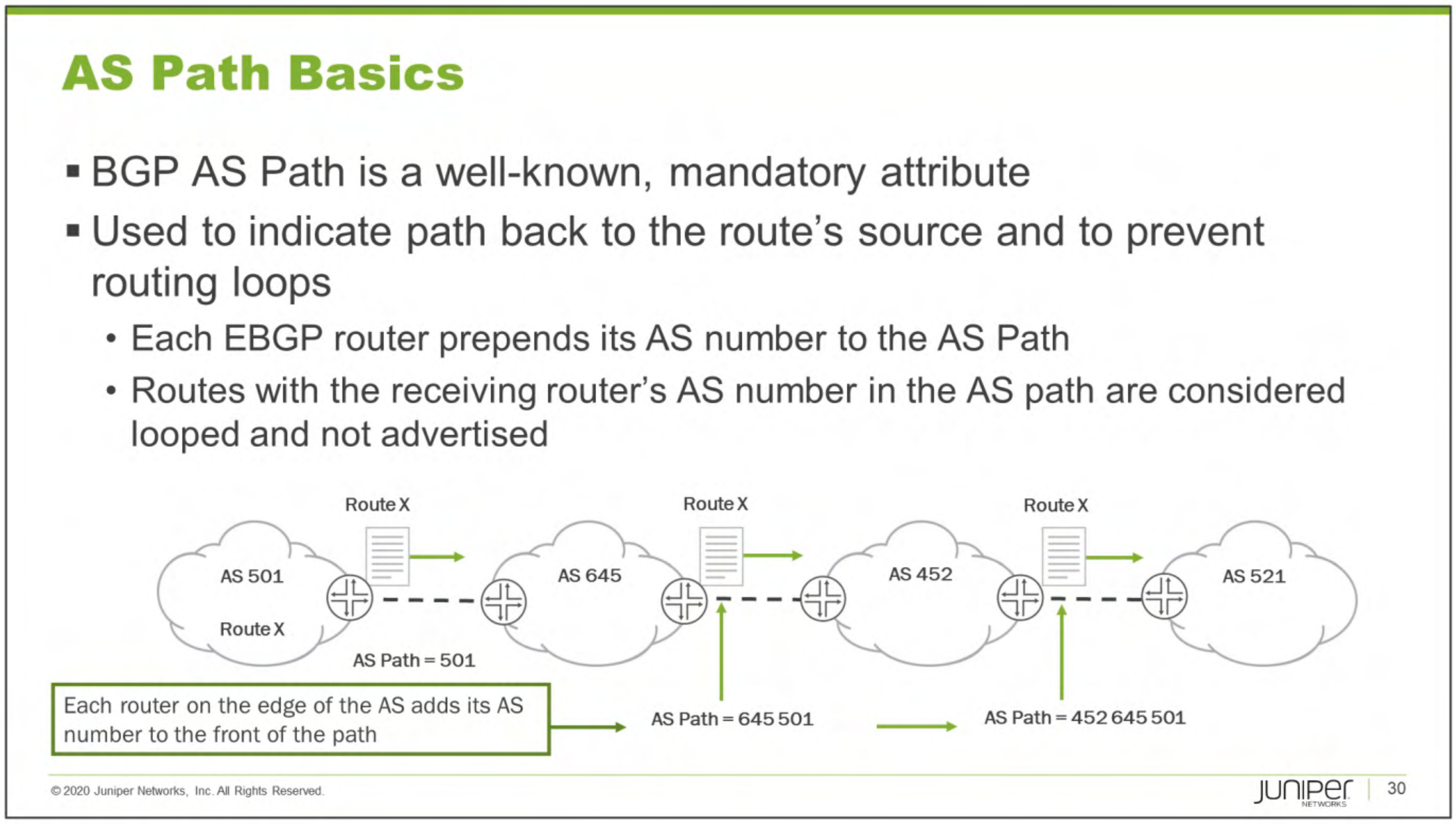
Agenda: BGP Attributes and Policy—Part 1

- BGP Policy Overview
- BGP Attributes Overview
- Local Preference
- →AS Path

© 2020 Juniper Networks, Inc. All Rights Reserved.

AS Path

The slide highlights the topic we discuss next.



The AS-Path Attribute

The AS-path attribute describes the path of autonomous systems that the route has been through since it was sourced into BGP. When a BGP router receives routes in an update message, the first action is to examine the current AS path to see if the local AS number (ASN) is in the path. If it is in the path, it indicates that the route has been through the AS already; accepting the route would cause a loop. Therefore, BGP drops the route. The Junos OS processes BGP routes differently. Prior to advertising a route, Junos performs a verification to ensure that the next router's AS number is not listed in the AS path. If the next router's AS number is listed in the AS path, the router does not advertise the route.

By default, the AS-path value is changed as a route transitions between autonomous systems. The AS-path value is null until the associated route is advertised out of the originating AS. As the route leaves an AS, the BGP router adds the local AS number to the front of the path before sending it to a peer. Using routing policy, you can prepend additional ASN information to the AS-path attribute. By prepending your ASN information to the AS-path multiple times, you can affect the routing decision made by routers in other autonomous systems and discourage those routes from using that path because of the longer AS-path.

The AS-path attribute is mandatory; thus, it must always be present for all BGP routes.

Autonomous System Numbers

- RFC 1930 defined ASNs as 16-bit integers
  - 65536 ASNs available (some reserved by IANA)
  - Private range is 64512-65534
- RFC 4893 defines 32-bit ASNs
  - Can be entered as a single 4-byte integer
  - Written as two 16-bit numbers: x.y
  - Old ASNs written as 0.y
  - 1.y and 65535.65535 are reserved
- Single 4-byte AS numbers are displayed by default.
- Two 2-byte number format can be displayed instead using the `asdot-notation` option

© 2020 Juniper Networks, Inc. All Rights Reserved.

16-bit Autonomous System Numbers

AS numbers are assigned in blocks by the Internet Assigned Numbers Authority (IANA) to Regional Internet Registries (RIRs). The RIR then assigns AS numbers to entities within its designated area.

RFC 1930 defined 65536 ASNs using 16-bit integers. The ASNs 0, 56320–64511, and 65535 are reserved by the IANA and should not be used in any routing environment. IANA designated AS numbers 64512 through 65534 to be used for private networks.

32-bit Autonomous System Numbers

As of January 1, 2009 the Internet Assigned Numbers Authority (IANA) only assigns 4-byte AS numbers, unless a 2-byte AS number is specifically requested.

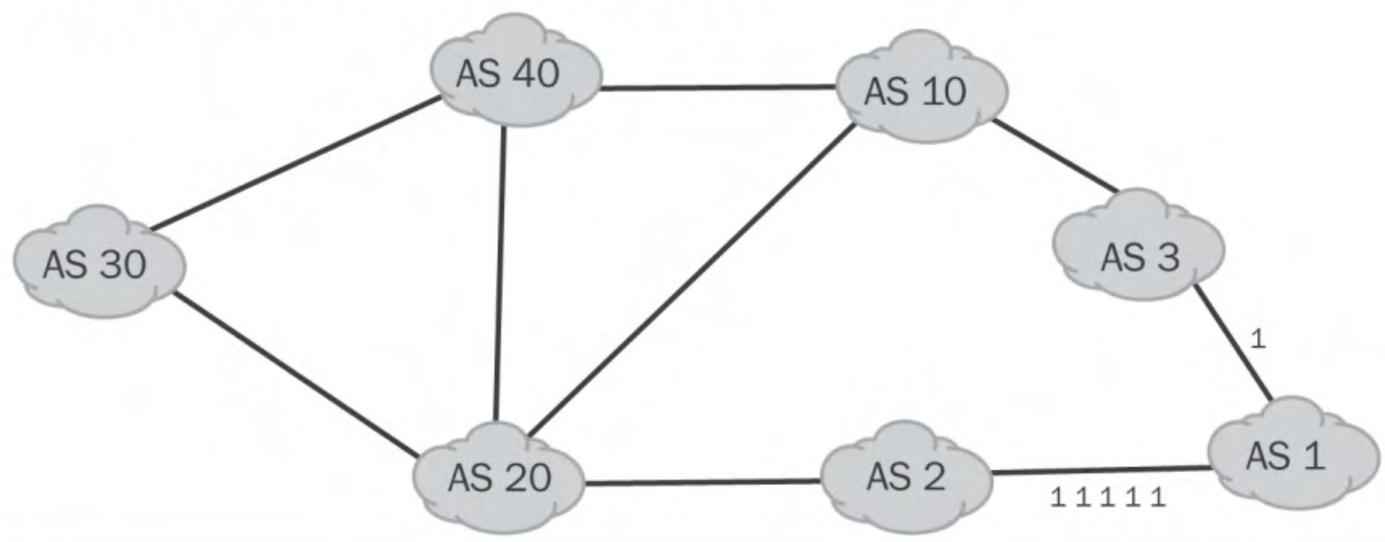
RFC 4893 defines 32-bit ASNs. These numbers are written either as simple integers, or in the form x.y, where x and y are 16-bit numbers. Numbers of the form 0.y are exactly the old 16-bit AS numbers, 1.y numbers and 65535.65535 are reserved, and the remainder of the space is available for allocation.

Regardless of how the AS number is entered, Junos will display the numbers in all of the `show` commands in the single 4-byte format. This default display option can be overridden by adding the `asdot-notation` option after the AS number configuration.

```
lab@mxA# show routing-options
autonomous-system 123456 asdot-notation;
```

Modifying AS-Path: Prepend

- Manipulating the AS-path attribute is a major way to favor or disfavor BGP routes
- AS 1 prepends its AS number four times to AS 2
  - How does AS 40 reach AS 1?
  - How about the other remote networks?



© 2020 Juniper Networks, Inc. All Rights Reserved.

Manipulating the AS Path

If the AS-path attribute is changed before the route is readvertised to other BGP routers, a route through the local AS can look less attractive to another AS. Note that making the route more attractive by shortening the AS path should not really be possible. However, you can lengthen the path to make the AS-path attribute another type of negative-bias path selection mechanism.

AS-Path Prepending

The only standard approach to alter the AS-path attribute is to add information to it by prepending. You can use a routing policy to extend (by prepending) AS-path information artificially onto an existing AS path. This type of a policy is often an attempt to influence traffic coming into an AS from another AS.

On the slide, AS 1 announces its routes to both AS 2 and AS 3. Using a routing policy, AS 1 prepends its own AS number four times (shown as 1 1 1 1 1 on the slide) onto all route announcements to AS 2. This action causes the following:

- AS 2 will use AS 20 to forward packets to AS 1;
- AS 10 will use AS 3 to forward packets to AS 1;
- AS 20 will use AS 10 to forward packets to AS 1;
- AS 30 will use AS 40 to forward packets to AS 1; and
- AS 40 will use AS 10 to forward packets to AS 1.

Note that this behavior on the part of AS 2 (using AS 20 instead of sending directly to AS 1) is unexpected and would not occur without the routing policy. This behavior is extended to AS 20 as well, because AS 2 cannot shorten the AS path advertised by AS 1, even if AS 2 would like to shorten it.

AS-Path Prepend Example

```
[edit routing-options]
autonomous-system 1;


[edit protocols]
bgp {
   group peer-AS2 {
      type external;
      export longer-as-path;
      peer-as 2;
      neighbor 10.10.10.2;
   }
}

[edit policy-options]
policy-statement longer-as-path {
   then as-path-prepend "1 1 1 1";
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Policy to Prepend AS Path

The slide shows the routing policy AS 1 used in the previous example.

A policy named `longer-as-path` was created on the AS 1 router. Because no `from` statement exists, all candidate routes match the policy. The action taken on all of the matched routes is to add AS 1 to each route four times. You can use either the **`as-path-prepend`** ***`as-number`*** or the **`as-path-expand last-as count`** ***`number`*** policy options to add AS numbers to the AS-path attribute in an attempt to make a given route appear less desirable. The latter option automatically locates the most recent AS in the path (the last entry) and prepends it the specified number of times prior to advertising the route to EBGP peers. This particular action is performed with the **`as-path-prepend`** statement. This routing policy then is applied for routes exported by BGP to the external BGP peer in AS 2.

Output of AS Path Information

- Output of **`show route`** displays various outputs in AS path
  - Brackets enclose the local AS number when additional AS numbers have been added
  - Braces enclose AS sets
  - Parentheses enclose a confederation

```
user@router> show route protocol bgp terse

inet.0: 42 destinations, 42 routes (42 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination        P Prf Metric 1   Metric 2    Next hop        AS path
* 64.168.48.0/24    B 170        100          5 >10.222.9.1      3944 2222 I
* 64.168.49.0/24    B 170        100          5 >10.222.9.1      7777 7777 [7777] I
* 64.168.50.0/24    B 170        100          5 >10.222.9.1      {444 555 7777} I
* 64.168.51.0/24    B 170        100         10 >10.222.9.1      (65009 65003 65000) 111 I
```

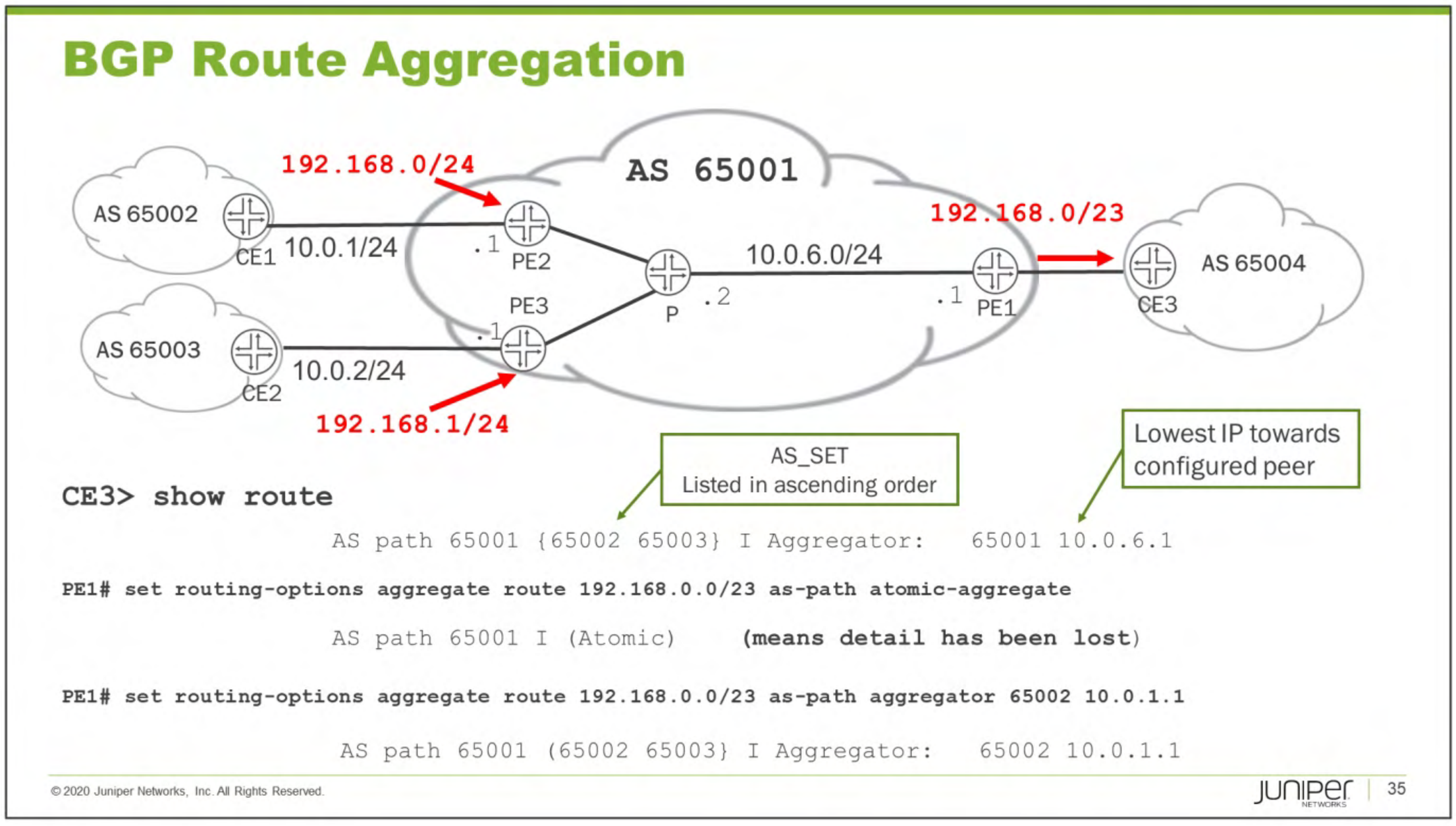
© 2020 Juniper Networks, Inc. All Rights Reserved.

The Symbol Tells All

A symbol can provide information on an object. For example, when looking at the AS path on a Juniper Networks router:

- *Brackets ([ ])*: Enclose the local AS number associated with the AS path if more than one AS number is configured on the router or if the AS number is being prepended.
- *Braces ({ })*: Enclose AS sets—groups of AS numbers in which the order does not matter. A set commonly results from route aggregation. The numbers in each AS set are displayed in ascending order.
- *Parentheses (( ))*: Enclose a confederation.

In the output on the slide, you can see four routes with different AS paths. The second route represents a bracket output, the third route represents an AS set, and the fourth route represents a confederation.



BGP Route Aggregation

The example on the slide shows how BGP handles route aggregation. In the slide AS 65002 is forwarding the 192.168.0/24 route to AS 65001 and AS 65003 is redistributing 192.168.1.0/24 into AS 65001. AS 65001 is then redistributing a single route 192.168.0/23 to AS 65004.

By going to CE3 and viewing the route's AS path, you can see the AS path number sent to AS 65004 is 65001, but as includes an AS_SET which saves the AS numbers of the contributing ASs (in this case 65002 and 65003). As part of the AS path output, an aggregator entry will appear that includes the AS number that aggregated the routes and the lowest IP address that faces the configured peer.

By including the `as-path atomic-aggregate` option when creating the aggregate route, only the AS of the aggregating device is posted but the word (Atomic) is included to indicate that the detail has been suppressed.

The aggregator option allows you to specify a different AS number than the AS that formed the aggregate route (encoded as two octets) followed by the IP address of the BGP system that contributed to the aggregate.

Regular Expressions and BGP Routes

- Often, BGP policy relies on finding prefixes based on their AS-path information
  - Used to enforce administrative policy
  - Sometimes easier than looking for specific prefixes
- Common policy requirements:
  - Find all routes originating in AS 1
  - Find all routes that transited AS 100
  - Find all the routes originating in my own AS
- AS-path regular expressions allow the selection of the proper prefixes

© 2020 Juniper Networks, Inc. All Rights Reserved.

Regular Expressions

Often, administrative BGP routing policies require that route announcements or prefixes be found and acted on based on the information contained within the AS-path attribute. This requirement might be to enforce some administrative policy regarding other AS networks. Sometimes, it is just easier to find routes based on their AS path than by looking for many specific prefixes and routes individually.

Other Uses

The slide lists some examples of the types of information that must be found in the AS path.

Regular Expressions Can Find Prefixes

Through the use of *regular expressions*, you can quite easily find this type of information in the AS-path attribute.

Regular Expressions

- A powerful *pattern-matching* engine
- Work not only on fixed strings (as do wildcards), but on variable patterns of text
- Combination of text and special operators
- Context-Sensitive Matching

© 2020 Juniper Networks, Inc. All Rights Reserved.

Regular Expressions for Pattern Matching

A regular expression (often called regex) is a powerful, *pattern-matching* tool you can apply in a routing policy to act on AS-path strings. This pattern-matching engine can find specific strings of text or textual patterns.

Different from Wildcards

When used in a routing policy, regular expressions work not only on fixed strings, like wildcard operators such as an asterisk (*), but also on variable patterns of text, through the combination of basic text patterns and special operators.

Text Plus Operators

Regular expressions are made up of a combination of basic text patterns and special operators.

Context-Sensitive Matching

Regular expressions allow information in a string to be found within a specific context. You can use regular expressions in conjunction with the BGP AS-path attribute to match routes within a policy.

Regular Expression Terms

- Regular expressions take the form *term operator*
- *Terms* are mandatory and identify the AS number:
  - Can be a single AS number
    - For example, 1024
  - Can be a complete AS path
    - For example, 1024 2685 3957
  - Can be a wildcard dot character (.), which represents a single AS
    - For example, 1024 . 3957
- Each AS number (not a character) represents a single *entity* to the regular expression parser

© 2020 Juniper Networks, Inc. All Rights Reserved.

Configure a Regular Expression

Regular expressions have two main components to them—the term and the operator. These expressions take the form *term operator*.

Terms Are AS Numbers in the AS-Path Attribute

The regular expression term is the core matching component to be found by the pattern-matching engine. The term is a mandatory object, and each regular expression must have at least one term. Terms identify the AS number. This AS could be a single number (1024), a complete AS path
(1024 2685 3957), or a wildcard character (.) representing any single AS number (1024 . 3957).

Each AS Is an Entity

Within the Junos AS-path regular expression syntax, the term is a complete AS value. You can use the wildcard character of the dot (.) to represent a single AS number as well. Thus, the Junos OS detects an AS number of 1024 (for example) not as the four character text string of 1, 0, 2, 4, but as the single *entity* of 1024. To the Junos OS, AS numbers are not sequences of characters.

Regular Expression Operators

- Regular expressions take the form *term operator*
- The operator is an optional pattern-matching character that applies to a single term
  - Operators immediately follow the term referenced
    - For example, 1024+ 2685
  - The pipe (|) operator is used between terms (| represents an OR)
    - For example, 1024 | 2685
  - The dash (-) operator is used between terms (- represents a range of values)
    - For example, 1024 - 2685
- One or more term-operator pairs can appear in an AS-path regular expression

Important to remember!

© 2020 Juniper Networks, Inc. All Rights Reserved.

Regular Expression Operators

All regular expressions take the form *term operator*. Use of the `term` is mandatory.

The Operator Is Optional

However, the regular expression operator is an optional component of the pattern-matching engine. You can associate an operator with a single, regular expression term. If used, the operator appears immediately after the term on which it is operating.

Two special regular expression operators can appear between regular expression terms. These special characters are the pipe (|) and the dash (-). You use the pipe (|) operator between terms to indicate OR (*1024 OR 2685 is 1024 | 2685*). You use the dash (-) operator between terms to indicate range (*1024 through 2685 is 1024-2685*).

More Than One

An individual regular expression can contain multiple term-operator pairs.

AS Path Regular Expressions

List of AS Path Regular Expressions Operators	
Operator	Description
{*m*,*n*}	Matches at least *m* and at most *n* repetitions of an AS number
{*m*}	Matches exactly *m* repetitions of an AS number
{*m*,}	Matches *m* or more repetitions of an AS number, up to infinity
*	Matches zero or more repetitions of an AS number
+	Matches one or more repetitions of an AS number
?	Matches zero or one repetitions of an AS number
|	Logical OR, matches one of the AS numbers on either side of the pipe
-	Matches an inclusive range of AS numbers
^	Regex start anchor (Optional) - Matches the first complete AS number in the AS path
$	Regex end Anchor (Optional) - Matches the last complete AS number in the AS path
(...)	Matches a group of AS numbers enclosed in the parentheses
()	Matches a null (empty) AS path

© 2020 Juniper Networks, Inc. All Rights Reserved.

Regex Operators for the AS Path

The table on the slide shows a subset of the possible regular expression operators you can use with routing policies. Some operators are a *shorthand* for their longer equivalents.

Of special note is the use of the parentheses. Typically, you use the parentheses ( ) operator to group multiple terms together in conjunction with an operator. For example, the regular expression (1|2)? is translated as *either AS 1 or AS 2 can be present in the AS path zero times* (absent), or at most one time (this is what the ? operator means).

The parentheses operator also has another special use. When used with no spaces in between, parentheses represent a *null AS-path* value. We cover the concept of the null AS path on future slides.

Regular Expression Examples

AS-path pattern to match:	Regex:	Sample matches:
Exactly one instance of AS 1234	1234	1234
0 or more instance of AS 1234	1234*	1234, 1234 1234, etc., or null AS path
0 or 1 instance of AS 1234	1234?	1234 null AS path
1 to 4 instances of AS 1234	1234{1,4}	1234, 1234 1234, 1234 1234 1234, 1234 1234 1234 1234
1 to 3 instances of AS 12 followed by 1 instance of AS 34	12{1,3} 34	12 34, 12 12 34, 12 12 12 34
Range of AS numbers to match a single AS	123 – 125	123 or 124 or 125

© 2020 Juniper Networks, Inc. All Rights Reserved.

AS-Path Regex Examples

The table on the slide shows some examples of regular expression pattern matching as applied to AS paths.

The first column shows the AS-path string that the routing policy is trying to match. The second column shows the regular expression used to match that pattern. The last column shows examples of values of the BGP AS-path attribute that the regular expression will match. In some cases, the list is not exhaustive, so more AS paths will match the pattern.

More Regex Examples

AS-path pattern to match:	Regex:	Sample matches:
Second AS *must* be 56 or 78	. (56|78)	1234 56, 34 78
Second AS *might* be 56 or 78	. (56|78)?	1234, 1234 56, 34 78
All paths from neighbor AS 1234	1234 .*	1234, 1234 5678, 1234 5 6 7 8
1 followed by 2, followed by one or more instance of 3	1 2 3+	1 2 3, 1 2 3 3, 1 2 3 3 3, etc.
One or more instance of 1, then one or more instance of 2, then one or more instance of 3	1+ 2+ 3+	1 2 3, 1 1 2 3, 1 1 2 2 3, 1 1 2 2 3 3, etc.
Any length path that contains the list 4, 5, 6 in it anywhere	.* 4 5 6 .*	1 2 3 4 5 6, 1 2 3 4 5 6 7 8 9, 4 5 6 7 8 9

© 2020 Juniper Networks, Inc. All Rights Reserved.

More AS-Path Regex Examples

The table on the slide shows more examples of regular expression pattern matching as applied to AS paths.

As before, the first column shows the AS-path string that the routing policy tries to match. The second column shows the regular expression used to match that pattern. The last column shows examples of values of the BGP AS-path attribute that the regular expression will match. In some cases, the list is not exhaustive, so more AS paths will match the pattern.

Regular Expression Formation

- AS regular expressions are defined as named objects

```
[edit policy-options]
user@user# set as-path name regular-expression
```

- Format:
  - *name* identifies the regular expression
  - *regular-expression* consists of the pattern to match in *term operator* format
  - The name can be up to 255 characters long
  - To include spaces in the name, enclose the entire name in double quotation marks
  - Can use the defined AS-path regex as a policy match condition

© 2020 Juniper Networks, Inc. All Rights Reserved.

Regular Expressions and Policies

You can use AS-path regular expressions as the match criteria within a routing policy. This application of the regular expression is similar in concept to the application of a policy. As such, AS-path regular expressions follow the Junos abstraction concept of first defining the entity and then applying the entity.

Format

Both the definition and application of the AS-path regular expressions occurs within the `policy-options` configuration hierarchy. The regular expression in proper *term operator* format is given a name with the `as-path` statement.

AS-Path Name

You can reference the regular expression name within a policy.

Spaces Are Allowed

The name can be up to 255 characters long and can even include spaces, as long as the name is enclosed in double quotation marks. In practice, however, this is not common.

Applied After Definition

Once defined, you can apply the AS-path regular expression as a policy match condition.

Regex Example

- Accept only routes with the exact AS path of 1234 56 78 9

```
[edit policy-options]
as-path digits-route "1234 56 78 9"

policy-statement from-digits-route {
   term digits {
      from as-path digits-route;
      then accept;
   }
   term reject-others {
      then reject;
   }
}

[edit protocols]
bgp {
   import from-digits-route;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

AS-Path Regex Policy Example

The example on the slide shows an AS-path regular expression used as a policy match condition. The goal is to have a BGP routing policy that accepts only routes with the exact AS path of `1234 56 78 9`.

In this example, the AS path contains only terms (no operators) and defines an exact AS-path match of "`1234 56 78 9`".

We then used the digits-route path definition within the `from-digits-route` policy to match routes and accept them. A second term in the policy rejects all other routes.

When you apply this policy as an import BGP policy, only routes matching the AS path defined in `digits-route` are accepted. All other routes seen by BGP are rejected.

Another Regex Example

- Reject any routes with AS 123, 124, or 125 anywhere in the AS path

```
[edit policy-options]
as-path not-a-good-route ".* 123-125 .*"

policy-statement from-not-a-good-route {
   term not-good {
      from as-path not-a-good-route;
      then reject;
   }
}

[edit protocols]
bgp {
   import from-not-a-good-route;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Another AS-Path Regex Policy Example

The example on the slide shows another AS-path regular expression used as a policy match condition. This time, however, the AS path uses both terms and operators.

The goal this time is to reject routes that either originate in ASs 123–125, transited through ASs 123–125, or were advertised directly from ASs 123–125.

To do this, we defined an AS path named `not-a-good-route`. The AS path contains both terms and operators. Thus, `not-a-good-route` defines an AS-path match as follows:

- The AS path starts off with any AS value zero or more times, followed by...
- Any AS value between 123 and 125, followed by...
- Any AS value zero or more times.

The `not-a-good-route` path definition is then used within the `from-not-a-good-route` policy to match routes and reject them.

When you apply this policy as an import BGP policy, only routes matching an AS path defined in `not-a-good-route` are rejected.

Test Your Knowledge

- Consider this policy statement:

```
policy-statement testing-as-paths {
   term as-paths {
      from as-path testing-1-2-3;
      then accept;
   }
   then reject;
```

- Will the router accept a route with the path `1024 44 44 2685`, given the following `as-path` statements?

  - `set as-path testing-1-2-3 ".* 1024"` ✗
  - `set as-path testing-1-2-3 "1024 .*"` ✓
  - `set as-path testing-1-2-3 ".* 1024 .*"` ✓
  - `set as-path testing-1-2-3 ".* 44{1,2} .*"` ✓
  - `set as-path testing-1-2-3 "2685 44{1,3} 1024"` ✗
  - `set as-path testing-1-2-3 "1024 44{1,3} 2685"` ✓

© 2020 Juniper Networks, Inc. All Rights Reserved.

Given a Policy

Consider the routing policy shown on the slide.

Look at the Results

Does this policy, when properly applied, accept a route with the path 1024 44 44 2685, given the `as-path` statements listed on the slide?

Using the different `as-path` definitions within the `testing-as-paths` policy gives the following results:

- `".* 1024"`: Starts with any AS zero or more times, followed by 1024. The route will not match the term `as-paths`. This definition does not allow for AS numbers after 1024. Therefore, it is *rejected* by the final action.
- `"1024 .*"`: Starts with 1024, followed by zero or more AS numbers. The route does match the term `as-paths`. It is *accepted*.
- `".* 1024 .*"`: Starts with any AS zero or more times, followed by 1024, followed by zero or more AS numbers. The route does match the term `as-paths`. It is *accepted*.
- `".* 44{1,2} .*"`: Starts with any AS zero or more times, followed by 44 once or twice, followed by zero or more AS numbers. The route does match the term `as-paths`. It is *accepted*.
- `"2685 44{1,3} 1024"`: Start with AS 2685, followed by 44 one to three times, followed by 1024. The route does not match the term `as-paths`. Therefore, it is *rejected* by the final action.
- `"1024 44{1,3} 2685"`: Start with AS 1024, followed by 44 one to three times, followed by 2685. The route does match the term `as-paths`. It is *accepted*.

AS Regular Expression Enhancements

- Support for **as-path-group**
  - Simplifies AS path-related policy in situations where:
    - You have a large number of individual AS-path regular expressions that are evaluated as a logical OR
    - You have a single large AS-path regular expression that is difficult to understand because of its size

```
[edit policy-options]
user@router# show
policy-statement test-as-group {
    term 1 {
        from as-path-group as-group-1;
        then accept;
    }
    term 2 {
        then reject;
    }
}
as-path-group as-group-1 {
    as-path path_1 ".* 1 .*;";
    as-path path_2 ".* 2 .*;";
    as-path path_3 ".* 3 .*;";
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

AS Path Regular Expression Enhancements

The AS-path regular expression examples shown thus far are perfectly workable. However, circumstances occur where support for `as-path-groups` can save you some configuration work and possible mental anguish. AS-path groups allow you to define a list of individual AS-path regular expressions for evaluation as a logical OR within a policy. The example on the slide shows an AS-path group named `as-group-1`, which comprises three individual AS-path regular expressions. The `as-group-1` AS-path group is referenced in the `test-as-group` policy using a single statement. You can specify multiple AS-path groups as part of a single `from` statement if needed.

Achieving the same result without the `as-path-group` feature requires that you define three separate AS-path regular expressions that are evaluated as a logical OR:

```
[edit policy-options]
user@router# show
policy-statement not-ideal {
    from as-path [ path_1 path_2 path_3 ];
}
as-path path_1 ".* 1 .*";
as-path path_2 ".* 2 .*";
as-path path_3 ".* 3 .*";
```

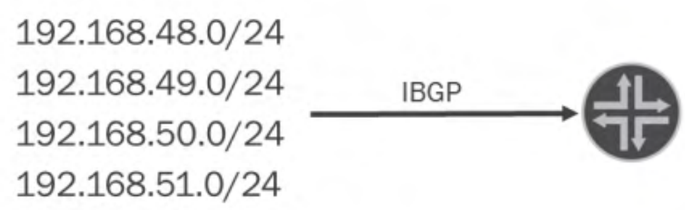
Or, achieving the same result without the `as-path-group` feature requires that you define a single, and possibly quite long, AS-path regular expression:

```
[edit policy-options]
user@router# show
policy-statement not-ideal-either {
    from as-path customers;
}
as-path customers ".* 1 .*|.* 2 .*|.* 3 .*";
```

Using the `as-path-group` feature alleviates both of these issues. The example on the slide is functionally identical to both of the alternatives shown here.

Null AS Path

- Routes that originated in your own AS have no AS numbers in the path yet
- To reference the null AS path within a policy, use the parentheses "()" regular expression




```
user@router> show route protocol bgp terse

inet.0: 42 destinations, 42 routes (42 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A V Destination        P Prf   Metric 1   Metric 2    Next hop      AS path
* ? 192.168.48.0/24    B 170        100          5 >10.222.9.1      I
* ? 192.168.49.0/24    B 170        100          5 >10.222.9.1      I
* ? 192.168.50.0/24    B 170        100          5 >10.222.9.1      I
* ? 192.168.51.0/24    B 170        100         10 >10.222.9.1      I
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Role of Null AS Path

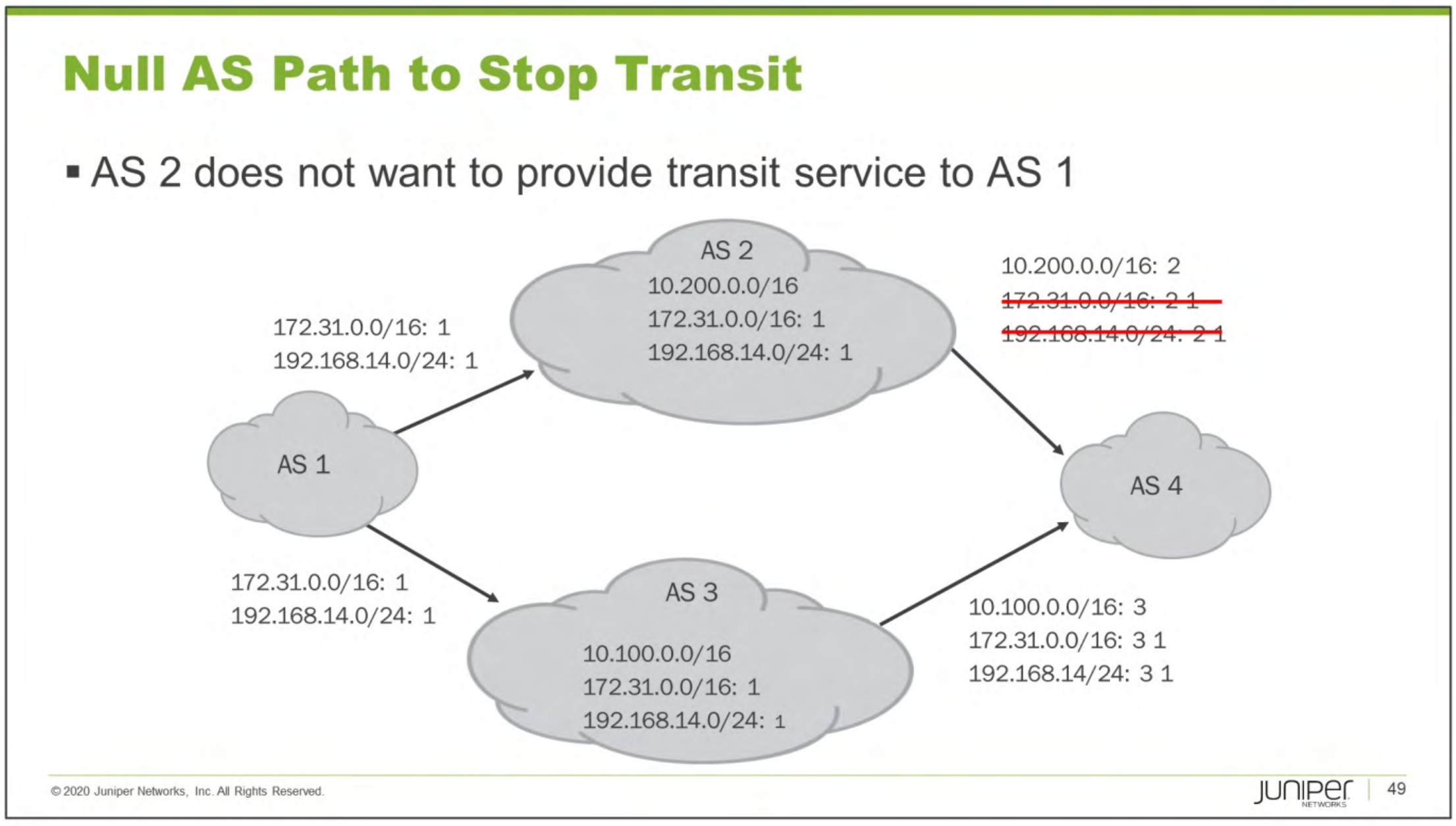
The concept of the null AS path is quite important for the Internet. Routes originating within a particular AS have yet to prepend the BGP AS-path attribute. Therefore, no information is contained in the AS-path attribute for routes originating within the AS, and the AS path is assumed to be null (empty).

So, how are routes originating within the AS that were advertised with BGP to be found with a routing policy using an AS-path policy? They are found by creating a match condition based on the null AS path.

Defining the Null AS Path

To specify the null AS path using a regular expression, use the parentheses with no space in between them.

In the example on the slide, the router receives four routes from IBGP. By examining the routing table, you can see that the AS path column is empty (`I` is for the origin code). Therefore, these routes were sourced within the router's own AS. The null AS path finds these routes.

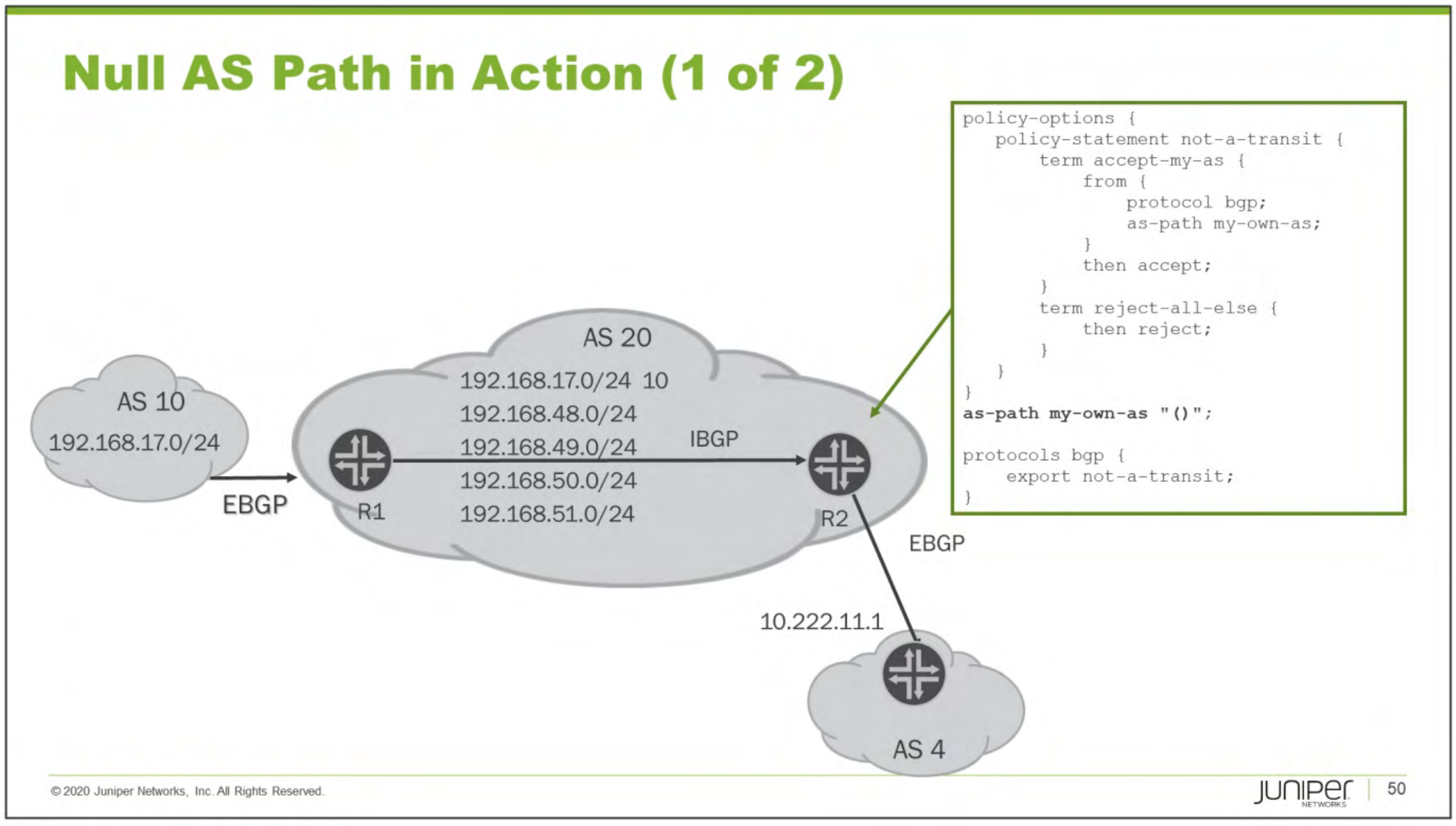


Finding Local Routes

You can use the null AS path to find routes advertised by BGP that originated within the local AS path. When would this be helpful? Usually, when one AS path does not want to carry transit traffic for another AS path.

In the example on the slide, the null AS path is used to halt BGP transit traffic from AS 1 through AS 2. AS 2 does not want to readvertise the routes from AS 1 to AS 4, which could lead to AS 4 routing traffic for AS 1 through AS 2. However, AS 2 still must advertise its own routes to AS 4 so that these prefixes are reachable. AS 2 defines an `as-path` applied as a BGP export policy towards AS 4 that rejects all routes except those with a null AS path.

An example follows on the next page.

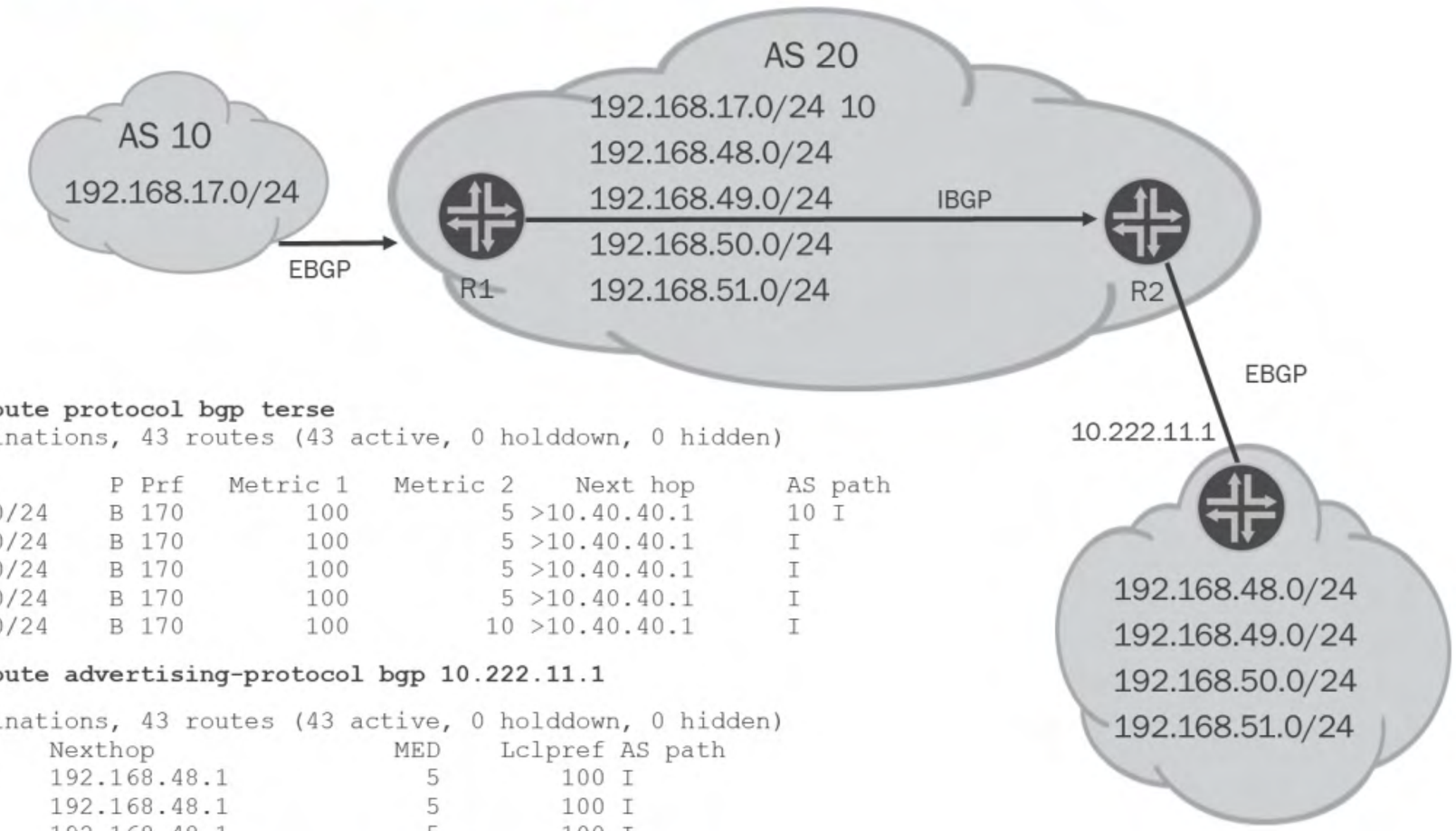


Use of Null AS Path, Part 1

The slide shows an example of null AS path in action.

We applied the `not-a-transit` policy on the slide as an export policy on R2 towards the EBGP peer on the right side of the diagram. The policy has a term called `accept-my-as` that matches BGP routes with an `as-path` regular expression of `()`. The `reject-all-else` term rejects all other routes.

Null AS Path in Action (2 of 2)




```
user@R2> show route protocol bgp terse
inet.0: 43 destinations, 43 routes (43 active, 0 holddown, 0 hidden)

A V Destination        P Prf   Metric 1   Metric 2    Next hop      AS path
* ? 192.168.17.0/24    B 170        100          5 >10.40.40.1      10 I
* ? 192.168.48.0/24    B 170        100          5 >10.40.40.1      I
* ? 192.168.49.0/24    B 170        100          5 >10.40.40.1      I
* ? 192.168.50.0/24    B 170        100          5 >10.40.40.1      I
* ? 192.168.51.0/24    B 170        100         10 >10.40.40.1      I

user@R2> show route advertising-protocol bgp 10.222.11.1

inet.0: 43 destinations, 43 routes (43 active, 0 holddown, 0 hidden)
Prefix             Nexthop              MED    Lclpref AS path
192.168.48.0/24    192.168.48.1           5        100 I
192.168.49.0/24    192.168.48.1           5        100 I
192.168.50.0/24    192.168.48.1           5        100 I
192.168.51.0/24    192.168.48.1          10        100 I
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Use of Null AS Path, Part 2

The slide shows the results of the previously configured and applied policy.

The **`show route protocol bgp`** output shows that the router R2 receives five BGP routes from its IBGP peer R1. One of these routes is from AS 1, and the other four are sourced from within the AS.

The **`show route advertising-protocol bgp`** output shows that only the four routes sourced from within the AS are sent to the EBGP peer. The null AS-path regular expression in the routing policy rejects transit routes.

Summary

- In this content, we:
  - Explain how policies function in BGP
  - Describe BGP attributes and explain how these attributes can be used to manipulate traffic
  - Show how Regex can be used in policies to manipulate AS-path

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- How policies function in BGP;
- BGP attributes and explained how those attributes can be used to manipulate traffic; and
- How Regex can be used in policies to manipulate AS-path.

Review Questions

1. Between which two RIBs does an import policy operate? What about an export policy?
2. What BGP attribute does Junos evaluate first in the BGP route selection process?
3. What is the intent of prepending a local AS number multiple times to all BGP routes?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

3.

Lab: BGP Attributes – Part 1

- Configure and manipulate BGP attributes

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: Configure and Manipulate BGP Attributes

This slide provides the objectives for this lab.

Answers to Review Questions

1.

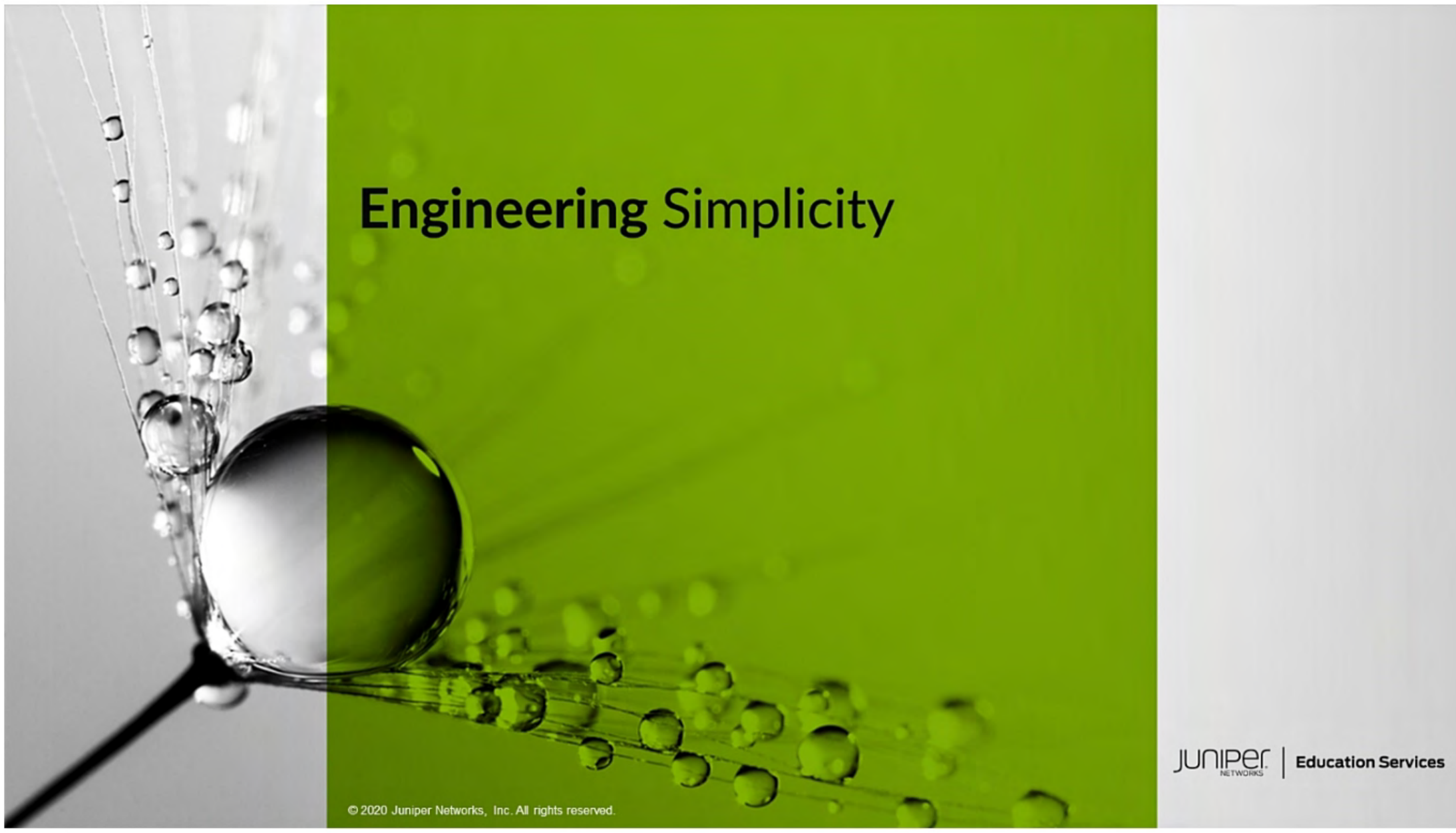
An import policy operates between the RIB-IN and the RIB-LOCAL. An export policy operates between the RIB-LOCAL and the RIB-OUT.

2.

The Junos OS uses local preference as the first selection criteria.

3.

The purpose of prepending the AS PATH is to make the route advertisement look undesirable.


Engineering Simplicity
© 2020 Juniper Networks, Inc. All rights reserved.
Juniper Networks
Education Services

Education Services

Advanced Junos Service Provider Routing

Chapter 12: BGP Attributes and Policy—Part 2

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - Describe the BGP attributes Origin, MED, and communities in detail and explain the operation of those attributes
  - Manipulate these BGP attributes using routing policy

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- The BGP attributes Origin, MED, and communities and explains the operation of those attributes; and
- The manipulation of those BGP attributes using routing policy.

Agenda: BGP Attributes and Policy—Part 2

→Origin
- Multi-Exit Discriminator
- Communities

© 2020 Juniper Networks, Inc. All Rights Reserved.

Origin

The slide lists the topics we will discuss. We discuss the highlighted topic first.

The Origin Code

- Installed by the originating router for the prefix (route)
- A tag of believability as to the origin of the route information *(Where did you get it from?)*
- BGP origin code is a well-known, mandatory attribute
- Origin can be internal, external, or unknown
  - I: Internal (0)—Learned from an IGP
  - E: External (1)—Learned from EGP
  - ?: Incomplete (2)—NLRI found by some other means
- I (0) is better than E (1), which is better than ? (2)
- All Junos OS BGP routes have origin IGP by default

© 2020 Juniper Networks, Inc. All Rights Reserved.

Set by Router Originating the Information

The first router to inject the route into BGP attaches the origin attribute to a prefix (that is, a route). Other routers can change this value, of course, as the route makes its way through an AS and on to other ASs.

How Believable Is This Information?

The intent of the origin code is to provide a measure of *believability* as to the origin of a particular route. In other words, the intent is to provide a kind of *where did you get this from?* clue for other routers seeing the route.

Well-Known Mandatory BGP Attribute

The origin code is a well-known, mandatory BGP attribute (the Type Code is 1), meaning that each route associated with BGP must have an origin code assigned to it.

Internal, External, or Incomplete

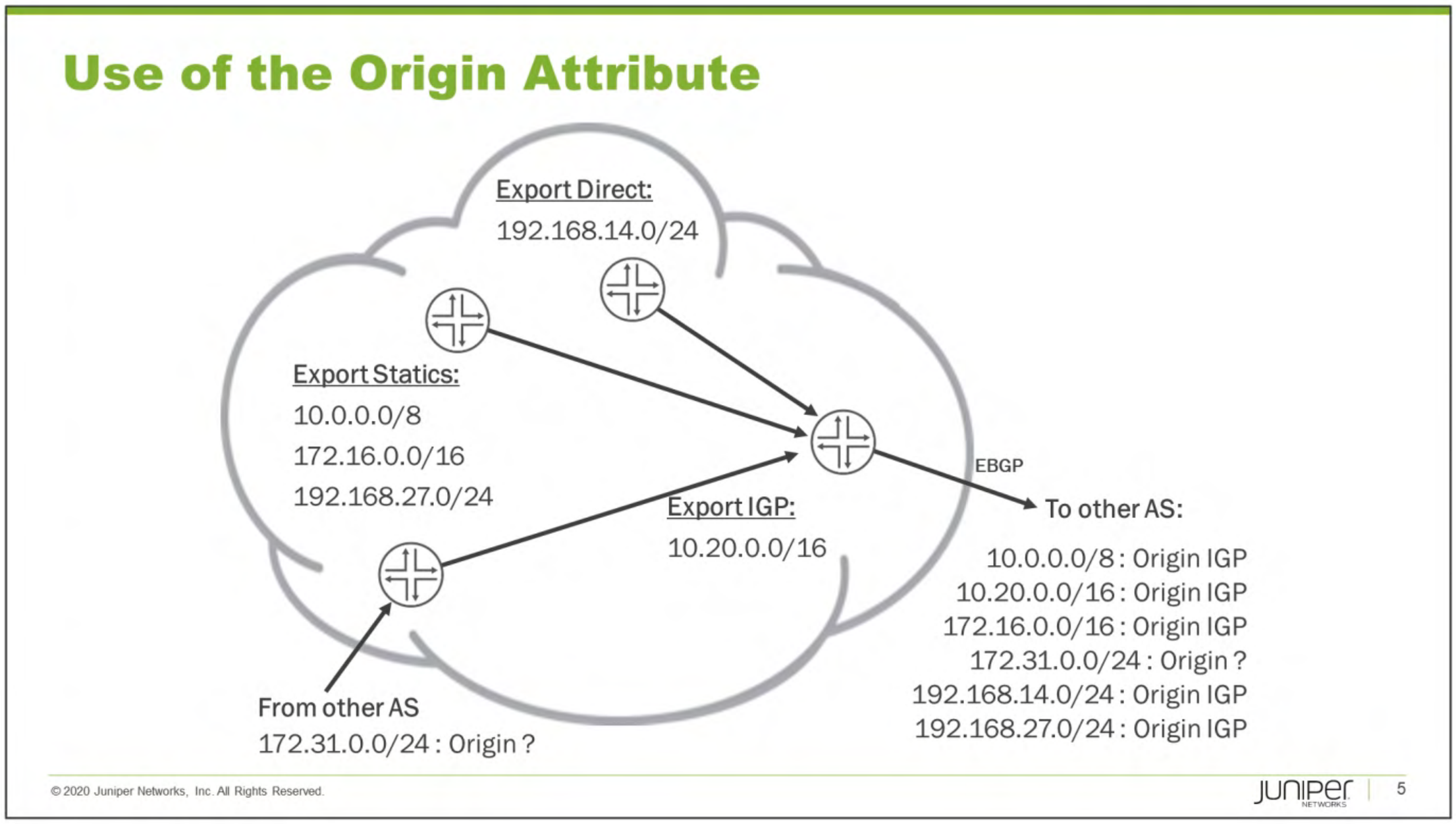
The values available for the origin attribute are internal, external, and incomplete. The internal (I) origin is a tag designated for all routes learned through a traditional IGP, such as OSPF, IS-IS, or RIP. These types of routes were typically seen as *best* sources of information because of their stability at the time BGP was created. The external (E) origin is a tag designated for routes learned through the original Exterior Gateway Protocol (EGP). This protocol was the precursor to BGP but was not as robust, and it was generally less reliable than the IGP routes. The last origin code of incomplete (?) is a tag designated for all routes that did not fall into either the internal or external categories.

Internal Better than External Better than Incomplete

Each of the origin tags is assigned a value for use in transmitting the attribute to other BGP speakers. The values are 0 for internal origin (I), 1 for external origin (E), and 2 for incomplete origin (?). A lower value is better, so routes learned from an IGP are preferred over routes learned from an EGP. EGP routes are better than incomplete routes.

The Junos OS Default Is Internal—IGP

Because all routing information eligible to be injected into BGP on a Juniper Networks router resides in inet.0, the Junos OS sees all possible routes as internal routes. These internal routes all receive a BGP origin code of internal when placed into BGP.



How Origin Is Used

The slide shows the default BGP behavior within the Junos OS with regard to the origin code.

Within the AS shown, the static routes of 10.0.0.0/8, 172.16.0.0/16, and 192.168.27.0/24 exist. An export policy placed these routes into BGP. A direct route of 192.168.14.0/24 exists that was exported into BGP. The route to 172.31.0.0/24 was learned from another AS altogether, and this route contains an origin attribute coded to 2, which indicates an unknown origin. Finally, an IGP-learned route of 10.20.0.0/16 exists in the network. The router does not know whether this route is an OSPF route or an IS-IS route, but the route had the appropriate export policy and, therefore, was placed into BGP.

To the Juniper Networks router, it does not matter that these routes are advertised to another AS through EBGP; the BGP origin code is not altered as the routes are advertised to an EBGP peer. On the basis of the origin attribute alone, the 172.31.0/24 prefix appears less attractive to the remote AS.

Changing the Origin Code

- Find IGP origin codes and change these origin codes to incomplete

```
[edit policy-options]
policy-statement change-all-igps {
   term igp-to-incomplete {
      from {
         protocol bgp;
         origin igp;
      }
      then origin incomplete;
   }
}

[edit protocols]
bgp {
   export change-all-igps;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Modifying Origin with a Policy

The slide shows an example of a BGP export policy that changes the origin code. This example finds all BGP routes and changes the origin code from internal (0) to incomplete (2).

The match criteria for the policy are all active BGP routes that currently have an origin code of internal. The action for the policy is to change the origin code to incomplete. Once applied as a BGP export policy and committed, this policy starts altering the origin code for all BGP routes advertised to all BGP peers.

The Junos OS has specific keywords to represent the different BGP origin codes. They are:

- `igp`: Internal (value 0);
- `egp`: External (value 1); and
- `incomplete`: Incomplete (value 2).

Agenda: BGP Attributes and Policy—Part 2

- Origin
- →Multi-Exit Discriminator
- Communities

© 2020 Juniper Networks, Inc. All Rights Reserved.

Multi Exit Discriminator

The slide highlights the topic we discuss next.

Multi-Exit Discriminator

- An optional, nontransitive attribute, MED is never passed through one AS to another AS
- A neighboring AS can use MED to prefer one of several paths to the local AS
- Informs neighboring AS which ingress path to use to reach the local AS in an attempt to influence inbound traffic
- Can perform some primitive load balancing
- Routes that are redistributed into BGP will have a BGP MED value equal to the metric of the original route
- Other AS networks can preempt MED with other BGP attributes

© 2020 Juniper Networks, Inc. All Rights Reserved.

MED Is Optional, Nontransitive

The BGP multiple exit discriminator (MED) attribute is an optional, nontransitive attribute of BGP. Thus, a BGP implementation does not have to understand or use MEDs at all, and a MED is not sent through one AS and on to another AS. MED values can be added to BGP routes via policy, configuration commands, or through route redistribution. Policies and configuration commands can also manipulate MED values.

Used with Multiple Inter-AS Links

The MED is a type of routing metric assigned to BGP routes. The function of the MED is to assist a neighboring AS to pick which of multiple links connecting the remote AS to the local AS (ingress paths) to use for traffic to a particular route (prefix).

Seeks to Influence Inbound Traffic

MEDs are an attempt by the local AS to influence the routing decisions in the remote AS for traffic *inbound* to the local AS, just like the origin attribute. And just like the origin attribute, MEDs are a negative-bias mechanism to make some paths look worse than others.

Primitive Load Balancing

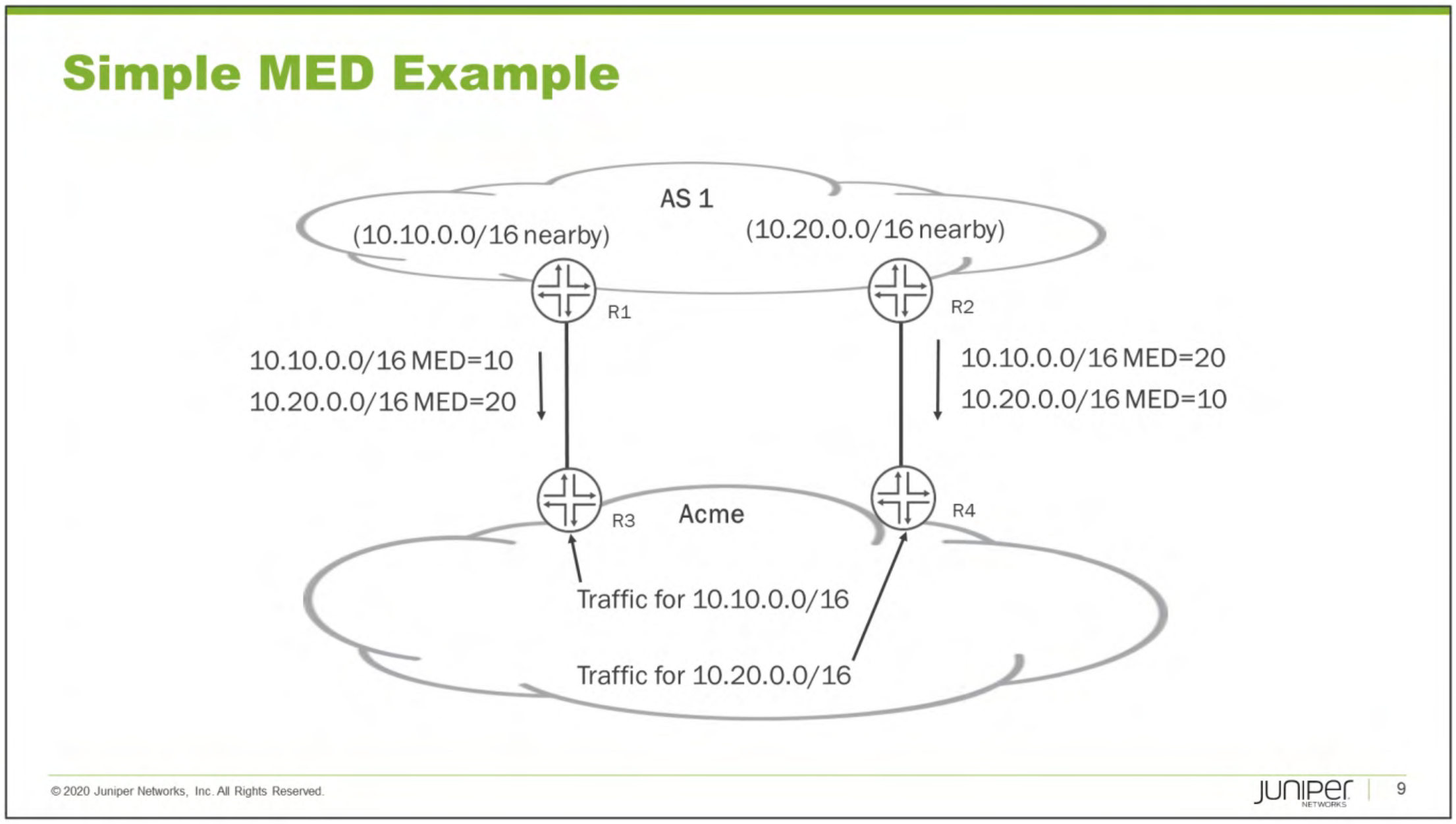
You can use MED values to perform some form of primitive load balancing between ASs with multiple links between them. However, the use of MEDs for load balancing is neither efficient nor particularly effective, compared to more sophisticated mechanisms available.

MED and IGP Metric

When a route is redistributed into BGP, the new BGP route will set its MED value to be equal to the metric of the original route. If the original router is an EBGP router, then the MED will be advertised to all IBGP and EBGP peers. If the original router is and IBGP router only the MED will be sent to all IBGP peers, however those peers will remove the MED before sending the route outside the AS.

Effects Not Guaranteed

Despite the best efforts on the part of a local AS to manipulate MED values to influence inbound traffic flows to the local AS, other ASs can always preempt, or even ignore, the MED. This reaction is not only because the MED is an optional BGP attribute, but also because several other BGP attributes are more important than MED in the BGP route selection process.



Simple MED Use

The slide shows a very basic example in the use of the BGP MED attribute to influence inter-AS traffic flows.

AS 1 assigned its IP address spaces so that it can summarize its network into two major segments. Furthermore, AS 1 is relatively cleanly divided into networks near the left most router (10.10.0.0/16 networks are nearby) and networks near the right most router (10.20.0.0/16 networks are nearby). Perhaps the split is between Eastern and Western operations, but there are many other alternatives.

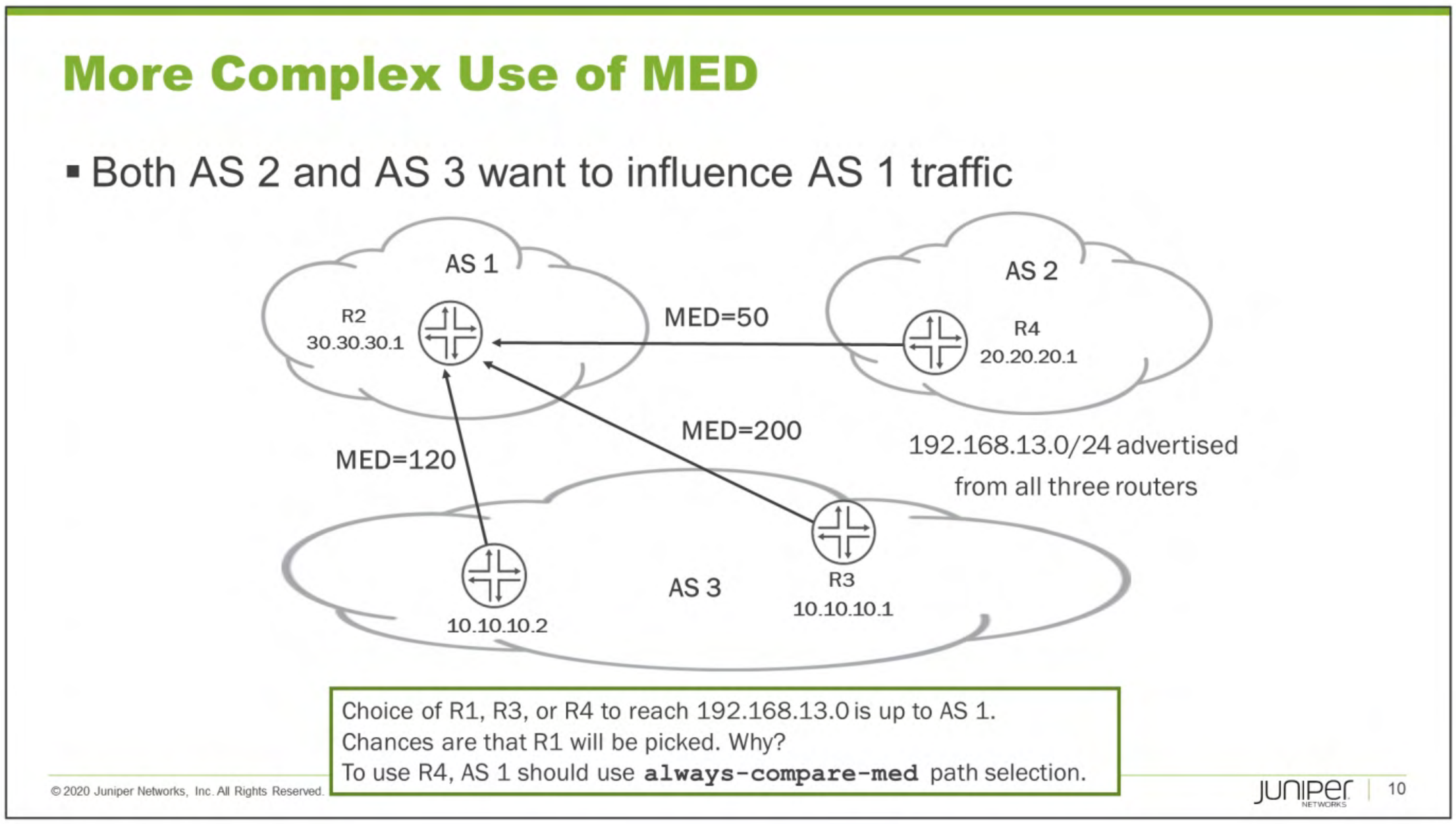
AS 1 has two EBGP sessions to the customer Acme and advertises both the 10.10.0.0/16 and the 10.20.0.0/16 networks to Acme on each EBGP session, as shown on the slide.

Naturally, AS 1 would like Acme to return traffic to the closest point in the network so that *timely* packet delivery and low latency can be achieved. Ordinarily, AS 1 would have no real way to convey this desire to Acme, and Acme would simply send traffic to AS 1 over whichever router Acme decides to use based on its own routing policies.

However, the MED attribute offers a method for AS 1 to influence the routing policy on Acme for traffic sent to AS 1. To accomplish this *closest point* goal, AS 1 alters the MED values on the routes that it advertises to Acme with a BGP export routing policy.

Both of the networks in AS 1 are advertised across both links for redundancy. All things being equal, the routers in Acme still see multiple network paths to the routes in AS 1 as the AS 1 routes are passed along throughout Acme. The Acme routers, however, use the MED values of 10 and 20 (10 being preferred) to choose the BGP path to install in their local routing tables.

Thus, within Acme, traffic to 10.10.0.0/16 networks flows to the left most router and out to AS 1, while traffic to 10.20.0.0/16 networks flows to the right-most router and out to AS 1. AS 1 influenced Acme, and, at the same time, achieved a primitive type of load balancing.



Sophisticated MED Use

The use of the MED attribute is straightforward when adjacent pairs of ASs are considered. The use of the MED attribute becomes a little more complicated when more than one AS is involved.

Consider the AS networks on the slide. Both AS 2 and AS 3 advertise the 192.168.13.0/24 route to AS 1 and want to influence the way AS 1 sends traffic to 192.168.13.0/24. All three of the advertisements have identical local-preference values, AS-path lengths, and BGP origin codes. The other IP addresses on the slide represent the router IDs of the three routers in R1, R3, and R4.

The MED value from the AS 2 router is the lowest among the three advertisements. Yet, when the router in AS 1 chooses the BGP path to use in its local routing table, the AS 1 router most likely chooses R1 in AS 3. Why should this be?

The problem in this scenario is that the default evaluation of the MED attribute only happens when route advertisements come from the *same* neighboring AS. In this scenario, only two of the three advertisements come from the same AS: those from R3 and R1. Between those two advertisements, the route from R1 is the best, because of its lower MED value.

The route from R4 in AS 2 cannot be compared to the two routes from AS 3 because it is from a different AS. At this point, AS 1 is left with the R1 and the R4 routes. The R1 route will most likely be selected as the active path because this router has a lower router ID than R4.

However, the routers in AS 1 can compare the MED values for all three of these routes with the use of the **`always-compare-med`** configuration statement. With this configuration, the path to 192.168.13.0/24 through R4 should be chosen, based on the lowest MED value.

Path Selection and MEDs

- By default, the Junos OS uses a deterministic MED comparison scheme for routes from the same AS
- **`always-compare-med`** compares MED values, regardless of whether the neighboring AS is the same
  - Use with caution—every network has a different interpretation of a *good* MED
- **`cisco-non-deterministic`** compares paths based on when they are received
  - *Not* recommended for use in your network
  - Can cause incorrect route selections

```
[edit]
user@router# set protocols bgp path-selection ?
Possible completions:
  always-compare-med   Always compare MED values, regardless of neighbor AS
  cisco-non-deterministic  Use Cisco IOS nondeterministic path selection algorithm
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Always Compare MEDs

By default, only the MED values from the same neighboring AS are compared to select a BGP path. However, the **`always-compare-med`** configuration statement allows you to override the default BGP behavior for MED evaluation.

When configured, *all* routes that have the shortest AS-path length are compared to each other to determine the route with the smallest MED value, not just routes from the same AS. The route with the lowest MED value is then selected as the active BGP path, regardless of the AS the route came from. The lowest MED value is selected as long as other path selection values for the route, such as local preference, are the same.

You must be cautious when comparing MED values from different ASs. Some inherent danger exists when using the **`always-compare-med`** configuration option to compare MED values from more than one AS because every AS in the Internet can set its own *good* and *bad* values for MEDs.

One AS might consider a MED of 50 as the best, while another AS might consider a MED of 5 to be good. To complicate matters further, some AS networks might not set the MED value at all (MEDs are optional), which essentially sets the MED value to 0.

Agenda: BGP Attributes and Policy—Part 2

- Origin
- Multi-Exit Discriminator
- →Communities

© 2020 Juniper Networks, Inc. All Rights Reserved.

Communities

The slide highlights the topic we discuss next.

BGP Communities

- BGP attribute (optional) passed along to other BGP peers (transitive)
- A group of destinations that share a common property
- Simplify routing policies by identifying routes based on logical bounds you establish
  - AS number is very broad (lots of routes)
  - IP prefix is very granular (route filter for each route)
- Used with other attributes to accept, prefer, or advertise BGP routes

© 2020 Juniper Networks, Inc. All Rights Reserved.

Optional, but Transitive

The BGP community attribute is an optional attribute, so not all implementations of BGP must recognize and use communities. However, if BGP communities are associated with a BGP route, the BGP community attribute must be passed along to all other BGP peers, both within an AS and between AS networks (transitive).

Routes Having Something in Common

The BGP community attribute is an administrative tag that you can use to associate routes that share common properties. The BGP community attribute is not complex. The main role of the BGP community attribute is to make it easy to group routes that should be treated the same by a routing policy. BGP communities are very flexible: a BGP route can belong to many BGP communities.

Make Routing Simpler

The attraction of BGP communities is that they can simplify routing policies. BGP communities identify routes based on the logical boundaries you establish and not what the AS number (too broad in most cases) or individual IP prefix (too granular in most cases) establishes.

Use with Other BGP Attributes

You can use the community attribute values with other BGP attributes to implement routing policies that accept, prefer, or advertise BGP routes.

Notes on Communities (1 of 2)

- Establishes *categories* for routes and prefixes
- Often sets local preference for a group
- Cuts down on manual reconfiguration and complexity of maintenance
- If a new prefix is placed in a community, no other changes to routing policy are necessary
- Too many communities require more manual maintenance
- Too many overlapping communities can be a nightmare

© 2020 Juniper Networks, Inc. All Rights Reserved.

Groups for Routes

BGP communities establish various logical categories for routes (prefixes). Think of BGP communities as *communities of interest* or even *groups* for routes. And just as people can belong to more than one group, routes can fit into more than one BGP community.

Can Carry Local-Preference Value Beyond Local AS

The BGP community attribute value is often used in policies within customer networks, to perform functions such as altering the local-preference attribute on incoming routes.

Goal: Less Work

The community attribute can assist you in the configuration and maintenance of policies. This cuts down on the time needed for manual reconfiguration and the complexity of the overall maintenance task.

Community Use Example

As an example, consider the addition of a new prefix for a new customer within an AS. If you use communities to enforce routing for customer networks, and you can place the new prefix into an existing community, you do not need any changes to overall routing policies. When new customers are brought into the network, you must only assign the new routes the proper community attribute. Without the use of communities, you might need to update each policy in the network to include the new customer information.

Too Many Communities

However, you must be careful when establishing communities for the first time. Too many communities defeat the whole purpose. Maintenance of the communities then becomes as tedious as maintaining a whole list of route filters.

Overlapping Communities

You should also avoid having too many overlapping communities. Customer routes belonging to multiple communities can also be an issue.

When it comes to communities, simplicity is always a worthwhile goal.

Notes on Communities (2 of 2)

- Well-known communities have a global meaning
- Must define local-use communities
- Community attribute is a list of four-octet, individual community attribute values associated with a route
  - Route can belong to many communities
  - Two high-order octets represent an AS number
  - Two low-order octets represent a value unique to that AS
  - Represented in decimal form of *AS:number* (for example: 200:123)
  - All values in AS 0 and 65535 are reserved
  - 4 Byte ASNs have the format: 1234567L:456 where L indicates the long format
  - 4 Byte ASNs can also be represented using format: 12345.12345:789

© 2020 Juniper Networks, Inc. All Rights Reserved.

Well-Known Communities

Certain *well-known communities* have a global meaning and special purposes. They are defined in RFC 1997. All BGP implementations that understand communities (communities are optional, but transitive) must know these communities and respect their functions.

Communities for Local Use

BGP communities that are not well-known are for local use. You must define local-use communities on each router where they will be used. The BGP community attribute follows the BGP route wherever it goes. Even local-use communities are circulated outside the AS. So, you must take care in using BGP community attribute values arriving from other AS networks. BGP communities are best thought of as a category for a group of routes (such as, *all customers*).

Community Format

The BGP community attribute itself is just a list of the individual community attribute values associated with the route (*tags*). Because no real limit exists on the number of tags in the list, a route can belong to many communities. No predefined, upper limit exists on the number of communities allowed on a route.

The community attribute values themselves are designed so the values can be uniquely assigned in the global Internet. The BGP community attribute itself is a 32-bit (4-octet) value that has two parts. The two high-order octets (16 bits) form the first part and are set aside for the AS number of the network that assigned the community to the route in the first place. The two low-order octets (16 bits) form the second part and are set aside for use by the specific AS networks. Because the AS value is included in the community, the value is guaranteed to be unique on the whole Internet.

When written in bits or hexadecimal format, community values can look very odd. Thus, communities are usually represented in decimal form as *AS:number*. For example, 200:123 is community 123 in AS 200. One restriction on possible community values is that the AS values of 0 and 65535 are reserved and cannot be assigned to a route. RFC 4360 provides support for an “extended community attribute”. An extended community attribute consists of eight octets. The BGP extended communities attribute format has three fields: *type:administrator:assigned-number*.

4 byte AS numbers can be represented in the format ASL:number or AS.AS:number (for example: 1234567L:123 or 12345.12345:123). The L must be capitalized and is added to the end of the 4 byte AS number.

Current releases of Juniper support the 4 byte format. If you are running an older version of Junos or using another vendor, these 4 byte formats may not be available.

Well-Known Communities

- Three standard values:
  - No-export (0xFFFFFF01): These routes must be distributed within the confederation (or AS), but no farther
  - No-advertise (0xFFFFFF02): These routes must not be advertised to other BGP peers
  - No-export-subconfed (0xFFFFFF03): These routes must not be advertised to external BGP peers (confined to sub-AS)
- No-export typically keeps aggregation optimal
- No-advertise has narrower scope
  - Often used when dual links exist between two routers
- Well-known communities must be defined as named objects in Junos in order to use them in policies

© 2020 Juniper Networks, Inc. All Rights Reserved.

The Three Well-Known Communities

Three community attribute values are designated as well-known communities. These well-known communities share the AS value of 65535 (0xFFFFxxxx). The three communities are:

- *No-export (0xFFFFFF01)*: This value tells routers to distribute routes with this community tag within the confederation or AS, but no farther.
- *No-advertise (0xFFFFFF02)*: This value tells routers not to advertise these routes to other BGP peers at all. (Despite appearances, this BGP community is quite useful.)
- *No-export-subconfed (0xFFFFFF03)*: This value tells routers not to distribute routes with this community tag to external BGP peers (thus, they are confined to the sub-AS).

No-Export

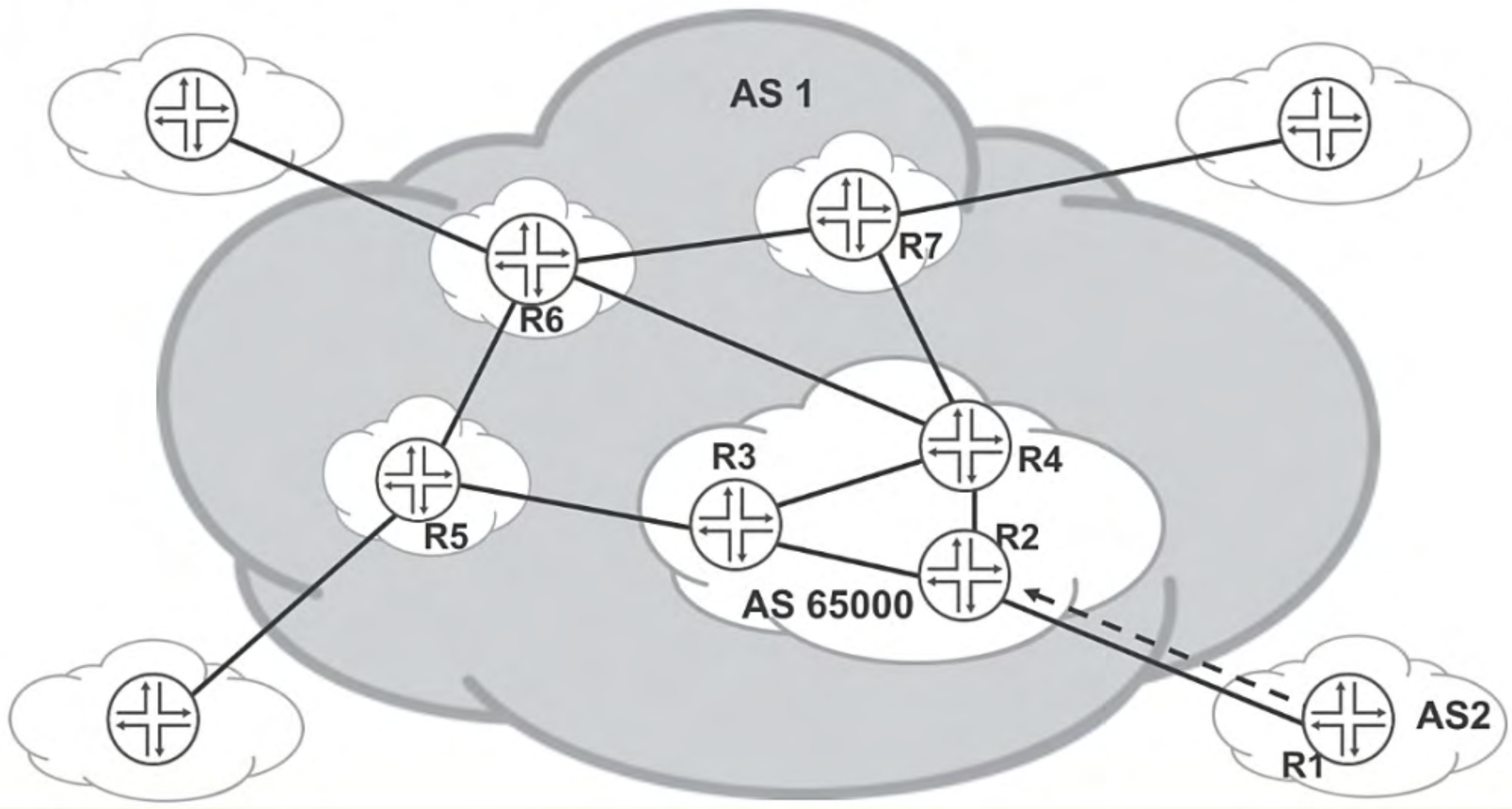
The no-export community typically makes sure that route aggregation is optimal by suppressing more specific routes. The no-export-subconfed just extends this aggregate concept to a confederation's sub-AS.

No-Advertise

The no-advertise community has a very narrow scope. Routes go to a BGP peer and no farther, usually because peers know the routes through other means. This community is often used in a LAN-connected router environment or when two BGP peers have multiple links between them.

No-Advertise

- Routes go no further than the next hop

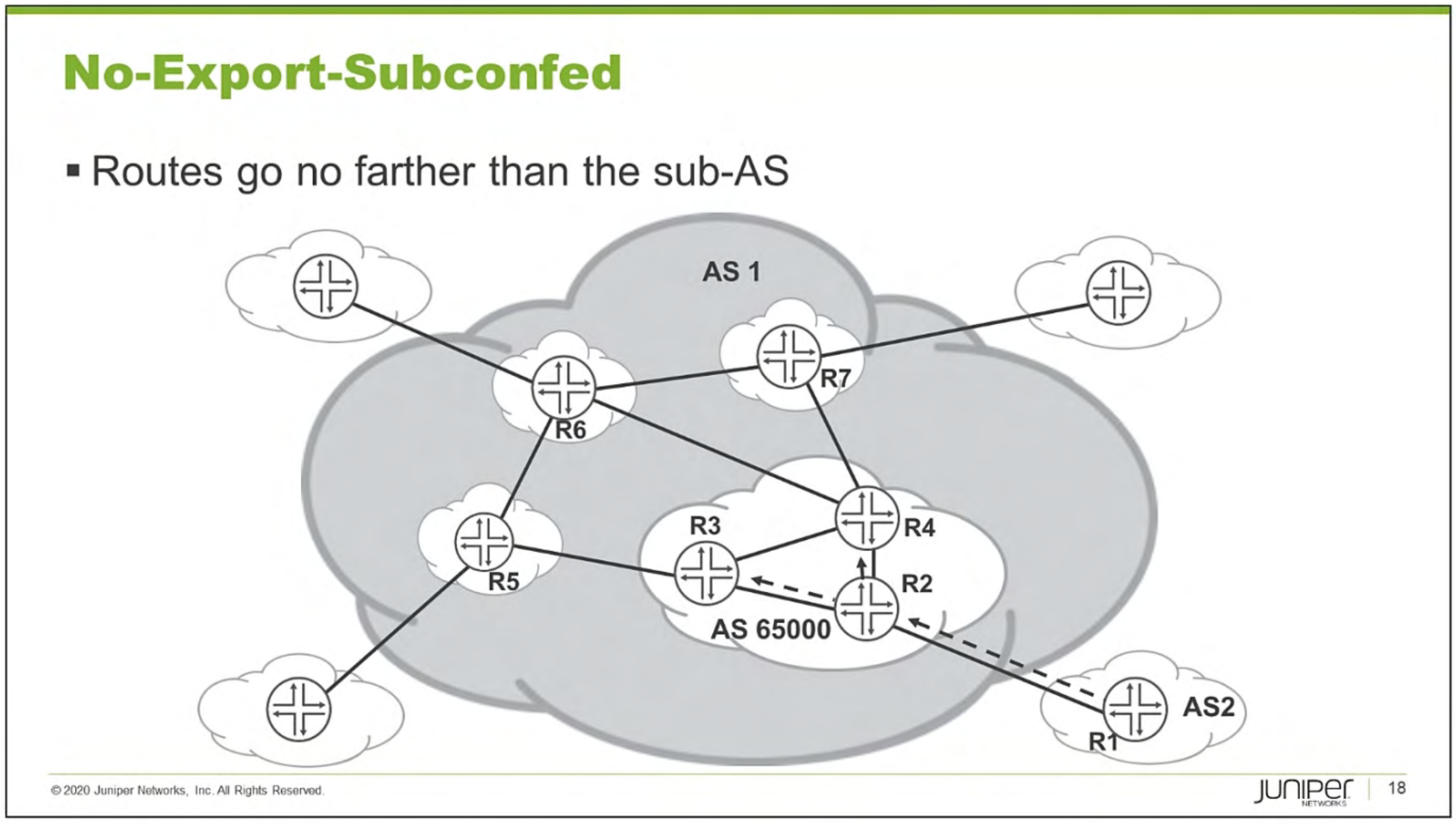


© 2020 Juniper Networks, Inc. All Rights Reserved.

Sample Use of No-Advertise

The slide shows an example of the scope of the no-advertise community. The arrow at the lower right shows a BGP route with the no-advertise community injected by R1 into an AS with sub-confederations.

The well-known community attribute value of no-advertise is designed such that a route can be sent to a single BGP peer and be advertised no farther. Routes are restricted to the next-hop router R2.



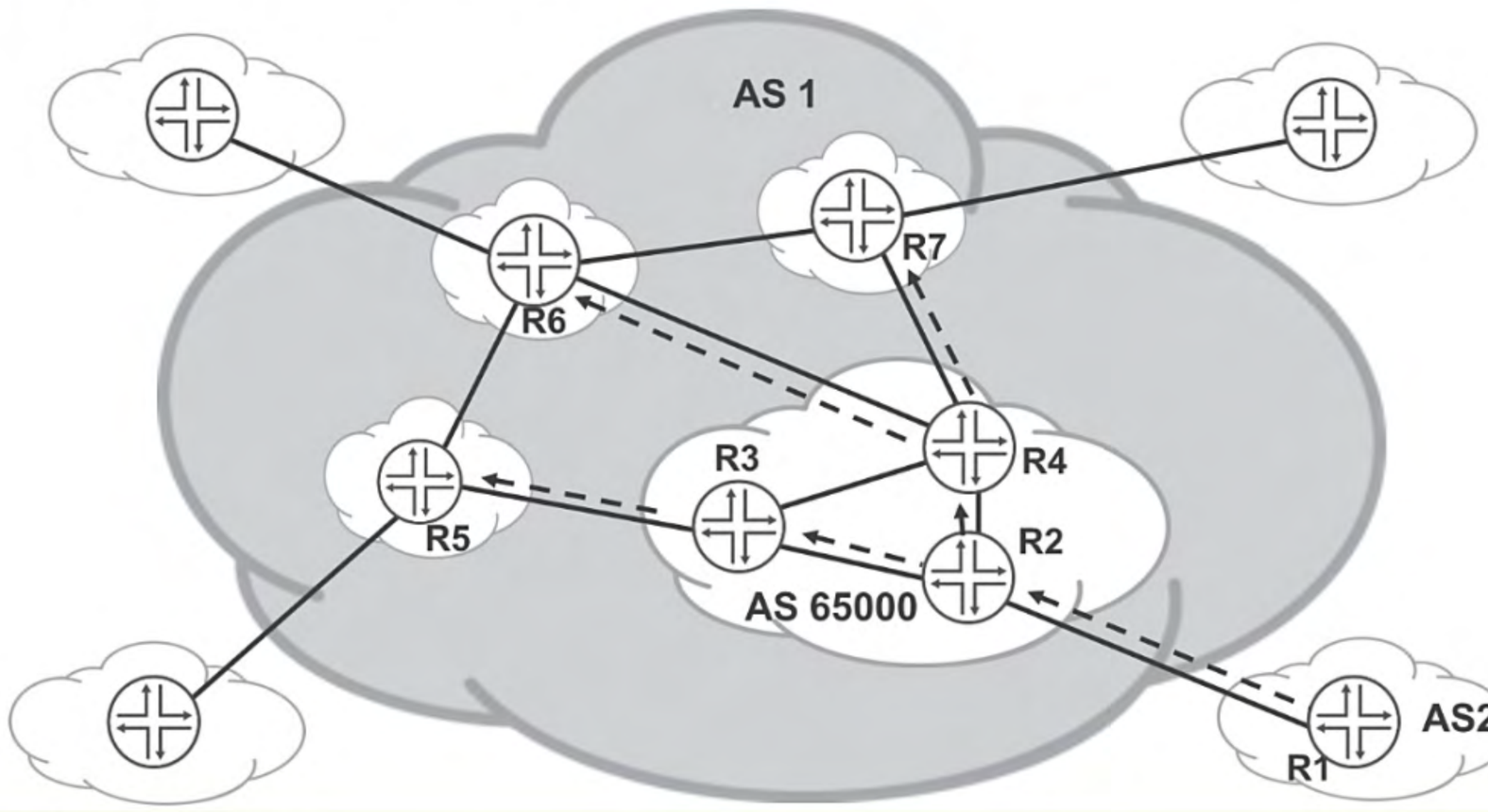
Sample Use of No-Export-Subconfed

The slide shows an example of the scope of the no-export-subconfed community. The arrow at the lower right shows a BGP route with the no-export-subconfed community injected by R1 into an AS with sub-confederations.

The well-known community attribute value of no-export-subconfed is designed such that a route can be sent into a BGP confederation network and have the information remain with a particular sub-AS. The routing information is advertised no farther than the sub-AS routers R3 and R4, as shown on the slide.

No-Export

- Routes go no farther than the entire AS

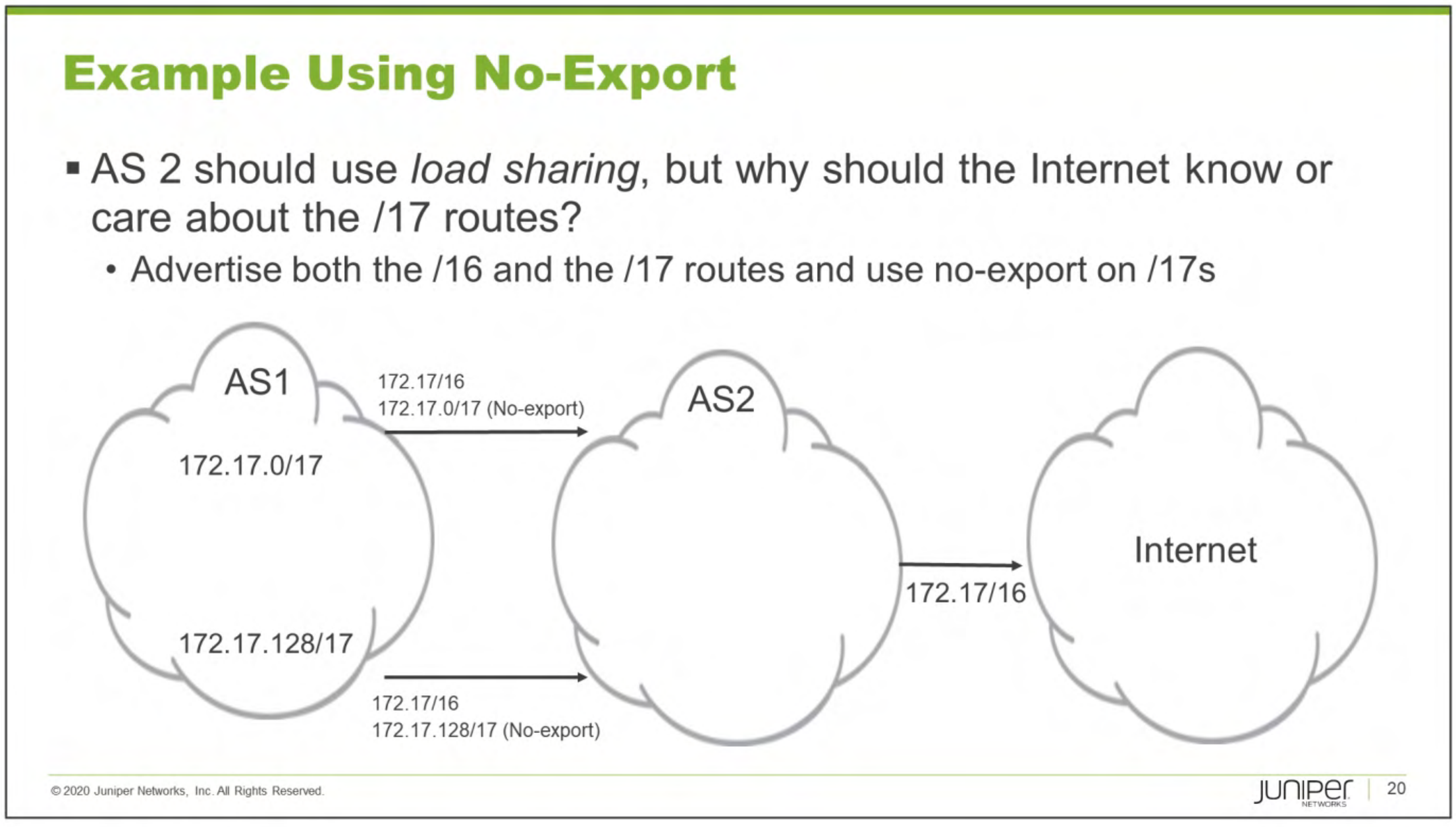


© 2020 Juniper Networks, Inc. All Rights Reserved.

Sample Use of No-Export

The slide shows an example of the scope of the no-export community. The arrow at the lower right shows a BGP route with the no-export community injected by R1 into an AS with sub-confederations.

The well-known community attribute value of no-export is designed such that a route can be sent into a neighboring AS and have the information remain within that neighboring AS. Normally, BGP communities are transitive and are passed from each AS to all others, even if the router does not support or use the BGP community option. However, the routing information with the no-export community tag is not advertised beyond the AS routers R5, R6 and R7, as shown on the slide.



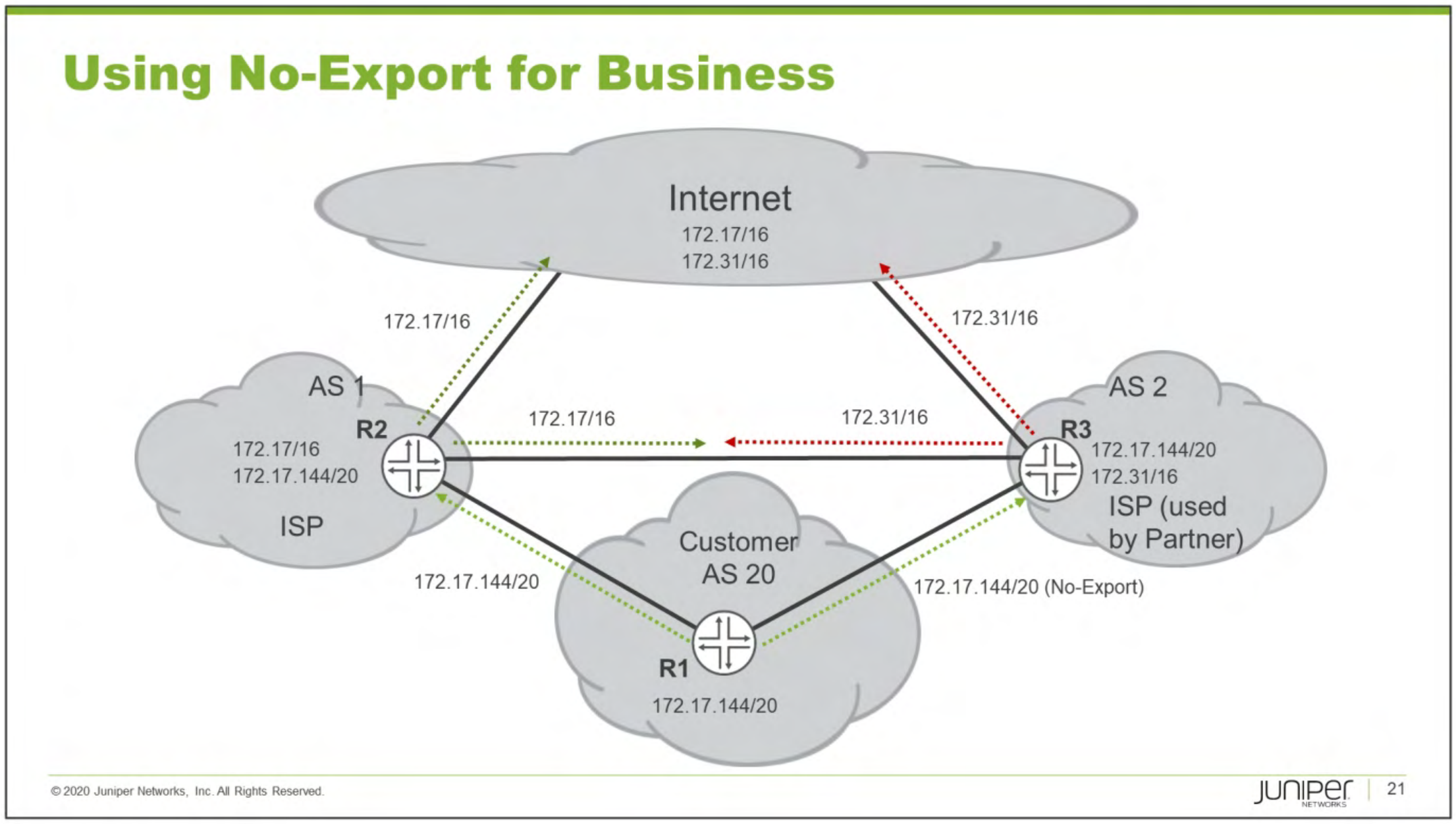
No-Export Example

The slide shows a typical use of the BGP no-export community.

AS 1 has multiple BGP sessions to its neighbor, AS 2. AS 1 also uses AS 2 as a transit AS for connectivity to the Internet. AS 1 wants to advertise 172.17.0.0/16 to the Internet because AS 1 owns that entire address space. In addition, AS 1 also wants to advertise more specific route information (shown as 172.17.0/17 and 172.17.128/17) only to its peer, AS 2.

Advertising the specifics as well as the aggregate to AS 2 would assist AS 2 to route user traffic into AS 1 more efficiently because *load sharing* could be used on the more specific routes, as shown on the slide. However, why should the whole Internet know or care about these specifics?

To assist AS 2 in finding and rejecting the more specific routes, AS 1 assigns the no-export community to the 172.17.0.0/17 and the 172.17.128.0/17 routes. The BGP edge routers in AS 2 that connect to the Internet automatically suppress and do not readvertise the /17 routes. Only the /16 route is readvertised to the Internet.



No-Export and Multihoming

The BGP well-known community no-export is also useful when a customer is multihomed to two ISPs. In the example on the slide, the customer wants to receive Internet traffic through one main ISP, but be able to receive locally originating traffic from the other ISP (perhaps a major trading partner uses AS 2 as its ISP).

One way to do this (among other ways) is with the BGP no-export community. In this example, the customer has AS 1 as its main ISP.

Customer AS 20 has address space 172.17.144.0/20, taken from AS 1's address space of 172.16.0.0/16. The 172.17.144.0/20 route is advertised with BGP to both AS 1 and AS 2. Because AS 2 is not the primary ISP, and only traffic originating in AS 2 should reach AS 20 directly, the route advertised to AS 2 is given the BGP no-export community. This route goes no farther than AS 2. AS 2 advertises 172.31.0.0/16 to AS 1 and the Internet, but does not advertise 172.17.144.0/20.

AS 1 covers the 172.17.144/20 address space with aggregate 172.17.0.0/16, as shown on the slide. This is advertised to AS 2 and the Internet. Thus, AS 20 is reachable through AS 1 from the Internet, but AS 20 is only reachable by local AS 2 users.

A drawback of this scenario is that if the link from AS 20 to AS 1 fails, the 172.17.144.0/20 route is not reachable from the Internet through AS 2. However, because AS 2 is not the primary ISP for AS 20, this might be acceptable.

Common Community Groupings

- Allows for documentation of information and actions useful for Service Providers and their Customers.
  - Informational Communities
    - Can be used to convey SP specific information (i.e. how and where a route is learned)
  - Action Communities
    - Can be used to control route attributes and how they should be exported to EBGP neighbors (i.e. AS-Path prepend, MED preference)
  - Local Preference Communities
    - Used to influence for best-path selection. (i.e. values for backup routes, and setting various local preference values)
  - Other Communities
    - Implement the ability to blackhole disruptive traffic
    - Congestion

© 2020 Juniper Networks, Inc. All Rights Reserved. juniper NETWORKS | 22

Organizing Communities

Communities can be used to carry and convey information. With the amount of possible information that can be conveyed, many Service Providers will organize their communities into groups. These groups will then be published on their customer service portals for customer reference and use.

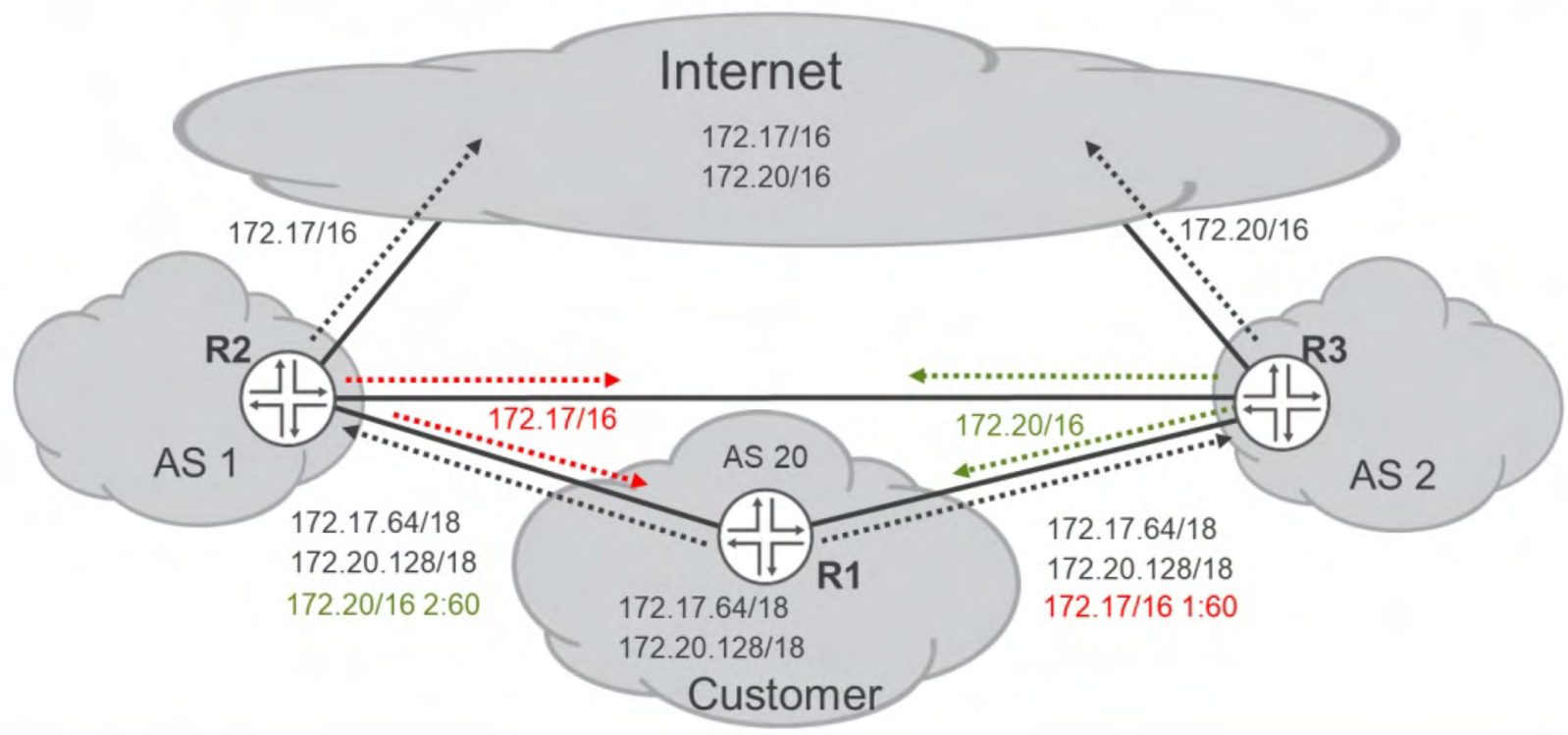
By using different number ranges, the various communities can be uniquely identified through regex comparisons.

Common Categories of Communities that a Service Provider might publish include the following:

- *Informational Communities*: These communities can be used to document information about a route, such as where and how the route was learned. Using a possible format ASN:#####, the number field can have each character represent some classification of the route origin. For example, the first position of the community value could be made to represent the entity that added the route. Then the next position could represent continent, then region, city, etc.
- *Action Communities:* These communities can be used to convey information about how the customer would like their routes to be exported from the Service Provider network. Using a possible format like ASN:####, the number field can convey actions for the Service Provider to apply to the received routes. For example, pre-pending the AS number 1, 2 or 3 times of certain routes or how to manage the MED value.
- *Local Preference Communities:* This is probably the most commonly used category of communities. Used to allow the customer to convey to the Service Provider what local preference to associate with the routes they receive. This is generally more preferred than having a customer use MED to influence routes. This allows the Service Provider to exert control over the route selection process.
- *Other Communities:* Communities can be used for many reasons. Some capabilities don't fit into fixed categories but can be used effectively in a network. A couple of examples include: marking disruptive routes that should be blackholed. These communities might be used with automation to report congestion in the network and when forwarded to a customer, the customer could decide to reroute traffic.

Backup Routes

- Customer agrees to provide backup transit service to AS 1 and AS 2
- AS 1 and AS 2 agree to set local preference based on community



© 2020 Juniper Networks, Inc. All Rights Reserved.

Transit AS for Backup

One of the most common uses of the BGP community attribute addresses one major limitation of the BGP local-preference attribute. Local preference is only distributed within an AS, and no local-preference information is ever sent between two AS networks. Yet, local preference is a valuable way to establish proper exit points for a route within an AS, especially when the AS has multiple links. How can another AS be informed of the local preferences of a route learned from another AS? The most popular way to do this is with BGP communities.

In the example on the slide, the customer AS at the bottom agreed to provide backup transit service to both AS 1 and AS 2 in case their links to each other are lost. The slide shows the address spaces used. The customer AS uses 172.17.64.0/18 and 172.20.128.0/18.

The customer wants to make the 172.20.0.0/16 route sent to AS 1 to have a *lower* local preference than the default of 100, and to make the 172.17.0.0/16 route sent to AS 2 to have a lower local preference than the default of 100 there as well.

The key to making this work is with the BGP communities on routes sent to AS 1 and AS 2. Route 172.20.0.0/16 sent to AS 1 is tagged as community 1:60. This says to AS 1, "AS 1, within your AS, this route should have a local preference of 60." The route 172.17.0.0/16 sent to AS 2 is tagged as community 2:70. This says to AS 2, "AS 2, within your AS, this route should have a local preference of 70." In this example, AS 1 and AS 2 only advertise aggregate routes to each other and the Internet.

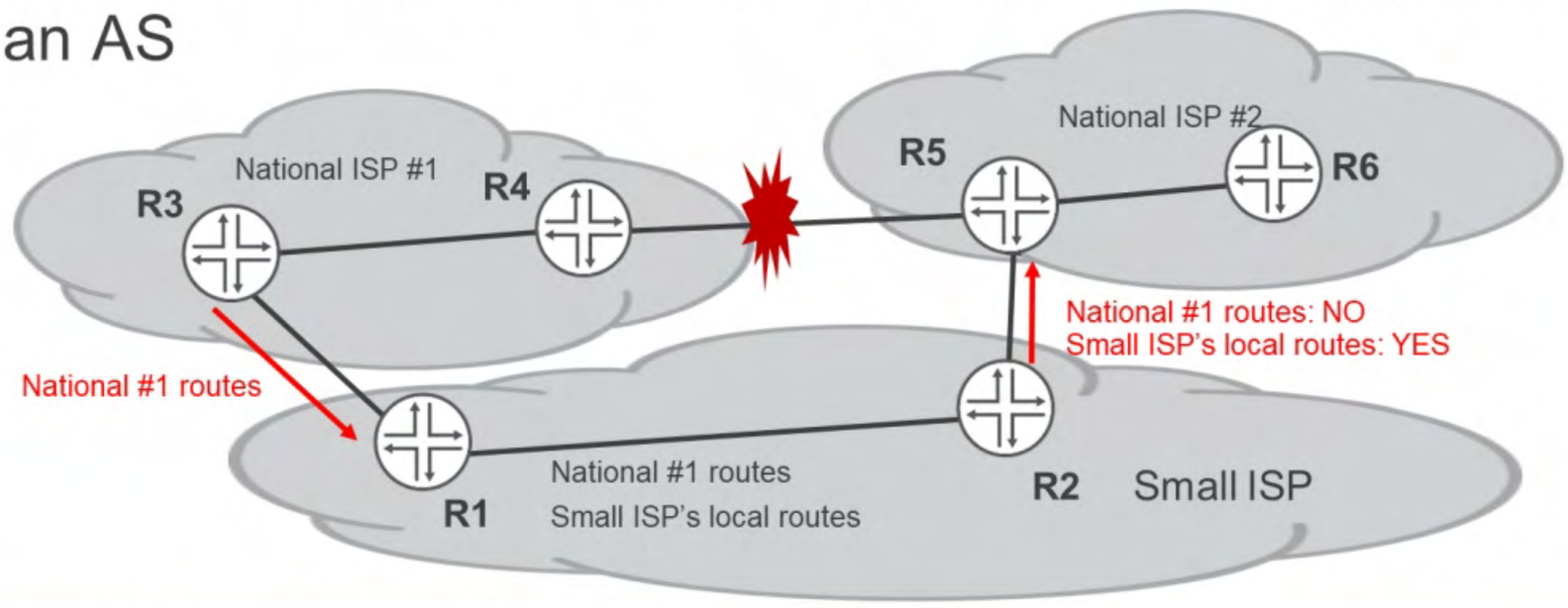
If AS 1 and AS 2 set the local preferences on these routes as requested, the exit points through the customer's AS are only active when there is no *normal* peering route available (local preference = 100).

Enforcing Communities

Of course, setting local preferences in other AS networks with communities requires all the AS administrators to cooperate. Nothing *makes* an AS respect the community attribute value.

Transit and Nontransit

- Can use policies in BGP to make an AS appear as a transit AS or nontransit AS for a route
- Communities make it easier to *hold back* routes that might be used to transit an AS



Small ISP needs to advertise its own routes but should *never* be use as a transit AS!

© 2020 Juniper Networks, Inc. All Rights Reserved.

Policies and Transit AS

The BGP community attribute plays a key role in defining whether an AS is a transit or nontransit network. A transit AS carries traffic that neither originates in that AS nor is destined for hosts within the AS. A nontransit AS only carries traffic that has its own addresses appearing as either the source or the destination. Nontransit AS networks must be careful when advertising BGP routes outside the AS. A nontransit AS can advertise only local routes.

Communities Can Hold Back Routes

You can use routing policies in combination with BGP communities to make an AS appear to be a transit AS or a nontransit AS. Communities make it much easier to *hold back* routes that might be advertised and attract transit traffic.

Generally, a small ISP must advertise its own local routes but never be a transit AS for larger ISPs. Such a situation could easily swamp the operation of a small ISP.

The slide shows a small ISP linked to two larger, national ISPs: National ISP #1 and National ISP #2. As an example of what could happen, consider that National ISP #1 advertises BGP routes to R1 of the small ISP as shown on the left. R1 advertises not only its own local routes to R2 in the small ISP, but also National ISP #1's routes, so that all users can reach these routes.

But R2 should never, ever advertise National ISP #1's routes to National ISP #2! R1's and R2's local routes are okay to send to National ISP #2. However, if R2 ever advertises the National ISP #1 routes to National ISP #2, and the link (or links) between the two national ISPs ever fail, National ISP #2 will think that a good way to reach National ISP #1 is through the small ISP! Hence, the small ISP is now a transit network.

Configuring a BGP Community

- Community ID format:
  - ***as-number*:*community-value***
  - Well-known communities can be created using the following Community-ID as the name:
    - **no-export**
    - **no-advertise**
    - **no-export-subconfed**
- Must be configured in Junos as a named object

```
[edit policy-options]
community <name> members community-id;
```

- Using multiple community-ids in a community is a logical AND

```
[edit policy-options]
community <name> members [community-id community-id];
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Configuring Communities

You configure BGP communities in the Junos OS at the `[edit policy-options]` CLI hierarchy level. You simply give the community a name and a number of members in the form of the community ID. When you define multiple community members, a logical AND is between them. Thus, a given name represents *Community1*, AND *Community2*, AND *Community3*, and so on.

Community ID Format

The community ID has an *as-number:community-value* format, with a colon (:) separating the high-order and low-order octets. You can use the keywords `no-export`, `no-advertise`, and `no-export-subconfed` to specify the well-known community values.

Can Use in Policy

When used in a routing policy, you can use the community name as a match condition (that is, *find these BGP communities*) or as an action. Actions applied to communities include the adding, deleting, or setting of community attribute values.

Community Actions: Add

- Leave existing communities alone and add in the specified value

```
user@router> show route 192.168.0/24
192.168.0.0/24 (2 entries, 1 announced)
    Communities: 64512:567 100:20 50:70 1234:66


[edit policy-options]
policy-statement community-actions {
    term add-a-community
        then community add test-comm;
    }
}
community test-comm members 65001:1234;


user@router> show route 192.168.0/24
192.168.0.0/24 (2 entries, 1 announced)
    Communities: 64512:567 100:20 50:70 1234:66 65001:1234
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Add to the Community String

The slide shows the definition and application of a community in a routing policy. This policy leaves the existing community tags on the route in place but also adds the specified community attribute value to the route.

Route 192.168.0.0/24 currently has four community tags on the route: 64512:567, 100:20, 50:70, and 1234:66.

Because the policy `community-actions` has no from statement, all routes are matched. It is not necessary to check for just the BGP routes because only BGP has a community attribute to change. (Including a `from protocol bgp` statement does not change the action of the routing policy.)

All BGP routes have the community tag `test-comm` value of 65001:1234 *added* to the existing community tags on the route. The action of `then community add test-comm` performs this test.

After you correctly apply this routing policy, the 192.168.0.0/24 route has *five* community tags on the route: 64512:567, 100:20, 50:70, 1234:66, and the added 65001:1234.

Community Actions: Delete

- Remove only the specified values and leave other existing communities alone

```
user@router> show route 192.168.0.0/24
192.168.0.0/24 (2 entries, 1 announced)
    Communities: 64512:567 100:20 50:70 1234:66


[edit policy-options]
policy-statement community-actions {
    term add-a-community
        then community delete test-comm;
    }
}
community test-comm members 64512:567;


192.168.0.0/24 (2 entries, 1 announced)
    Communities: 100:20 50:70 1234:66
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Delete from the Community String

This slide also shows a policy that defines and applies a community in a routing policy. This policy removes only the specified values of the existing community tags and leaves other existing community tags in place.

Route 192.168.0.0/24 currently has the same four community tags on the route: 64512:567, 100:20, 50:70, and 1234:66.

Because the policy `community-actions` has no from statement, all routes are matched.

All BGP routes have the community tag `test-comm` value of 64512:567 *deleted* from the existing community tags on the route. The action of `then community delete test-comm` performs this task.

After you correctly apply this routing policy, the 192.168.0.0/24 route has only *three* community tags on the route: 100:20, 50:70, and 1234:66.

Community Actions: Set

- Remove ALL existing communities and add the specified values

```
user@router> show route 192.168.0/24
192.168.0.0/24 (2 entries, 1 announced)
    Communities: 64512:567 100:20 50:70 1234:66


[edit policy-options]
policy-statement community-actions {
    term add-a-community
        then community set test-comm;
    }
}
community test-comm members 65001:1234;


192.168.0.0/24 (2 entries, 1 announced)
    Communities: 65001:1234
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Set the Community String

The slide also shows a policy that defines and applies a community in a routing policy. This policy removes *all* the values of the existing community tags and adds (that is, sets) the new community tag (or tags) in place.

Route 192.168.0.0/24 currently has the same four community tags on the route: 64512:567, 100:20, 50:70, and 1234:66.

Because the policy `community-actions` has no from statement, all routes are matched.

All BGP routes have the community tag `test-comm` value of 65001:1234 set as the existing community tag on the route. The action of `then community set test-comm` performs this task.

After you correctly apply this routing policy, the 192.168.0.0/24 route has only one community tag on the route: 65001:1234.

Community Example 1

- Create and apply a community named *customers*
  - Set BGP local preference to 200 (instead of 100)

```
[edit policy-options]
policy-statement from-customers {
        from community customers;
        then {
            local-preference 200;
            accept;
        }
}
community customers members [56:2379 23:46944];
```

Export policy

```
[edit protocols bgp]
group my-group {
    export from-customers;
    neighbor 200.200.1.1;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

First Example Using Community

The slide shows a realistic application of the BGP community attribute. The goal is to create a community named *customers*. We define the community as having two members: 56:2379 AND 23:46944.

When we use this community in a routing policy to find (that is, match) routes, the community matches routes that have *both* 56:2379 AND 23:46944 as a community tag value.

Once found, these routes are assigned a local-preference value of 200 (instead of the default 100).

When multiple communities are placed in a single named object the relationship is a logical AND. This means both groups must exist in the route for a match to be made.

Community Example 2

- Create and apply a community named ***Classful-accept***
  - Reject routes more specific than /8 for Class A, /16 for Class B, and /24 for Class C

```
[edit policy-options]
community Classful-accept  members 567:1;
policy-statement drop-specifics {
    term drop-specifics {
        from {
            route-filter 0.0.0.0/1 upto /8 {
                community add Classful-accept;
                next policy;
            }
            route-filter 128.0.0.0/2 upto /16 {
                community add Classful-accept;
                next policy;
            }
            route-filter 192.0.0.0/3 upto /24 {
                community add Classful-accept;
                next policy;
            }
        }
        then reject;
    }
    term reject_all {
         then reject;
    }
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Second Example Using Community

The slide shows another realistic example of a BGP community. The goal is to define and attach a community named *my-accept* to routes that are accepted by the policy. In this example, the policy drops routes that are too specific, based upon the prefix's address class, while tagging all other routes with the community value of 567:1.

The *drop-specifics* routing policy accepts the desired routes using a series of route filters that are based upon address classes (A, B, and C) and the allowed prefix length. The `community add` statement attaches the 567:1 community to any existing communities on routes that match the route filters.

The final `reject` action serves to reject all routes that are not matched by the previous route filters. It bears stressing that this `reject` statement is part of an unnamed term with no match criteria.

Continued on the next page.

Second Example Using Community (contd.)

An equally workable approach that makes use of a single unnamed term is shown here:

```
[edit policy-options policy-statement drop-specifics]
user@router# show
from {
    route-filter 0.0.0.0/1 upto /8 {
        community add my-accept;
        next policy;
    }
    route-filter 128.0.0.0/2 upto /16 {
        community add my-accept;
        next policy;
    }
    route-filter 192.0.0.0/3 upto /24 {
        community add my-accept;
        next policy;
    }
    route-filter 0.0.0.0/1 upto /32;
    route-filter 128.0.0.0/2 upto /32;
    route-filter 192.0.0.0/3 upto /32;
}
then reject;
```

In this approach, a series of class-based route filters are added to match on Class A, B, and C addresses that have prefix lengths upto a mask length of /8 /16, and /24 respectively. Note that the final series of route filters do not have any actions specified in the route filter statement. As a result, traffic that matches these statements is subjected to the unnamed term's `reject` action.

Communities in Routing Policy

- Some routing policies *delete* communities, which might be useful at AS boundaries
- By default, communities are sent to peers
- To stop community advertising, delete the community
- This policy deletes all communities—the asterisk (*) wildcard matches any possible value

```
[edit policy-options]
community wild-match members "*:*";
. . .
policy-statement delete-all-communities {
    term all-gone {
        then community delete wild-match;
    }
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Routing Policies and Deletions

In contrast to other BGP attributes like AS path, BGP allows you to delete some or all the community attribute values on a BGP route. In fact, deleting these values is very useful to do at the boundaries of an AS because there is no guarantee that any other AS understands or respects the values of the communities established in one AS.

Default Is to Send

If you do not delete community attribute values, by default, all BGP communities are sent to all peers inside an AS and outside the AS. Why clutter up the routing updates with useless and potentially harmful information?

To Stop, Use Delete

To stop the community from being advertised beyond the local AS, you must delete the community.

Delete All Communities

The slide shows a routing policy that deletes *all* communities from a BGP route. This example uses the wildcard asterisk character (`*`) to match all communities. The action of `community delete` *`wild-match`* performs this task.

Note that you must apply the wildcard to *both halves* of the community—the AS number as well as the community value. The syntax is therefore `"*:*"` to match all communities.

Regex and Communities

- When applied to a community, a regex `term` is a single digit or character not the entire value.
- Community regex expressions can use wildcard values of asterisk (`*`) or dot (`.`)
  - The asterisk matches the whole AS number portion or community value portion of a community string
  - The dot matches any single digit within the community value
- Examples of simple community regex matches:
  - All communities from particular AS: *`as-number`*`:*`
    - For example, 600:* matches all AS 600 communities
  - All AS networks with the same value: `*:`*`community-value`*
    - For example, `*:20` matches value 20 from all AS areas
  - All community values where the 3rd digit is any number
    - For example, `1111:50.0` matches 5000, 5010, 5020, etc. up to 5090 from AS 1111

© 2020 Juniper Networks, Inc. All Rights Reserved.

Simple Community Regular Expressions

You can use a regular expression (also called, regex), first introduced in an earlier section on AS paths, also with BGP communities to produce a powerful pattern-matching system for finding communities that match a given regex. Regular expressions used with communities implement the full capabilities of a complete regex implementation, unlike the AS-path regex syntax, which used a subset.

Consider two forms of regular expressions used with routing policies concerned with BGP communities. These two forms are *`simple`* and *`complex`* regular expressions. These are informal terms, defined in this course as follows. Simple community regular expressions contain only the asterisk (`*`) or dot (`.`) wildcard characters separately. Complex community regular expressions can use the asterisk and dot in conjunction with each other. Further, the complex regex statements can use additional operator syntax characters.

The asterisk matches any single AS number or community value. The dot matches any single digit within the AS number or community value. Note that the combination of these characters (`.*`) is a complex community regex, which we discuss on a later slide.

Examples of Simple Community Regular Expressions

Some examples of simple community regex matches are:

- `*:1000` = Any possible AS number with a value of 1000.
- `65001:*` = AS 65001 with any possible community value.
- `65001:100.` = AS 65001 with community values of 1000, 1001, 1002, 1003, 1004, 1005, 1006, 1007, 1008, or 1009.
- `11.1:1000` = AS 1101, 1111, 1121,1131, 1141, 1151, 1161, 1171, 1181, or 1191 with a community value of 1000.

Community Matching (1 of 2)

```
user@router> show route community *:20 terse

inet.0: 123 destinations, 123 routes (123 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A Destination        P Prf Metric 1   Metric 2    Next hop        AS path
* 192.168.128.0/24   S   5          5             Reject

user@router> show route community *:20 detail

inet.0: 123 destinations, 123 routes (123 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.128.0/24 (1 entry, 1 announced)
        *Static Preference:   5
 Next hop type: Reject, Next hop index: 0
                Address: 0x9f2ec44
                Next-hop reference count: 2
                State: <Active Int Ext>
                Local AS: 65501
                Age: 1:17
                Validation State: unverified
                Task: RT
                Announcement bits (1): 0-KRT
                AS path: I
                Communities: 1:20
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Matching the Community, Part 1

You can use standard CLI commands with simple regular expressions to find the routes (if any) associated with any community. The slide shows a few examples.

To find all routes having a community value of 20, regardless of AS number, use the command **`show route community *:20 terse`**. This command shows the routes but not the complete community attribute values associated with the routes. To see the communities and more detailed information, use **`show route community *:20 detail`**.

Community Matching (2 of 2)

```
user@router> show configuration policy-options

policy-options {
    community community-1 members 1:20;
}


user@router> show route community-name community-1 detail

inet.0: 123 destinations, 123 routes (123 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.128.0/24 (1 entry, 1 announced)
        *Static Preference:   5
                Next hop type: Reject, Next hop index: 0
                Address: 0x9f2ec44
                Next-hop reference count: 2
                State: <Active Int Ext>
                Local AS: 65501
                Age: 1:17
                Validation State: unverified
                Task: RT
                Announcement bits (1): 0-KRT
                AS path: I
                Communities: 1:20
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Matching the Community, Part 2

In addition to the standard CLI commands seen previously, you can also locate a route using the name of a community.

The example on the slide shows a community called *community-1* defined within `[edit policy-options]`. This community represents the value of 1:20. You can use this community name in the **`show route community-name community-1 detail`** command to view all routes having this community assigned.

More Complex Regex

- Can use more complex regular expressions with communities
  - Community regex is *character based* (not like AS path)
  - Format is still `term operator`
  - Regex anchors (^) and ($) are not required, but can be helpful
  - Used in both **show route** and within a policy as a match condition
    - **show route community *regex***
    - **community *match-this* members *regex***
- The combination of wildcards (.*) results in matching one or more digits in either the AS number portion or community value portion of the community string.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Complex Community Regular Expressions

You can use more complex regular expressions (regex) with communities. A community regular expression is character based, unlike the regular expressions used with AS paths, which match entire AS numbers.

The format for the community regular expression is still `term operator` as in AS-path regular expressions, but the application of the term and operator is different.

When formulated for use with communities, the regular expression anchors of start (^) and end ($) are not required, but these anchors can be helpful to organize and clearly represent the regular expression.

You can use complex regular expressions in both the **show route** CLI commands and within a policy as a match condition.

Community Regular Expressions

List of Community Regular Expressions Operators	
Operator	Description
{*m*,*n*}	Matches at least *m* and at most *n* repetitions of the community term
{*m*}	Matches exactly *m* repetitions of the community term
{*m*,}	Matches *m* or more repetitions of the community term
*	Matches zero or more repetitions of the community term
+	Matches one or more repetitions of the community term
?	Matches zero or one repetition of the community term
|	Logical OR, matches one of the community terms on either side of the pipe
^	Regex start anchor (Optional) - Matches communities that have matching starting terms
$	Regex end anchor (Optional) - Matches communities that have matching ending terms
[]	Matches a range of an array of characters or digits
(...)	Matches a group of community terms enclosed in the parentheses

© 2020 Juniper Networks, Inc. All Rights Reserved.

Community Regular Expression Operators

The table on the slide shows a list of the possible regular expression operators you can use with BGP communities. Some operators are shorthand for their longer equivalents. For example, the plus (+) is the same as `{1,}`. Both match one or more repetition of the *`term`* preceding the *`operator`*.

The square brackets ([ ]) match a range (`[2-8]`) or array (`[256]`) of numbers. Thus, the first regular expression in the previous sentence matches 2 through 8, and the second one matches 2 or 5 or 6.

Of special note is the use of the parentheses. Typically, you use the parentheses ( ) operator to group multiple terms in conjunction with an operator. The parentheses operator also has another special use. When used with no spaces in between, parentheses represent a *null* value.

Complex Regex Examples

Community pattern to match:	Regex:	Matches:
AS is 56 or 78	"((56)|(78)):*"	56:1000, 78:65000
AS is 56 and value starts with a 2	"56:(2.*)"	56:234, 56:2, 56:222
AS can be anything and value ends with either a 5, 7, or 9	"*:(.*[579])"	1234:5, 78:2357, 34:65005
AS is 56 or 78, value starts with a 2 and ends with any value between 2 and 8	"((56)|(78)):(2.*[2-8])"	56:22, 56:21197, 78:2678

© 2020 Juniper Networks, Inc. All Rights Reserved.

Examples of Complex Community Regular Expressions

The slide shows some examples of quite complex regular expressions you can use to match communities in routing policies.

The first column shows the BGP community string that the routing policy is trying to match. The second column shows the community regular expression used to match that pattern. The last column shows examples of values of the BGP community attribute that the regular expression matches. In some cases, the list is not exhaustive, so more possible communities match the pattern.

Note the presence of the colon (:) to separate the AS number of community value sections of the community.

Test Your Knowledge

- Does the following community regex meet the criteria?
  1. The AS can be either 105, or 207, or 309 ✓
  2. The value must be a 4- or 5-digit number starting with a 1 ✓
  3. Or, the value must start with a 2, 5, or 6 ✓
  4. Or, the value must end with a 3, 4, 7, or 9 ✓

```
"^((105)|(207)|(309)):((1.{3,4})|([256].*)|(.*[3479]))$"
```

1 2 3 4

© 2020 Juniper Networks, Inc. All Rights Reserved.

Complicated Community Regular Expression Example

The answer to the question on the slide is *yes*, the community regular expression on the slide meets the criteria outlined. This complex regular expression uses multiple term-operator pairs in its definition. To understand better how the expression operates, let's reconstruct it from the ground up.

First, we have a basic expression of `^():()$`. This part of the expression sets the foundation for the complete expression. Loosely speaking, we search for an exact match of an AS number followed by a colon (:), followed by an exact match of a community value.

Next, we assign the AS value. This AS value is actually a regular expression in itself that states the AS is either 105, 207, or 309. The pipe (|) operator separates the three AS numbers into logical OR groupings. The extra parentheses are used to set aside each AS number explicitly from the pipe operator. The regular expression now appears as `^((105)|(207)|(309)):()$`.

Now we can define the community values. As with the AS numbers, the values can be one of three separate entities. Again, we use the pipe operator to separate the values and the parenthesis to define the possible values explicitly. The regular expression is now `^((105)|(207)|(309)):(()|()|())$`. Each of the possible community values in this example is an individual expression itself. We'll examine each one in turn.

The first possible value is a 4- or 5-digit number, where the first character is a 1. Because the community expressions are a character-based match, the expression starts simply with a 1. The following characters in the value can be any number, so the wildcard of dot `(.)` is used to represent that. Because the total value should be 4 or 5 digits, the wildcard can be operated upon with a `{3,4}`, which means at least three instances of any number can appear, but no more than four instances. Combined with the character of 1 to start the value, the wildcard-operator pair makes the value a 4- or 5-digit number. The regular expression now appears as: `^((105)|(207)|(309)):((1.{3,4})|()|())$`.

Continued on the next page.

Complicated Community Regular Expression Example (contd.)

The second possible value can be any length, but it must start with either a 2, 5, or 6. Again, the community expressions are character based, so this value should also start with a character. These are actually three possible characters in this single position, so the brackets are used (`[256]`) to signify a range of possible values. Following that, there can be more characters in the value, or there could be no characters in the value. To represent this possibility, we once again use the wildcard dot (`.`). In this case, the wildcard is operated on by the asterisk (`*`), which results in a `.*` notation. This represents any possible value present zero or more times. Combined with the `[256]`, this wildcard-operator pair gives any possible value of any length as long as it starts with a 2, 5, or a 6. The regular expression now appears as:
`^((105)|(207)|(309)):((1.{3,4})|([256].*)|())$.`

The final possible value can again be any length. This time, it must end with either a 3, 4, 7, or 9. The logic for this value is exactly the opposite of the logic for the second value, so we can use the same operators. In this case, the value starts with the `.*`, which again represents any possible value present zero or more times. This is followed by the bracket notation of `[3479]` to represent a single character in that single position. When combined, the result is any possible value of any possible length, as long as it ends with a 3, 4, 7, or a 9. The regular expression now appears as:
`^((105)|(207)|(309)):((1.{3,4})|([256].*)|(.*[3479]))$.`

Summary

- In this content, we:
  - Described the BGP attributes Origin, MED and communities in detail and explain the operation of those attributes
  - Manipulated BGP attributes using routing policy

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- The BGP attributes Origin, MED, and communities in detail and explained the operation of those attributes; and
- How to manipulate those BGP attributes using routing policy.

Review Questions

1. What Origin code does the Junos OS inject on routes that are redistributed into BGP?
2. Why is the default not to compare the MEDs that come from two different ASs?
3. What are the well-known communities?
4. What is the difference between the add and set community actions?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

3.

4.

Lab: BGP Attributes – Part 2

- Influence routing using the Origin attribute
- Influence routing using the MED attribute
- Influence routing using the community attribute

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: BGP Attributes: Local Preference and Communities

The slide provides the objectives for this lab.

Answers to Review Questions

1.

The Junos OS codes all routes that are redistributed into BGP as I. I is for Internal and the underlying value is 0.

2.

Different ASs can set the MED values differently. A good MED value from one AS may be a bad value from another AS.

3.

The well-known communities are: No-export; No-advertise; and No-export-subconfed.

4.

The add community action adds the specified community string to the existing community attribute. The set community action deletes any existing communities and adds the specified community string.

Education Services

Advanced Junos Service Provider Routing

Chapter 13: Route Reflection and Confederations

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - Describe the operation of BGP route reflection
  - Configure a route reflector
  - Describe the virtual route reflector
  - Describe the operation of optimal route reflection
  - Describe the operation of a BGP confederation
  - Configure confederations
  - Describe peering relationships in a confederation

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- The operation of BGP route reflection;
- How to configure a route reflector;
- Virtual route reflection;
- The operation of optimal route reflection;
- The operation of a BGP confederation;
- How to configure confederations; and
- Peering relationships in a confederation.

Agenda: Route Reflection and Confederations

→Route Reflection Operation
- Configuring Route Reflection
- Virtual Route Reflection
- Optimal Route Reflection
- Route Reflection Case Study
- BGP Confederations

© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Reflection Operation

The slide lists the topics we will discuss. We discuss the highlighted topic first.

Scaling BGP

- IBGP full-mesh peer requirement has an n^2 problem
  - Addition of a new router requires new peering with all current IBGP speakers
  - Current IBGP speakers must update their configurations
- Two primary scaling mechanisms:
  - Route reflection (RFC 4456)
  - Confederations (RFC 3065)

© 2020 Juniper Networks, Inc. All Rights Reserved.

IBGP Full Mesh

Unlike link-state routing protocols, internal BGP (IBGP) does not flood routing updates. Instead, IBGP uses an explicit peering model that normally results in the exchange of routing information to peers that are connected in a full mesh. The need for a full-mesh IBGP topology stems from the fact that BGP uses the autonomous system (AS) path attribute to provide loop detection, but IBGP speakers do not add the local AS number in the updates they send to other IBGP speakers. Lacking AS number-based loop detection, IBGP speakers are normally precluded from readvertising routes to other IBGP speakers when the route in question was learned from an IBGP speaker. This default IBGP behavior leads to the need for a full mesh of IBGP peerings.

Requiring that all IBGP peers within an AS be fully meshed has inherent scalability problems. For example, every time a new router is added to the AS, each existing IBGP router must have its configuration updated to include a peering statement for the router that has been added. This process can become quite an issue when there are 1000, 200, or even 100 routers in an AS. In fact, with only 100 routers in a full IBGP mesh, each router is required to maintain 99 IBGP peering sessions, with the network having to support a total of 4,950 IBGP sessions! Surely there has to be a better way.

Two Ways to Improve Scalability

The two primary ways to eliminate the need for a full BGP mesh are route reflection, as defined in RFC 4456, and BGP confederations, as defined in RFC 3065. This chapter explores the configuration and operation of both mechanisms.

Route Reflection Concepts

- Allows an IBGP speaker to readvertise an IBGP-learned route to another IBGP speaker
- Route reflector only re-advertises the active route to clients
- Route reflector does not, by default, change existing IBGP attributes
- Two new BGP attributes to prevent loops:
  - Cluster list
    - Contains one or more cluster ID values
  - Originator ID

© 2020 Juniper Networks, Inc. All Rights Reserved.

IBGP Peers Can Readvertise Routes

BGP route reflection relaxes the restriction that an IBGP peer should not readvertise IBGP-learned routes to other IBGP speakers. The routers allowed to override this default behavior are known as route reflectors (RR).

Route Reflector Sends the Active Route

RRs only readvertise the active routes to their *clients*. You configure an RR by using the **`cluster`** statement within an IBGP peer-group configuration. BGP considers each of the peers configured within that peer group to be clients of the RR. The RR clients require no configuration changes; they do not have any knowledge of the presence of the RR—they simply see the RR as an IBGP peer.

IBGP Attributes Not Changed

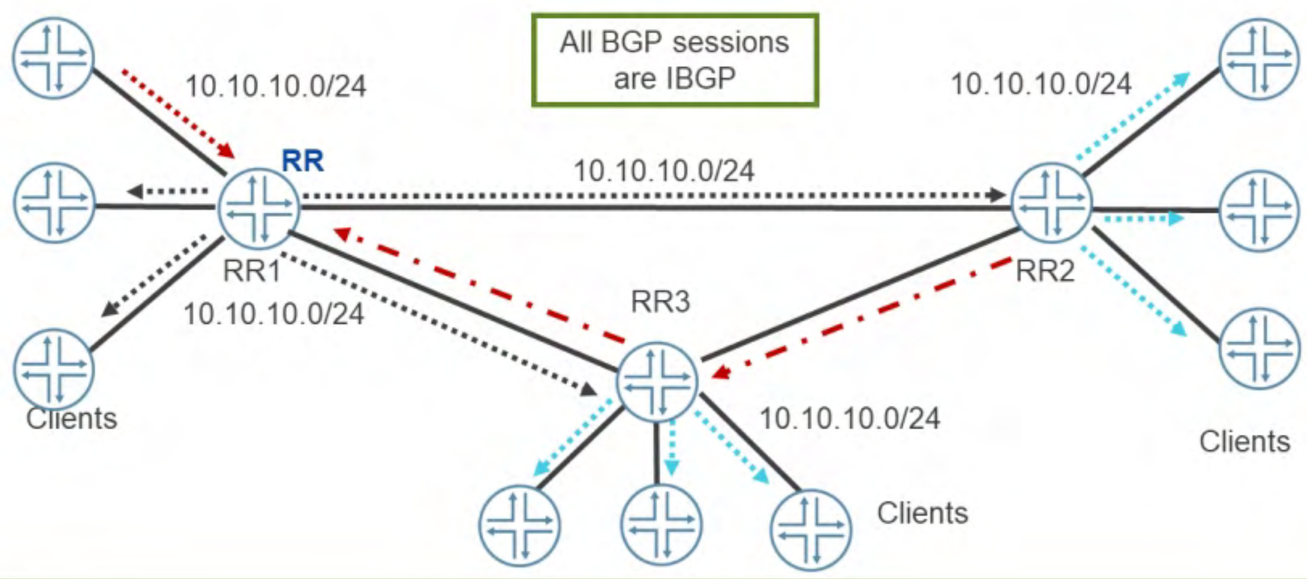
One of the primary drivers behind requiring the IBGP full mesh in the first place was loop prevention, because the AS path attribute is not modified within an AS. Route reflection does not change that behavior. In fact, *none* of the existing BGP attributes change, by default, when BGP uses route reflection in an AS. However, loop prevention is still a critical part of BGP, so new BGP attributes were introduced to facilitate loop detection in a route reflection network.

New BGP Attributes

Two new BGP attributes are defined to support route reflection; these attributes are the *cluster list* and the *originator ID*. An RR creates or modifies these attributes when it readvertises the routes to both clients and non-clients. The route reflector's cluster ID is added to all routes that the RR touches, meaning that both clients and non-clients receive the cluster list attribute. This attribute contains a sequence of all cluster IDs that represent all RRs that have handled the route update.

New Cluster Attributes Prevent Loops

- Steps:
  1. Client sends routes to RR
  2. RR sends routes to all clients in the cluster and all RRs
  3. Those RRs send the routes to all their peers forming a loop



© 2020 Juniper Networks, Inc. All Rights Reserved.

New Cluster Attributes Prevent Loops

Without the new cluster attributes, a loop can be created:

1. Client sends routes to RR1;
2. RR1 sends routes to all clients and to RR2 and RR3;
3. Route reflection will also send routes to other IBGP peers, causing RR2 to send the routes to RR3; and
4. Because RR3 has no way of knowing the routes received from RR2 came from RR1, RR3 sends them to RR1, forming a loop.

Route Reflection Attributes

- Cluster list:
  - Operates like an AS path, used by RR for loop prevention
  - Also used in the route selection algorithm
  - Contains a sequence of cluster IDs
    - Cluster ID represents each RR cluster in the network
    - RR drops routes that have already transited the cluster
    - Added to the cluster list when a RR touches a route
- Originator ID:
  - Identifies the first router to inject a route in an RR network

© 2020 Juniper Networks, Inc. All Rights Reserved.

Cluster List

The cluster list attribute is analogous to the AS path attribute and is used to prevent loops. If an RR receives a route with its own cluster ID in the cluster list, it drops the route. In addition, each router in the network can use this attribute in the BGP path selection algorithm prior to using the peer IP address attribute. BGP chooses the route with the shortest cluster list length. This process follows the same theory as the AS path attribute.

The cluster ID is very similar to an AS number and should be unique within an individual AS. The cluster ID is added to the cluster list attribute when a route is sent to clients and non-clients.

Originator ID

The originator ID attribute provides the router ID of the first router to advertise the route in the AS. In the rare case where the cluster list is the same length, then the originator ID can be used for route selection.

Agenda: Route Reflection and Confederations

- Route Reflection Operation
- →Configuring Route Reflection
- Virtual Route Reflection
- Optimal Route Reflection
- Route Reflection Case Study
- BGP Confederations

© 2020 Juniper Networks, Inc. All Rights Reserved.

Configuration of Route Reflection

The slide highlights the topic we discuss next.

Route Reflection Configuration

- Route reflector clients are configured in a separate peer group
- Each peer group uses the **`cluster`** keyword
  - Cluster ID uses unique 32-bit number
    - Often the router ID of the RR is used

```
[edit protocols bgp]
group int-peers {
   type internal;
   local-address 172.16.1.1;
   cluster 172.16.1.1;
   neighbor 172.16.2.2;
   neighbor 172.16.3.3;
   neighbor 172.16.4.4;
}
```

- Clients only peer to their route reflectors

```
[edit protocols bgp]
group int-peers {
   type internal;
   local-address 172.16.2.2;
   neighbor 172.16.1.1;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Clients in a Peer Group

Within Junos configuration syntax, all clients of an individual RR are placed within a single peer group. This placement allows the route reflector to easily determine which peers are clients of a particular cluster. No configuration changes are needed in the client's configuration.

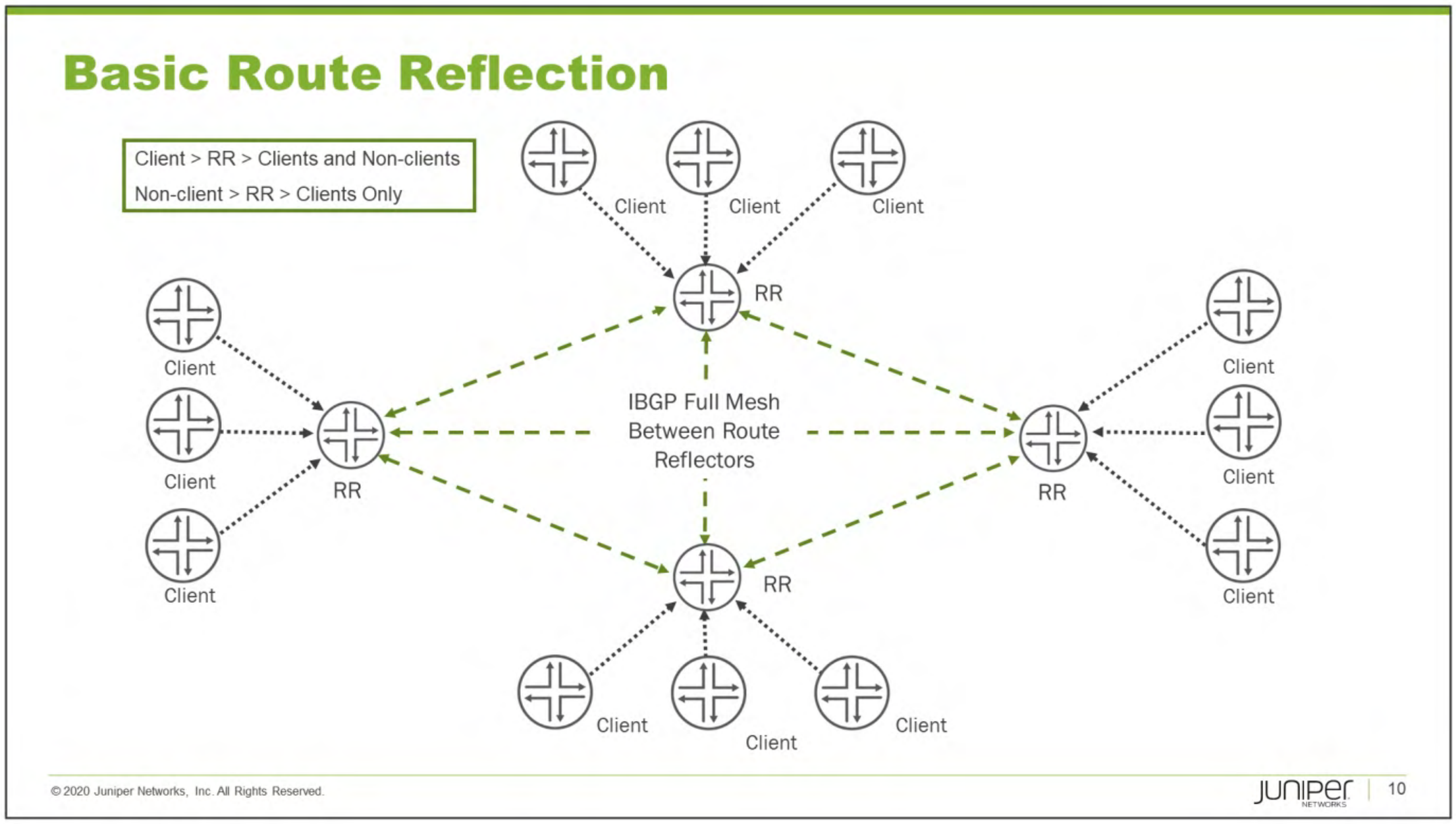
Route Reflector Uses the `cluster` Command

Once the clients are placed into their respective peer groups, use the **`cluster`** command to activate the route reflection process of the route reflector. The **`cluster`** command is used to assign each cluster its cluster ID. This cluster ID is a 32-bit value that uniquely describes the cluster within the BGP AS. If only a single route reflector exists in the cluster, the router ID of the route reflector is often used as the cluster ID.

One of the choices when configuring the RR is whether or not to use the same or different cluster IDs on a redundant pair of route reflectors. Assigning the same cluster ID will reduce the total number of routes stored as routes from one cluster ID will not be stored on the other router with the same Cluster ID. Using different cluster IDs results in all the routes that are exchanged between route reflectors being saved. While this approach increases the number of routes being stored, it can offer increased redundancy in certain situations. There are advantages and issues with either choice. If the ability to manage the extra routes does not create issues, the choice to maintain the extra information should be considered.

Clients Peer to Route Reflectors

The clients in the cluster must peer to the route reflector itself. The clients have no knowledge of the cluster and see the reflector as a *regular* IBGP peer.



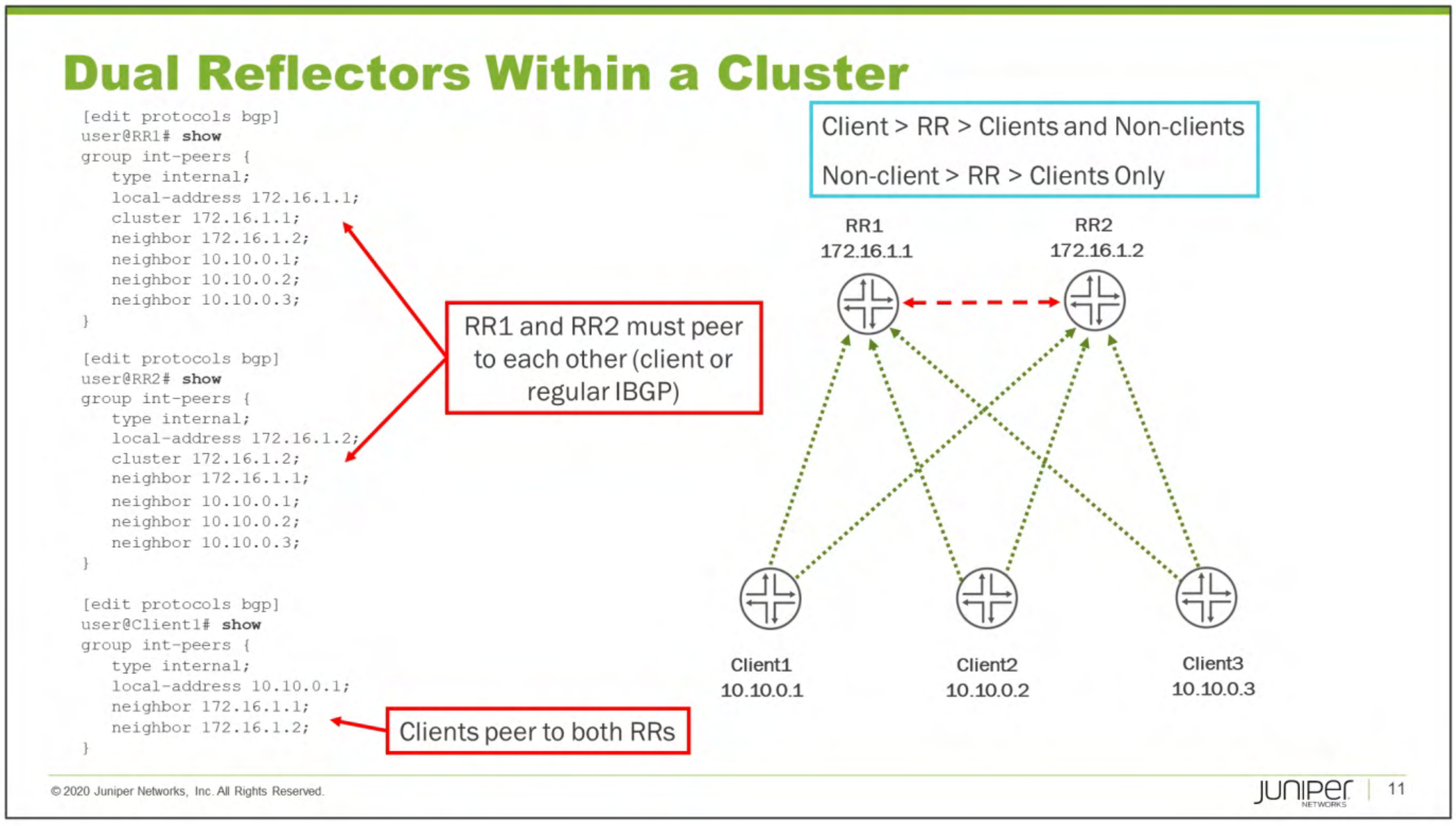
Basic Route Reflection

The slide shows an AS network using a typical route reflection topology.

BGP-speaking routers along the edge of the network all have a single peer configured to peer with the route reflector for the local cluster.

The route reflectors are, in turn, fully meshed using standard IBGP peering procedures. The result is that all routes received by any BGP router will eventually be seen by all other BGP routers in the AS.

It is a common best practice to have the logical route reflection topology follow the physical topology of the network.

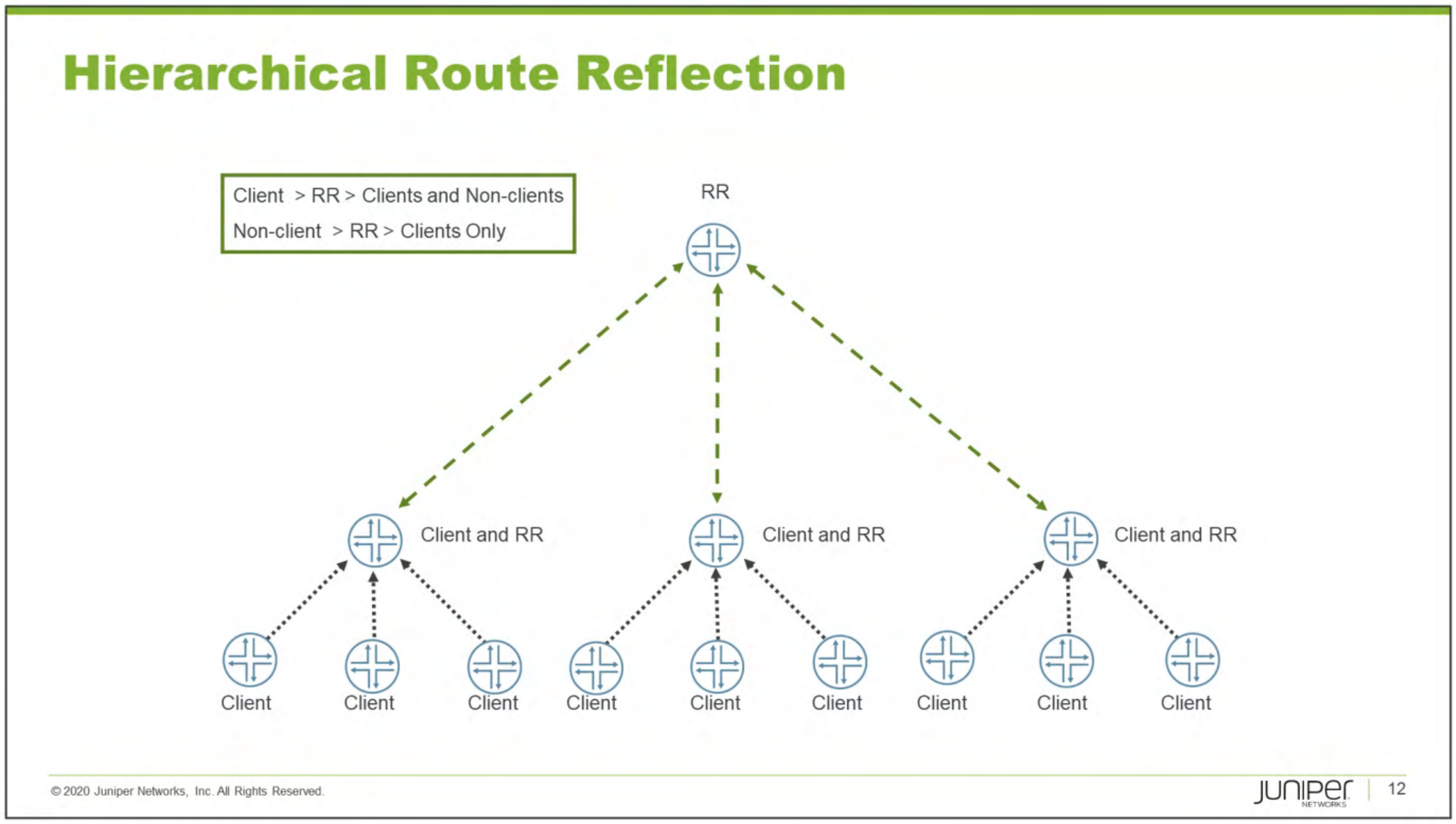


Dual Route Reflectors in a Cluster

The slide shows a cluster containing two route reflectors. This type of cluster design is popular because it avoids a single point of network failure. When a cluster has only a single route reflector, the clients might become segmented from the network in the event of a failure of that RR.

Each of the client routers configures two IBGP peers and forwards EBGP-learned routes to both route reflectors. The route reflectors themselves can peer either within the cluster as clients of each other or outside of the cluster as normal IBGP peers.

Each of the route reflectors also establishes IBGP peering sessions with the other RRs in the entire AS.



Hierarchical Route Reflection

The slide shows an AS network using a more complex, hierarchical, route reflection topology.

Hierarchical route reflection occurs when the route reflectors for some clusters are themselves clients in another route reflection cluster. Very often AS networks evolve to this type of setup when the reflector full mesh shown on a previous slide becomes too large to manage. In this case, the internal route reflector full mesh might evolve into a route reflection cluster.

RR Client Full Mesh

- Clients can also peer with other members of the RR cluster
  - To stop unnecessary advertisements, configure the RR with the **no-client-reflect** command

```
[edit protocols bgp]
user@rr-client-1# show
group int-peers {
   type internal;
   local-address 172.16.2.2;
   neighbor 172.16.1.1;
   neighbor 172.16.3.3;
}

[edit protocols bgp]
user@rr-client-2# show
group int-peers {
   type internal;
   local-address 172.16.3.3;
   neighbor 172.16.1.1;
   neighbor 172.16.2.2;
}
```

```
[edit protocols bgp]
user@route-reflector# show
group int-peers {
   type internal;
   local-address 172.16.1.1;
   cluster 172.16.1.1;
   no-client-reflect;
   neighbor 172.16.2.2;
   neighbor 172.16.3.3;
}
```

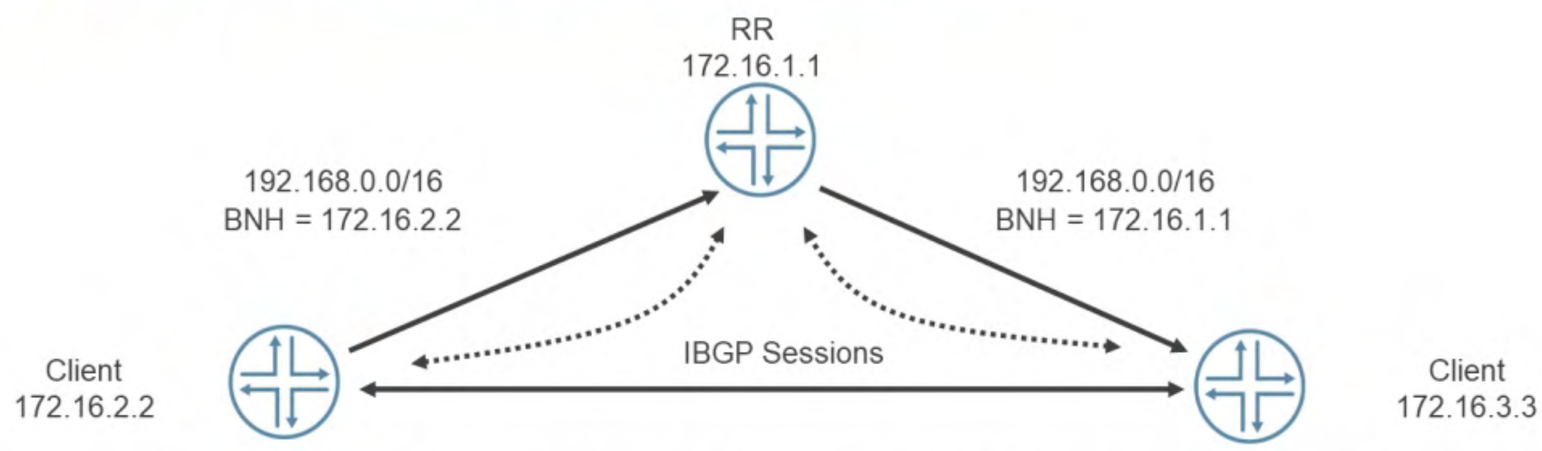
© 2020 Juniper Networks, Inc. All Rights Reserved.

Clients Can Peer with Other Clients

Clients within a cluster can peer with other clients in a full-mesh environment. This ability does not change the operation or need for the route reflector. The reflector still sends routes from the clients to the remainder of the IBGP network and forwards routes from the IBGP network into the cluster. What the client full mesh does provide is the ability for clients to use other clients' routes *natively* when logical BGP connectivity exists between the clients.

In this situation, each of the clients receives two versions of the route. One version is from the other client, and one version is from the route reflector. Because the *extra* copy of the route from the reflector is not needed, you can disable the internal cluster readvertisements using the **no-client-reflect** command. Once configured, the route reflector only forwards to the clients routes that arrive from outside of the cluster.

Modifying Attributes on the RR



- Route reflector can modify any BGP attribute using a routing policy
- Presence of RRs should not affect forwarding paths
  - Use of **`next-hop self`** can result in inefficient forwarding paths
  - In this example, the RR incorrectly overwrites the BGP next hop for the 192.168.0.0/16 route
    - Packets are now forwarded through the reflector instead of directly between the clients

© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Reflector Can Modify Attributes

The default operation of a route reflector is to not modify any BGP attributes. However, Junos does allow an applied routing policy to do just that. The reason for this action is the support of customer networks. For reasons outside the scope of this course, some network administrators engineer traffic flows by altering attribute values when the route reflector readvertises routes from a non-client into the cluster. This alteration is supported through the use of routing policies applied to the cluster's peer group.

Forwarding Paths Should Be Unaffected

Although a client learns of a route from the cluster's route reflector, the route reflector itself does not have to be in the forwarding path for packets sent from clients toward the route destination. In fact, oftentimes it is not.

In the example on the slide, the 172.16.2.2 cluster client advertises the 192.168.0.0/16 route to the cluster's RR with the BGP next hop set to its router ID. Because of a misconfigured next-hop-self policy on the RR, the BGP next hop is overwritten with the router ID of the RR—172.16.1.1. When client 172.16.3.3 receives and installs this route, suboptimal forwarding occurs as packets are sent through the RR instead of directly to 172.16.2.2. This situation might occur when the route reflector also has EBGP peering sessions established. Most network designs avoid this problem by placing their route reflectors within the core of their networks.

The solution to this problem is a selective next-hop-self policy on the RR that modifies the BGP next hop for *only* EBGP-learned routes. This type of policy normally makes use of the `from neighbor` or `from interface` match conditions, as shown here. In this example, the RR has an EBGP peering session with the 172.16.0.1 address:

```
[edit policy-options policy-statement selective-nhs]
user@router# show
term only-EBGP-routes {
    from {
        protocol bgp;
        neighbor 172.16.0.1;
    }
    then {
        next-hop self;
    }
}
```

RR Designs

- Inline
  - RRs are in the path of forwarding traffic
  - Commonly deployed
  - Hierarchical RR normally used
  - RRs perform route reflection and forwarding
- Off-path
  - Supports a centralized route reflection design
  - Becoming more prevalent as virtual RRs are being used
  - Does not require BGP forwarding information in the FIB

© 2020 Juniper Networks, Inc. All Rights Reserved.

Inline RR Designs

Up to this point, the discussion has centered around configuring route reflectors inline with the forwarding of traffic. This has been the most commonly deployed RR design. This design tends to keep route reflectors close to their clients which ensures optimal forwarding of traffic. This design also requires the RRs to perform both route reflection and forwarding which increases the resource requirements of the RR devices. As BGP has been asked to provide more and more network functionality such as L2VPNs, L3VPNs, EVPN, FlowSpec, Multicast VPNs, etc, the need to buy larger network devices to support the state generated by these features has created a desire to move to a more centralized RR design and implement the RR off-path.

Off-path RR Designs

The off-path RR design takes the RR out of the forwarding path of the traffic, allowing it to perform only RR functions. This relieves the core devices from having to be larger and more powerful due to the removal of the forwarding function. This also removes the requirement for the BGP information to be copied and maintained in the forwarding information base (FIB).

The move to the off-path design is aided by the release of the virtual route reflector (VRR) which is discussed next.

Agenda: Route Reflection and Confederations

- Route Reflection Operation
- Configuring of Route Reflection
- →Virtual Route Reflector
- Optimal Route Reflection
- Route Reflection Case Study
- BGP Confederations

© 2020 Juniper Networks, Inc. All Rights Reserved.

Virtual Route Reflector

The slide highlights the topic we discuss next.

Virtual Route Reflector

- A software only implementation of Juniper's RR functionality
- Improved scalability (depending on the server core hardware use)
- Scalability of the BGP network with lower cost using vRR at multiple locations in the network
- Fast and more flexible deployment using Intel servers rather than router hardware
- Space savings through elimination of router hardware
  - Can be installed on a general purpose virtual machine
  - KVM, VMware, and OpenStack supported
  - 64-bit Intel-based blade server or appliance

© 2020 Juniper Networks, Inc. All Rights Reserved.

Virtual Route Reflector

The virtual route reflector (VRR) is a software only implementation of Juniper's RR functionality. Since the core function of the vRR is to receive, process, and forward BGP NLRIs to both clients and non-clients, it is reasonable to separate that functionality from the core routing services. In this way the power requirements can be determined by the server core hardware that is in use. The footprint of the server as well as the number of CPUs and amount of memory can be adjusted much easier on commercially-based virtual servers as opposed to scaling a vendor's fixed form factor networking product. Basing the hardware on the needs of the RR provides a way to distribute them in the network where they are best positioned to provide the needed service.

Juniper's VRR can be installed on any general purpose 64bit x86 virtual machine (VM) that can support a KVM, VMware, or OpenStack environment.

Agenda: Route Reflection and Confederations

- Route Reflection Operation
- Configuring Route Reflection
- Virtual Route Reflector
- →Optimal Route Reflection
- Route Reflection Case Study
- BGP Confederations

© 2020 Juniper Networks, Inc. All Rights Reserved.

Optimal Route Reflection

The slide highlights the topic we discuss next.

What is BGP ORR

- IETF draft
  - Draft-ietf-idr-bgp-optimal-route-reflection, (2011)
  - As of July 2019 in its 19th revision
- Two types of ORR
  1. Optimal BGP path selection based on client perspective
     - Software solution only, no BGP protocol changes
  2. Optimal BGP path selection based on policy (traffic engineering data)
     - Not fully defined
- Juniper's implementation of ORR
  - Needs full knowledge of network topology
  - Requires link state protocol: OSPF or IS-IS

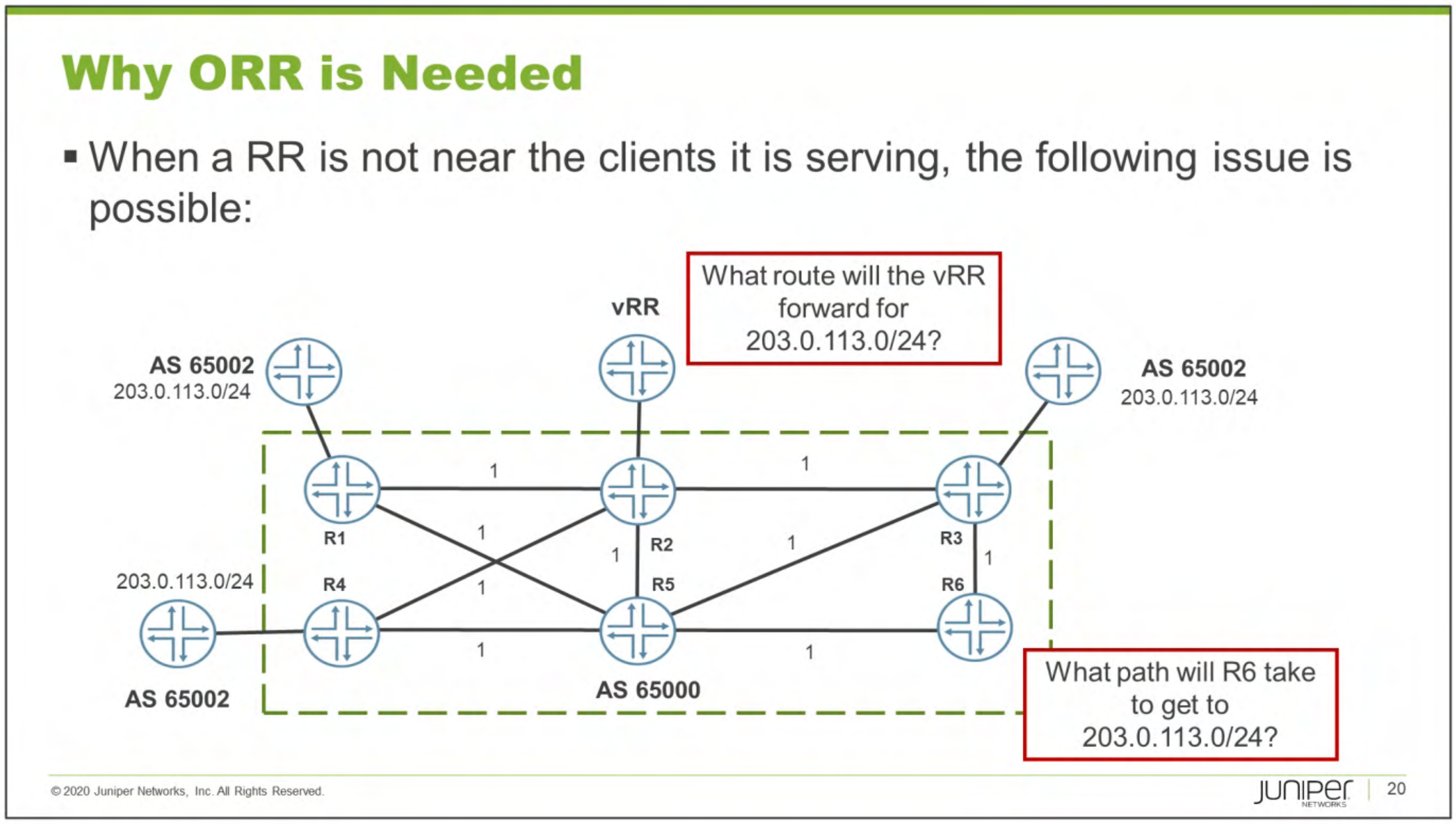
© 2020 Juniper Networks, Inc. All Rights Reserved.

What is BGP ORR

Optimal Route Reflection is an IETF draft that addresses the issue caused by a RR receiving multiple copies of the same route. The RR chooses the best route to forward based on its view of the network. When the client receives the route, it uses the next hop of the route received from the RR which may not reflect the optimal path for the client. The draft was introduced in 2011 and as of July 2019 is in its 19th revision.

Two types of ORR path selection solutions have been put forth. The first solution uses a link state database stored on the RR and running the SPF algorithm based on the client perspective and using those metrics to determine the best path. This has been implemented by most vendors that offer BGP ORR. The second solution involves using path selection based on policy and using traffic engineering data. This method has been slower to be fully developed and is not as widely supported.

Juniper supports the first type of ORR. This requires that the VRR participates in the link-state protocol (OSPF or IS-IS) and understands the full topology of the network.



Why ORR is Needed

This slide depicts a network receiving the 203.0.113.0/24 network from several connections to AS 65002. All PE devices peer with the vRR. The vRR receives the routes and using its BGP route selection process, determines the best route. This is the route it will forward. BGP best route selection uses the following criteria in this order: local preference, as-path, origin, MED, EBGP over IBGP, IGP metric, Cluster ID list, and then lowest router ID. In this case, no criteria defines a specific route as best, prior to the Router ID. Using this option, R1 will be chosen as the preferred router to reach this prefix. This is good for R1 and R4, but not for R6.

Possible Suboptimal RR Issue Solutions

- Hierarchical RRs
  - RRs in close proximity to clients
  - Limits where you can deploy RR
- Advertise multiple paths through the RR
  - Add-path reduces the benefit of route reflection as additional route information is reintroduced
  - More BGP update churns
- Internet in a VRF with unique RDs per peering router
- Use tunnels to make farther locations look closer
- Optimal Route Reflection

© 2020 Juniper Networks, Inc. All Rights Reserved.

Possible Suboptimal RR Issue Solution

Hierarchical RRs

The problem of suboptimal route reflection presents itself when a RR is not close to its clients. Most hierarchical designs keep the RR close to the client and, as such, does not normally run into the suboptimal route issue.

Add-path

Add-path will send multiple routes for the same prefix to the RR clients. This will allow the client to make their own decision about the best path to a given destination. While fixing the suboptimal routing problem, it adds additional routes and processing to the local routers that route reflection was meant to reduce. Not only that but this solution requires special configuration of both the cluster device and each client that will receive the additional routes.

Internet in a VRF with unique RDs per peering Router

In this design option, all routes from each peering Router is assigned its own Route Distinguisher. Since this makes all routes unique, all routes are forwarded to all PE devices. This also creates a large amount of route information that route reflection was meant to reduce.

Use tunnels to make farther away locations look closer

This option requires a bit of planning and management and, while it can work, it is not a common practice.

Optimal Route Reflection

ORR allows a router in the network to be defined as the primary router and to be the router from which the path costs will be calculated. A secondary router can be defined in case the primary router is down. If a single router cannot be found to provide the best route to all of the other clients, additional peer group clusters can be defined, with their own primary and secondary routers, to localize the best route calculation to parts of the network.

Advantages of Optimal Route Reflection

- Place RR anywhere in the topology
- Solves Hot Potato routing
- No change needed to BGP RR clients
- Can work with Add-Path

© 2020 Juniper Networks, Inc. All Rights Reserved.

Advantages of Optimal Route Reflection

There are a number of advantages to using optimal route reflection. This includes:

- The RR can be placed anywhere in the network. Since ORR takes away the need for the RR to be close to the clients, it can be placed where it can be easily managed and accessed.
- ORR solves the issue of Hot potato routing where traffic needs to exit the network as quickly as possible, allowing the best path to be determined by the location of network device in the network and not by the location of the RR.
- ORR does not require any configuration changes to the RR clients.
- When portions of the network are not able to determine the shortest best path out the network with ORR alone, then ORR can work with add-path to get those devices the additional routes needed to make the best route decision.

Configuring BGP ORR

- Configured at the `[edit protocols bgp group <group-name>]` hierarchy level

```
[edit protocols bgp group <group-name>]
user@router# set optimal-route-reflection
```

- Configure one of the clients as the primary node in a BGP peer group so that the IGP metric from the primary node is used to select the best path.

```
[edit protocols bgp group <group-name> optimal-route-reflection]
user@routes# set igp-primary <ipv4-address>
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Configuring BGP ORR

BGP ORR is configured at the `[edit protocols bgp group <group-name>]` hierarchy level using the `optimal-route-reflection` command.

Once turned on, the ip address of the primary router in the cluster from which the best route will be determined needs to be configured as shown in the slide above.

Agenda: Route Reflection and Confederations

- Route Reflection Operation
- Configuring Route Reflection
- Virtual Route Reflector
- Optimal Route Reflection
- →Route Reflection Case Study
- BGP Confederations

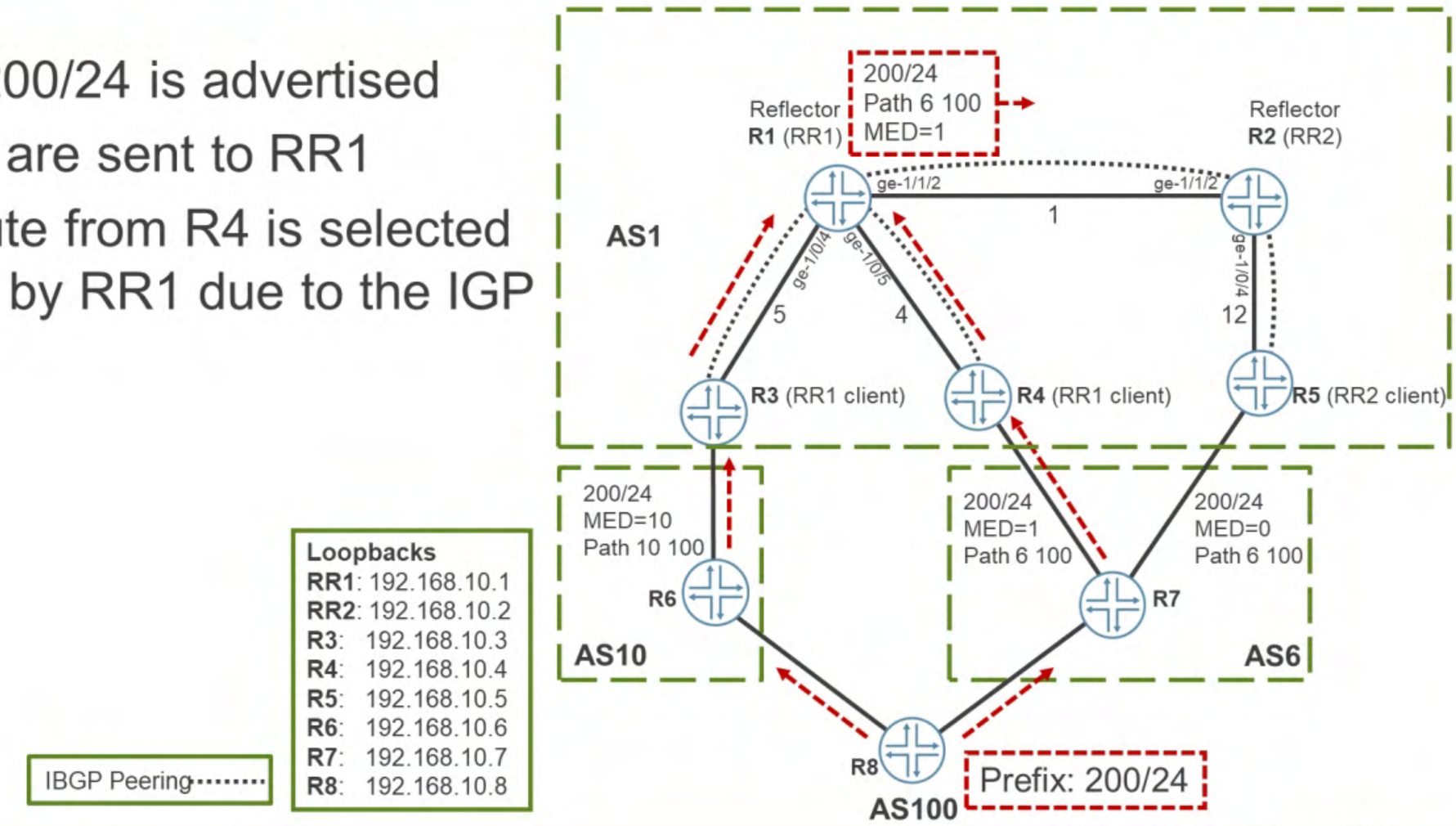
© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Reflection Case Study

The slide highlights the topic we discuss next.

Case Study: RFC3345 Oscillation (1 of 5)

- Prefix 200/24 is advertised
- Routes are sent to RR1
- The route from R4 is selected as best by RR1 due to the IGP metric



Case Study: RFC 3345 Oscillation, Part 1

Some route reflector topologies are susceptible to a persistent prefix oscillation. This oscillation is triggered because of the use of multiple exit discriminators (MEDs) in a route reflector or confederation environment. This problem is not a corner case. In fact, it has happened several times and major Internet backbones were affected by this problem. The issue is simple to describe but requires a working knowledge of BGP path selection.

Consider the sample topology and the following sequence of events:

1. The R8 router is advertising the 200/24 prefix toward the R6 and R7 routers.
2. The R6 and R7 routers apply three different MED values and advertise the route to their peers (R3, R4, and R5).
3. As route reflector clients, R3, R4, and R5 advertise this route to their respective route reflector.
4. RR1 now has visibility of the 200/24 route through both R3 and R4, and has chosen R4 (through ge-1/0/5) as the best exit point due to IGP metric (4 compared to 5). It also advertises that best path to route reflector RR2. You can view sample outputs from the RR1 router on the following page.

Continued on the next page.

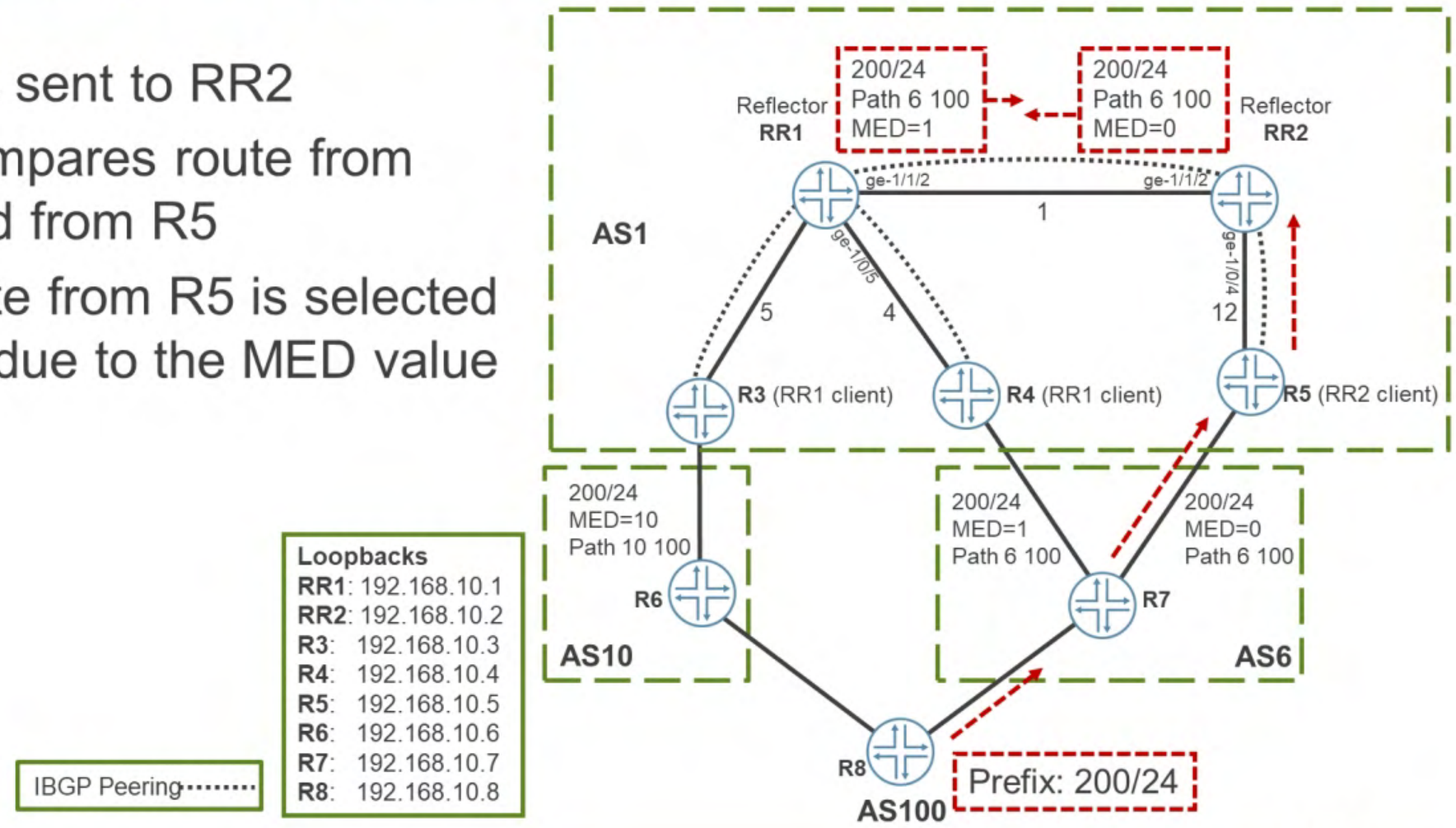
Case Study: RFC 3345 Oscillation, Part 1 (contd.)

The following are sample outputs from the RR1 router on the preceding page:

```
user@RR1> show route 200/24
inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
200.0.0.0/24       *[BGP/170] 00:00:01, MED 1, localpref 100, from 192.168.10.4
                     AS path: 6 100 I, validation-state: unverified
                    > to 20.0.0.2 via ge-1/0/5.0
                    [BGP/170] 00:00:01, MED 10, localpref 100, from 192.168.10.3
                     AS path: 10 100 I, validation-state: unverified
                    > to 10.0.0.2 via ge-1/0/4.0

user@RR1> show route advertising-protocol bgp 192.168.10.2
inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 200.0.0.0/24            192.168.10.4         1       100        6 100 I
```

Case Study: RFC3345 Oscillation (2 of 5)

- Route is sent to RR2
- RR2 compares route from RR1 and from R5
- The route from R5 is selected as best due to the MED value



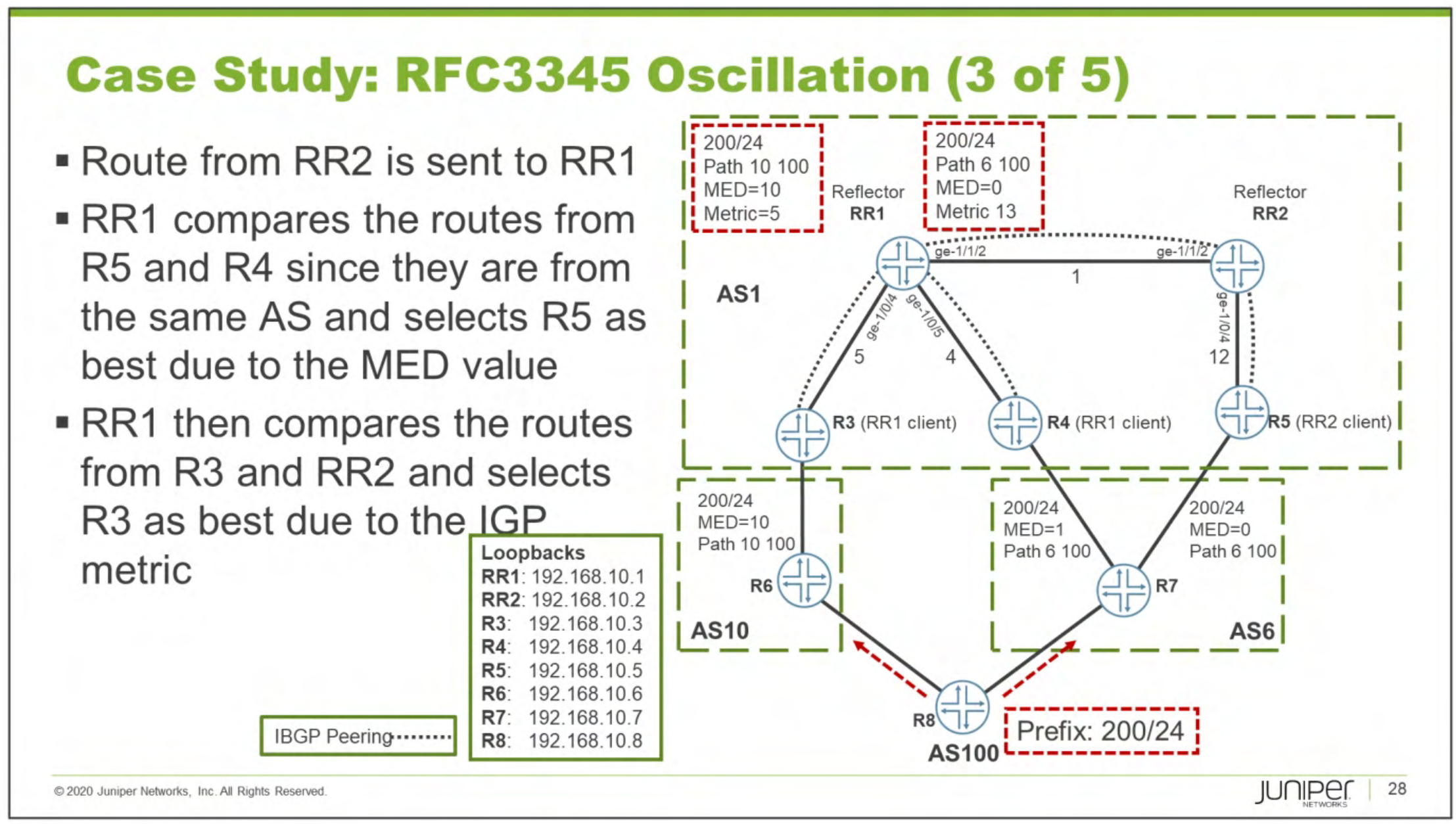
Case Study: RFC 3345 Oscillation, Part 2

Our sequence of events continues as follows:

5. In addition to the RR1 advertisement, route reflector RR2 has also has visibility of the 200/24 route through the R5 router.

6. RR2 chooses the path through R5 as best path because of the lower MED value (0 compared to 1), and advertises it back to RR1. Up to this point, there are no problems. Sample output from RR2 follows.

```
user@RR2> show route 200/24
inet.0: 13 destinations, 14 routes (13 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200.0.0.0/24       *[BGP/170] 01:18:01, MED 0, localpref 100, from 192.168.10.5
                      AS path: 6 100 I, validation-state: unverified
                    > to 40.0.0.2 via ge-1/0/4.0
                    [BGP/170] 00:00:00, MED 1, localpref 100, from 192.168.10.1
                      AS path: 6 100 I, validation-state: unverified
                    > to 30.0.0.1 via ge-1/1/2.0

user@RR2> show route advertising-protocol bgp 192.168.10.1
inet.0: 13 destinations, 14 routes (13 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path

* 200.0.0.0/24            192.168.10.5         0       100        6 100 I
```



Case Study: RFC 3345 Oscillation, Part 3

Our sequence of events continues as follows:

7. RR1 now has to compare three routes to the 200/24 prefix; one through R3, one through R4, and the last one through the other route reflector, RR2.

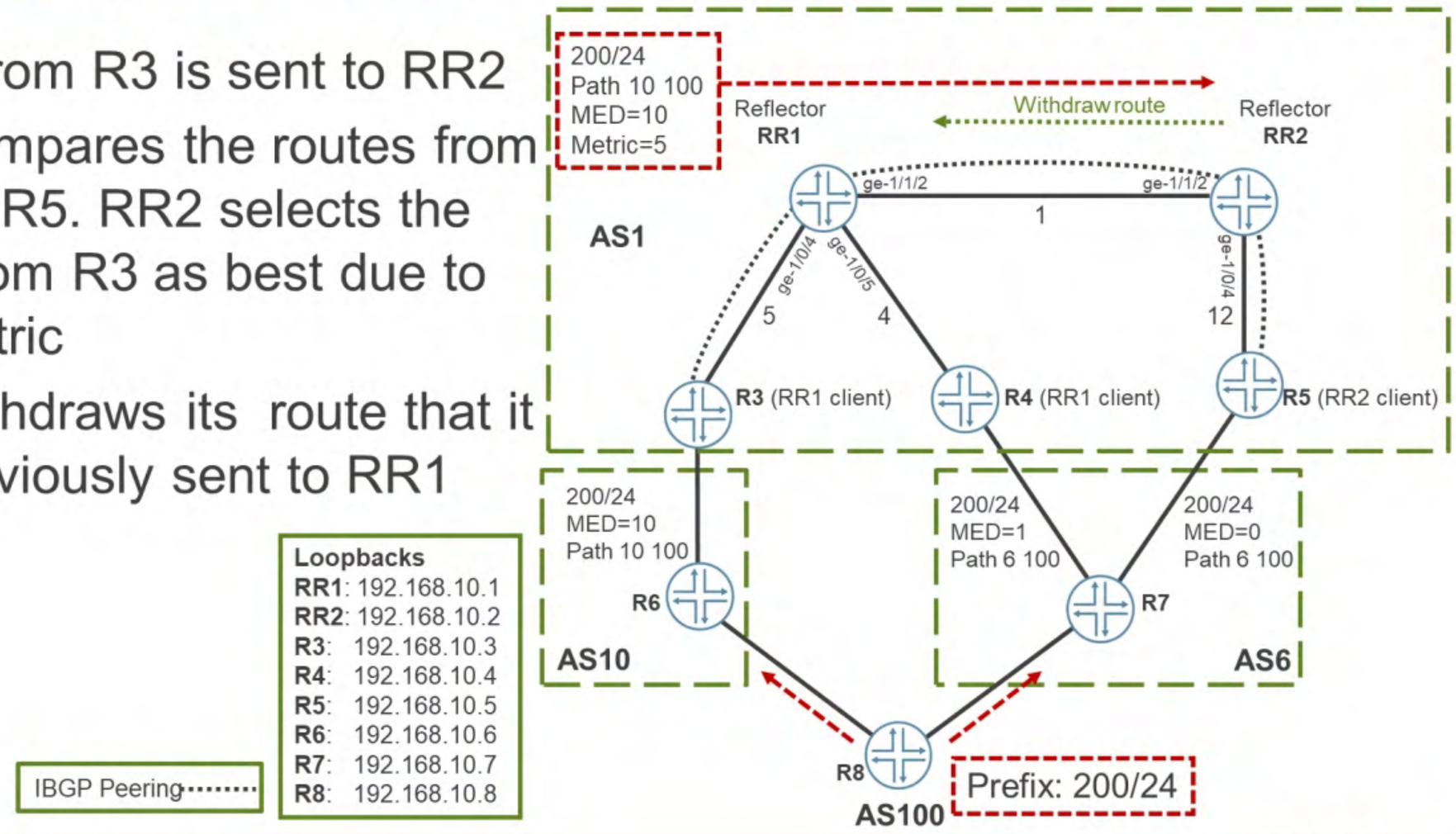
8. Following the BGP decision process, RR1 considers the route through the RR2-R5 path better than the path through R4 because of the lower MED value (0 compared to 1). However, the route through R3 is then considered best, because of the lower IGP cost (5 compared to 13). Keep in mind that the two routes come from different autonomous systems, meaning the MED values are not considered.

9. RR1 then advertises the preferred path through R3 to RR2 as shown in the following output:

```
user@RR1> show route advertising-protocol bgp 192.168.10.2

inet.0: 14 destinations, 15 routes (14 active, 1 holddown, 0 hidden)
    Prefix                  Nexthop              MED     Lclpref    AS path
* 200.0.0.0/24            192.168.10.3          10      100        10 100 I
```

Case Study: RFC3345 Oscillation (4 of 5)

- Route from R3 is sent to RR2
- RR2 compares the routes from R3 and R5. RR2 selects the route from R3 as best due to IGP metric
- RR2 withdraws its route that it had previously sent to RR1

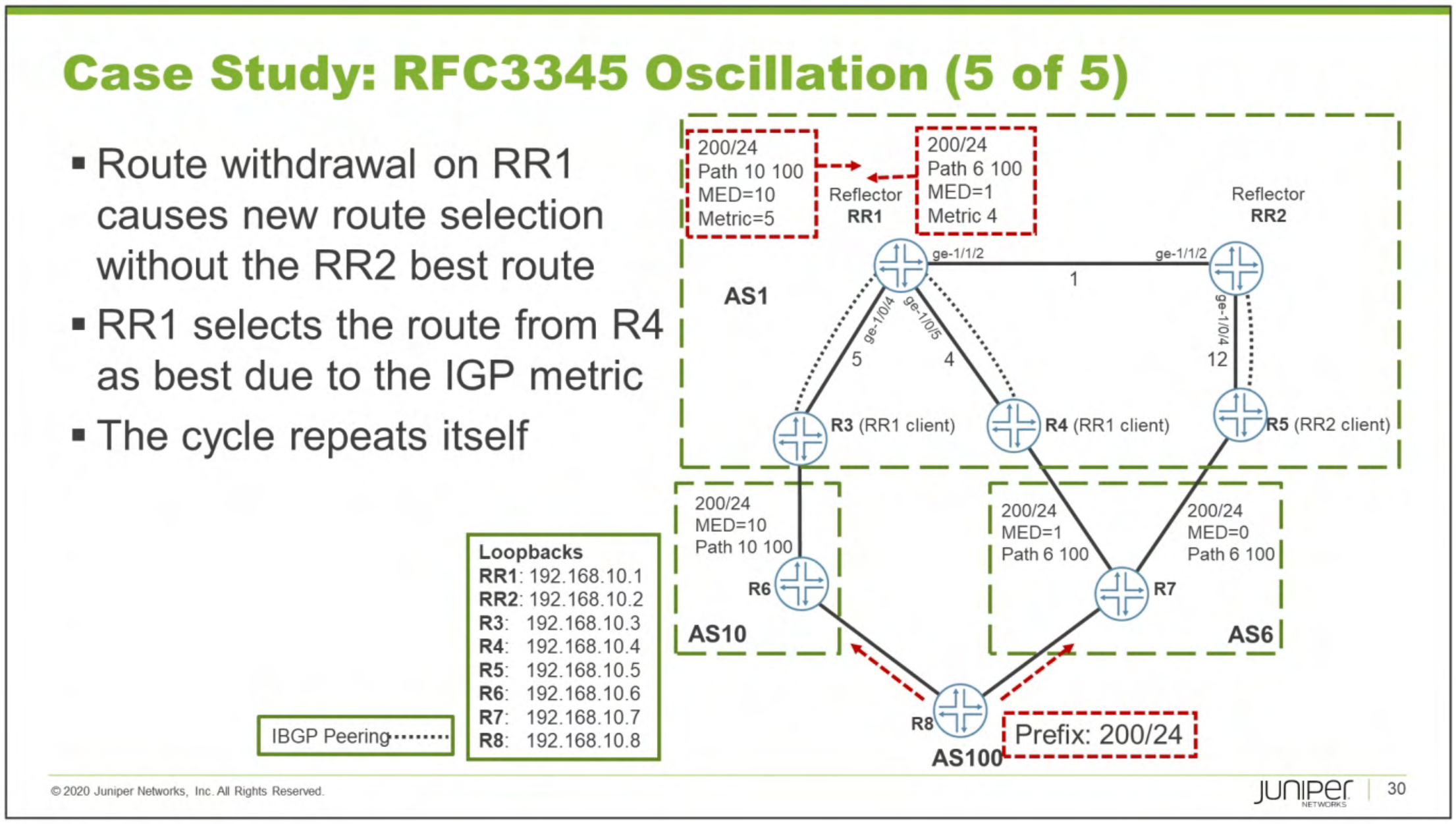


© 2020 Juniper Networks, Inc. All Rights Reserved.

Case Study: RFC 3345 Oscillation, Part 4

The following are the final steps in our sequence of events:

10. At this point, RR2 now compares the path through RR1 and R3 with the one through R5; the winner is the route through RR1 and R3 due to a lower IGP metric (6 compared to 12).
11. This causes RR2 to stop advertising the path through R5 to RR1. RR1 is now left with the two original paths, one through R3 and one through R4.

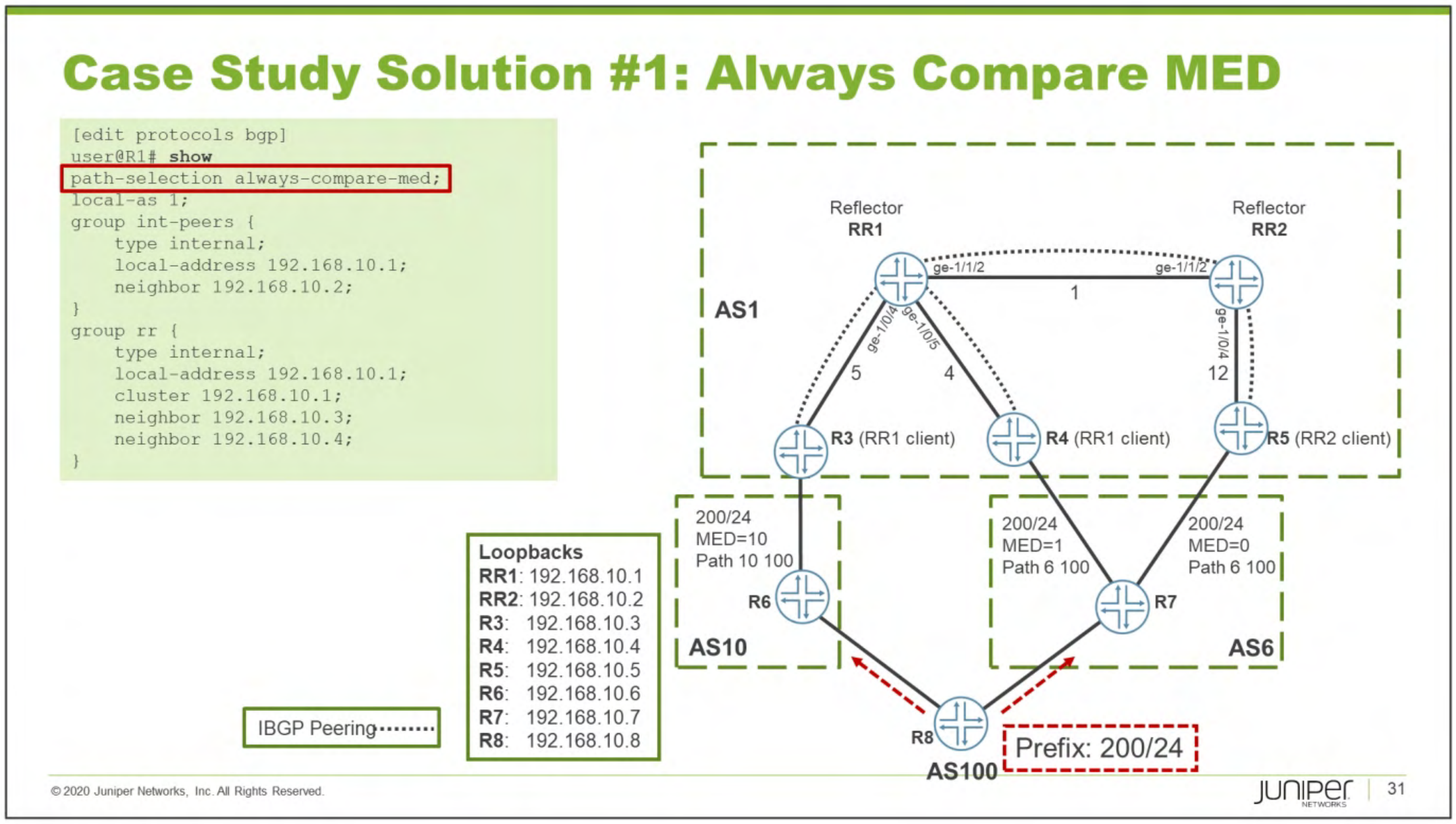


Case Study: RFC 3345 Oscillation, Part 5

We are back where we started. RR1 picks the route through R4 and the cycle repeats itself.

In the following sample output, you can view the output from RR1 that demonstrates the route churn described in this case study. Take note of the route timestamps; the output encompasses only seven seconds:

```
user@RR1> show route 200/24

inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200.0.0.0/24       *[BGP/170] 01:19:08, MED 1, localpref 100, from 192.168.10.4
                      AS path: 6 100 I, validation-state: unverified
                    > to 20.0.0.2 via ge-1/0/5.0
                    [BGP/170] 01:19:08, MED 10, localpref 100, from 192.168.10.3
                      AS path: 10 100 I, validation-state: unverified
                    > to 10.0.0.2 via ge-1/0/4.0

user@RR1> show route 200/24

inet.0: 14 destinations, 16 routes (14 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200.0.0.0/24       +[BGP/170] 01:19:10, MED 10, localpref 100, from 192.168.10.3
                      AS path: 10 100 I, validation-state: unverified
                    > to 10.0.0.2 via ge-1/0/4.0
                    [BGP/170] 00:00:00, MED 0, localpref 100, from 192.168.10.2
                      AS path: 6 100 I, validation-state: unverified
                    > to 30.0.0.2 via ge-1/1/2.0
                   -[BGP/170] 01:19:10, MED 1, localpref 100, from 192.168.10.4
                      AS path: 6 100 I, validation-state: unverified
                    > to 20.0.0.2 via ge-1/0/5.0
```

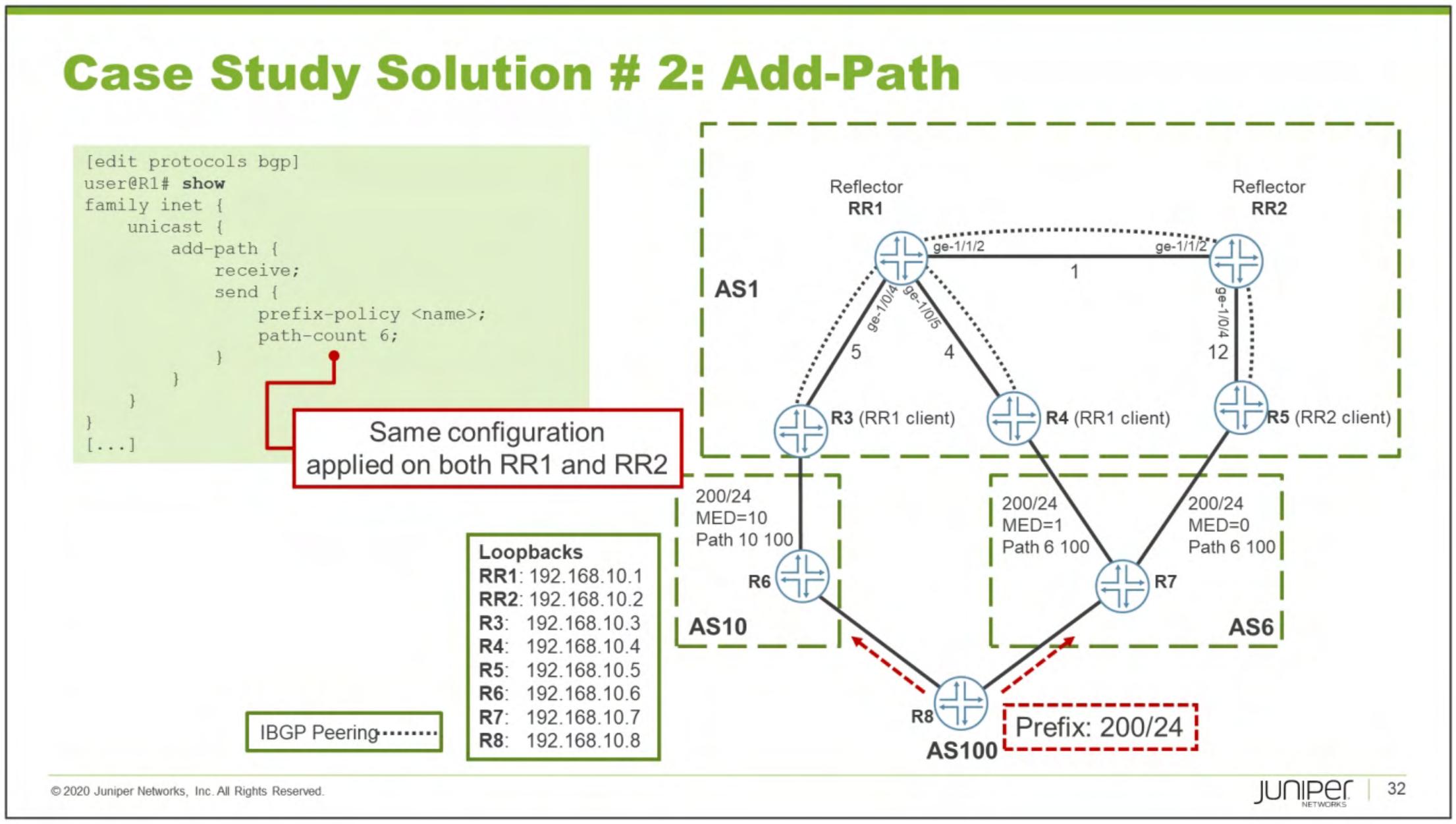


Case Study Solution: Always Compare MED

In real-life scenarios, the topology can be much more complex. For example, multiple levels of route reflection, or oscillations between more than two clusters. Some potential workarounds are using the **always-compare-med** parameter, using the **add-path** option, or making sure that the metric between routers in different clusters is higher than the metric between routers within the same cluster.

The **always-compare-med** option is demonstrated on the slide. You configure this option with the **set protocols bgp path-selection always-compare-med** statement. Once committed, the oscillation problem is solved. The following sample output shows the 200/24 route after this fix has been applied. The path with the lowest MED value is chosen.

```
user@RR1> show route 200/24

inet.0: 14 destinations, 15 routes (14 active, 0 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both
200.0.0.0/24       *[BGP/170] 00:00:16, MED 0, localpref 100, from 192.168.10.2
                     AS path: 6 100 I, validation-state: unverified
                    > to 30.0.0.2 via ge-1/1/2.0
                    [BGP/170] 00:00:12, MED 10, localpref 100, from 192.168.10.3
                     AS path: 10 100 I, validation-state: unverified
                    > to 10.0.0.2 via ge-1/0/4.0
```



Case Study Solution: Add-Path

There might be times when using the **always-compare-med** option is not suitable. As mentioned previously, another way to solve this oscillation problem is with the **add-path** option. This option was introduced in Junos OS Release 11.3 and allows a BGP router to advertise multiple paths instead of advertising only the active path.

As shown on the slide, you can configure an Add-Path speaker to be a receiver only, sender only, or both (as shown). You can also configure the number of paths to advertise, up to a maximum of 6. Furthermore, you can use the **prefix-policy** option to control which routes get distributed using the Add-Path algorithm.

The following output is sample output from the RR1 router after configuring the **add-path** option on both the RR1 and RR2 routers. Note that all three paths are now present and the oscillation issue has stopped.

```
user@RR1> show route 200/24

inet.0: 14 destinations, 16 routes (14 active, 1 holddown, 0 hidden)

+ = Active Route, - = Last Active, * = Both

200.0.0.0/24       *[BGP/170] 00:14:56, MED 10, localpref 100, from 192.168.10.3
                      AS path: 10 100 I, validation-state: unverified
                    > to 10.0.0.2 via ge-1/0/4.0
                    [BGP/170] 00:14:42, MED 0, localpref 100, from 192.168.10.2
                      AS path: 6 100 I, validation-state: unverified
                    > to 30.0.0.2 via ge-1/1/2.0
                    [BGP/170] 00:14:52, MED 1, localpref 100, from 192.168.10.4
                      AS path: 6 100 I, validation-state: unverified
                    > to 20.0.0.2 via ge-1/0/5.0
```

Agenda: Route Reflection and Confederations

- Route Reflection Operation
- Configuring Route Reflection
- Virtual Route Reflector
- Optimal Route Reflection

→BGP Confederations

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Confederations

The slide highlights the topic we discuss next.

Scaling BGP—Confederations

- Breaks a global AS into multiple pieces (sub-AS)
- Within each sub-AS:
  - Use private AS numbers
  - An IBGP full-mesh topology is still required
- Between each sub-AS:
  - EBGP-type configurations are required (multihop, and so forth)
  - Only the AS path attribute is changed
    - Prevents loops in the network
    - Sub-AS networks are *not* used when comparing AS path lengths
  - Other BGP attributes are not modified by default
    - Next hop, local preference, and MED are all unaffected

© 2020 Juniper Networks, Inc. All Rights Reserved.

Break Up the Global AS

A confederation takes a global AS and breaks it into smaller subautonomous systems which are interconnected through an new BGP peering type referred to as a *confederation BGP* (CBGP) peering.

Bear in mind that confederations are considered legacy and are almost never deployed because of two inherent limitations:

1. It is not possible to migrate to or from a confederation setup without a complete BGP shutdown; and
2. Unlike route reflectors, confederations are not scalable in regard to services and address families; you could, for example, have a dedicated set of route reflectors for Layer 3 virtual private networks (VPNs), and another set for VPLS services, but there is no way of achieving this with confederations.

Within a Sub-AS

The confederation sub-AS networks act just like a *real* AS; they require a full mesh of IBGP connectivity within themselves. Should the full mesh of the sub-AS grow too large, route reflection might be used within a sub-AS to scale the network.

Each sub-AS must have a unique AS number defined, and most administrators use a private AS number from the 64512 to 65535 range.

Between Each Sub-AS

CBGP is a special type of EBGP; certain attributes, such as the BGP next hop, are handled differently across CBGP sessions.

CBGP peers modify the AS path attribute to include the sub-AS numbers. This modification is performed to provide loop prevention within only the confederation network. All other BGP attributes, such as local preference and the BGP next hop, remain unchanged when sent across a CBGP link.

Because the router views connections between the sub-AS networks as being EBGP, some special configuration might be warranted. The router expects to use the physical address of the CBGP for the BGP session, but many administrators prefer to peer the CBGP routers using loopback addresses. This is accomplished through the use of the **`multihop`** command.

Confederation AS Path Segments

- AS confederation sequence:
  - Each sub-AS is added to the AS path attribute
  - `(65000 65001 65002) 100 200` shows a sequence
  - Used for loop prevention only
  - Sequence values are not counted as AS hops
- AS confederation set is used when an aggregated route loses the granularity of the sequence:
  - 192.168.24.0/24 `(65000 65001) 100`
  - 192.168.100.0/24 `(65000 65002) 100`
  - 192.168.0.0/16 `({65000 65001 65002}) 100`

© 2020 Juniper Networks, Inc. All Rights Reserved.

AS Confederation Sequence

As a route is advertised over a CBGP link, BGP modifies the AS path attribute to include the sub-AS number. BGP places this sub-AS number into an AS confederation sequence, as denoted by parentheses, within the AS path attribute. The sequence is a new AS path segment attribute with a type code of 3.

The sub-AS values are sequenced in the order in which the route has traversed the network, with the primary purpose being loop prevention within the confederation network. The confederation sequence is *not* used in the calculation of AS path length for the BGP active route selection algorithm. For routers within a confederation network, the confederation sequence appears as a single, internal BGP AS network.

AS Confederation Set

Should some routing aggregation occur within the confederation network, the granularity of the confederation sequence might be lost. This process is very similar to conventional BGP route aggregation.

In this situation, the AS confederation sequence becomes an AS confederation set and is denoted by curly braces within the AS path output. The set is also a new AS path segment attribute with a type code of 4.

The routers within the confederation view the AS confederation set in the same manner as a sequence. That is, the set is still used for loop prevention even though the ordering of the sub-AS numbers is no longer significant.

Confederation Configuration

- The global AS appears as a whole network when viewed externally by peer networks
- All routers remove all confederation information at the edge of the global AS
  - Other AS peers do not see the details of the confederation
  - No need for **remove-private**

```
[edit routing-options]
user@router# show
autonomous-system 65000;
confederation 201 members [ 65000 65001 65002 65003 65004 ];
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Global AS Appears Whole

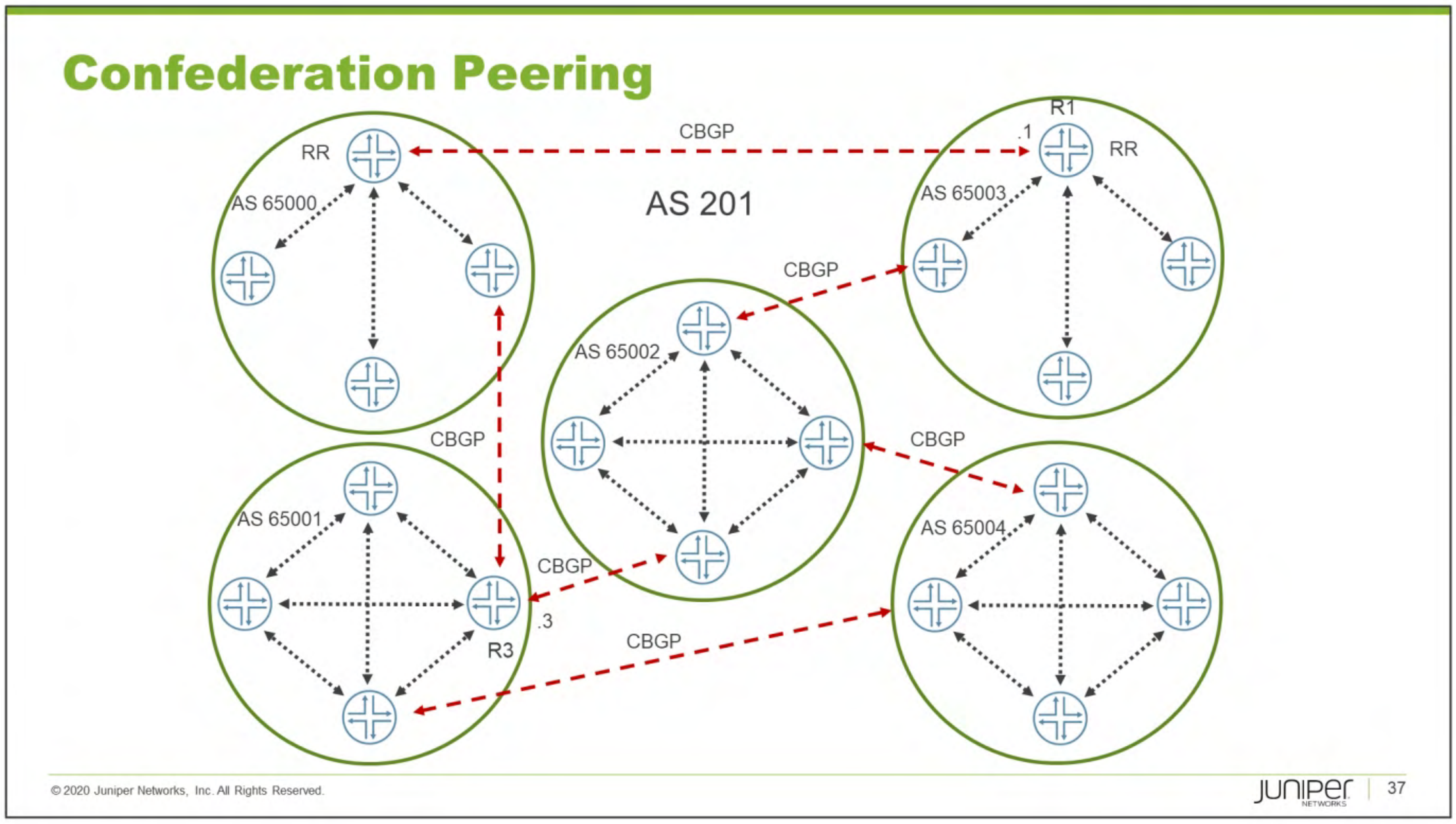
The Internet views the confederation as a single autonomous system. The AS path received by other autonomous systems contains only the globally assigned AS number. The AS path contains only this number because all sub-AS numbers are removed from the AS path attribute as the route is advertised to EBGP peers.

To operate a confederation network effectively, all BGP routers in the AS must know the globally unique AS number as well as all the configured sub-AS numbers. This information is defined using the **confederation** command within the `routing-options` stanza, as shown on the slide.

Confederation Information Removed

At the edge of the confederation network, the routers remove all confederation-related AS numbers, both sequences and sets, from the AS path attribute of all routes prior to EBGP advertisement. This removal allows the details of the confederation network to be hidden from all other networks in the Internet.

Note that the **remove-private** command is not required to remove sub-AS numbers when you operate a confederation network. This behavior is the default for a BGP confederation and is controlled by the **confederation** command.



Confederation Peering

The slide shows an example of a BGP confederation network. The global AS of 201 is split up into five sub-AS networks—65000, 65001, 65002, 65003, and 65004. The sub-AS networks are connected with EBGP-like connections (sometimes called CBGP). Because the CBGP links behave like EBGP, there is no need for a full-mesh topology between each sub-AS. Therefore, you can provision CBGP connections wherever it is logically or physically convenient.

Some of the sub-AS networks on the slide are using route reflection within the sub-AS to eliminate the need for a full IBGP mesh. The combination of BGP confederations *and* route reflection simultaneously allows for great flexibility in scaling your AS to hundreds or thousands of routers.

Configuration of Peers

```
[edit]
user@AS65001-R3# show protocols bgp
group sub-AS-65001 {
    type internal;
    local-address 192.168.1.3;
    neighbor 192.168.1.1;
    neighbor 192.168.1.2;
    neighbor 192.168.1.4;
}
group sub-AS-65000 {
    type external;
    multihop;
    local-address 192.168.1.3;
    peer-as 65000;
    neighbor 192.168.0.3;
}
group sub-AS-65002 {
    type external;
    multihop;
    local-address 192.168.1.3;
    peer-as 65002;
    neighbor 192.168.2.4;
}
```

```
[edit]
user@AS65003-R1# show protocols bgp
group sub-AS-65000 {
    type external;
    multihop;
    local-address 192.168.3.1;
    peer-as 65000;
    neighbor 192.168.0.1;
}
group sub-AS-65003 {
    type internal;
    local-address 192.168.3.1;
    cluster 192.168.3.1;
    neighbor 192.168.3.2;
    neighbor 192.168.3.3;
    neighbor 192.168.3.4;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Peering Session Configurations

The configuration example on the left side of the slide represents Router 3 in sub-AS 65001. A full mesh of IBGP peering sessions exists within the sub-AS, as seen in peer group *sub-AS-65001*. The remaining peer groups on that router represent CBGP connections to other sub-AS networks in the confederation. Each group uses a connection type of `external`, uses the **`multihop`** command, and configures both a peer and local AS value.

The configuration example on the right side of the slide represents Router 1 in sub-AS 65003. This sub-AS uses route reflection to replace the IBGP full mesh, and Router 1 is the route reflector for the cluster. This configuration is seen in peer group *sub-AS-65003* where the **`cluster`** command is configured. The other peer group on the router represents the CBGP peering connection to sub-AS 65000.

Summary

- In this content, we:
  - Described the operation of BGP route reflection
  - Learned how to configure a route reflector
  - Described the virtual route reflector
  - Described the operation of optimal route reflection
  - Described the operation of a BGP confederation
  - Learned how to configure BGP confederations
  - Described peering relationships in a BGP confederation

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- Operation of BGP route reflection;
- How to configure a route reflector;
- The virtual route reflector;
- The operation of optimal route reflection;
- The operation of a BGP confederation;
- How to configure BGP confederations; and
- Peering relationships in a BGP confederations.

Review Questions

1. Which two BGP attributes support route reflection?
2. How is a cluster configured in Junos?
3. In a fully-meshed cluster, which command can you use to stop excess advertisements?
4. What does a BGP route reflector do with BGP routes received from clients and non-clients?
5. How are loops prevented in a confederation network?
6. Which form of BGP is run between:
   1. Routers within a sub-AS?
   2. Routers across a confederation boundary?
   3. Routers external to the confederation?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

3.

4.

5.

6.

Lab: Scaling BGP

- Configure and monitor route reflection and confederations

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: Scaling BGP

The slide shows the objectives for the lab.

Answers to Review Questions

1.

Cluster ID and Cluster List support route reflection.

2.

The **`cluster`** statement is configured only on the route reflector.

3.

`No-client-reflect` can be used to stop excess advertisements.

4.

An RR readvertises routes received from clients and non-clients, adding the cluster ID and cluster list attributes.

5.

Loops are prevented by examining the confederation AS sequence or AS set.

6.

Routers in a sub-AS run IBGP. Routers across a confederation boundary run CBGP. Routers external to the confederations run EBGP.

Education Services

Advanced Junos Service Provider Routing

Chapter 14: BGP FlowSpec

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - Describe DDoS attacks
  - Describe DDoS mitigation techniques
  - Describe FlowSpec Operations
  - Configure and Monitor FlowSpec

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- DDoS attacks;
- DDoS mitigation techniques;
- FlowSpec operations; and
- How to configure and monitor FlowSpec.

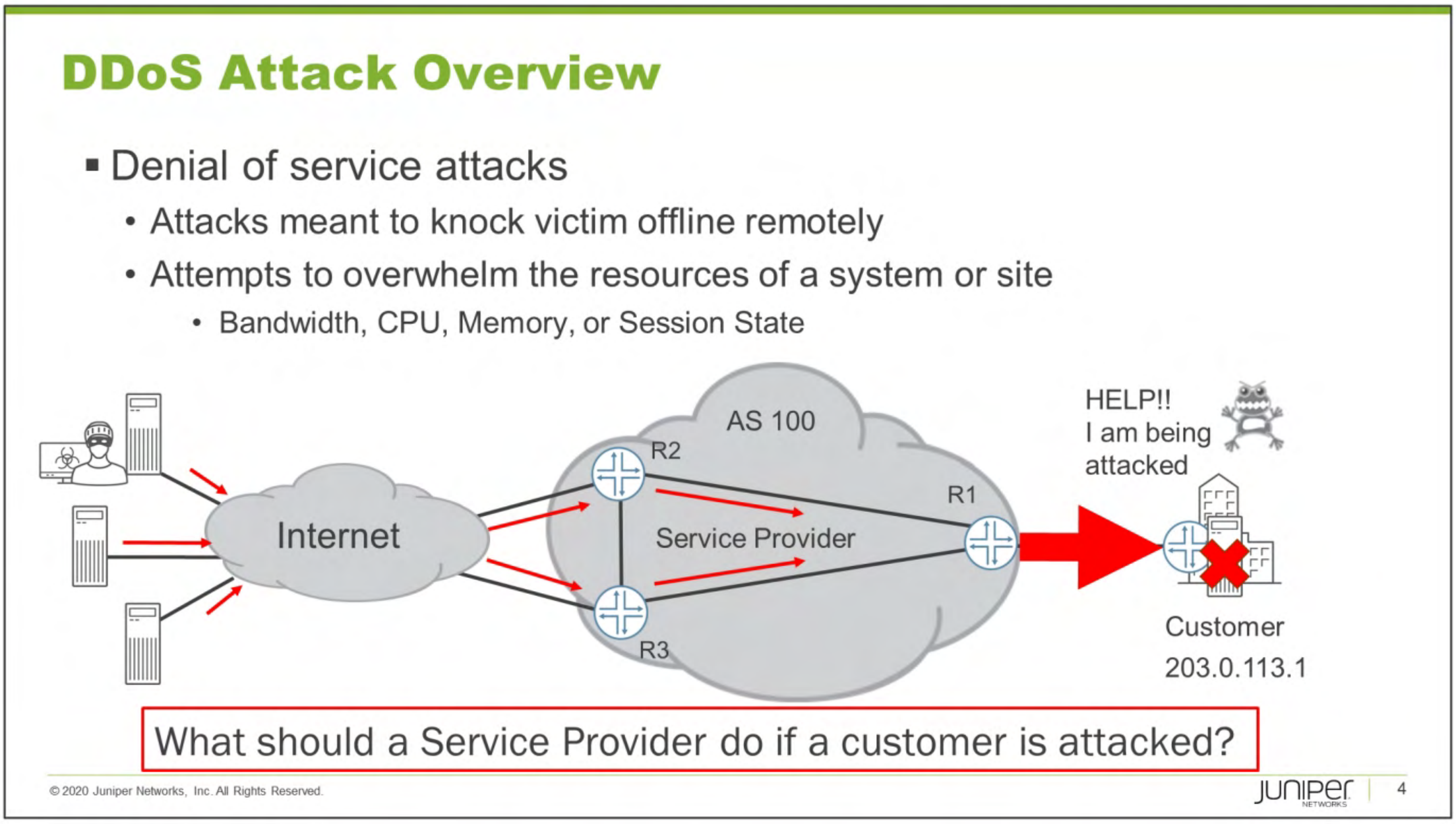
Agenda: BGP FlowSpec

→DDoS Overview
- DDoS Mitigation Techniques
- FlowSpec Overview
- Configuring and Monitoring FlowSpec

© 2020 Juniper Networks, Inc. All Rights Reserved.

DDoS Overview

The slide lists the topics we will discuss. We discuss the highlighted topic first.



DDoS Attack Overview

A distributed denial of service (DDoS) attack consists of two or more sources being used to temporarily or permanently deny the legitimate use of services or access to hosts connected to a network. DDoS attacks flood a network with traffic that overwhelms the available bandwidth or consumes a system's resources which prevent it from performing its function. A common endpoint of a DDoS attack is an organization's Website or other public facing interfaces.

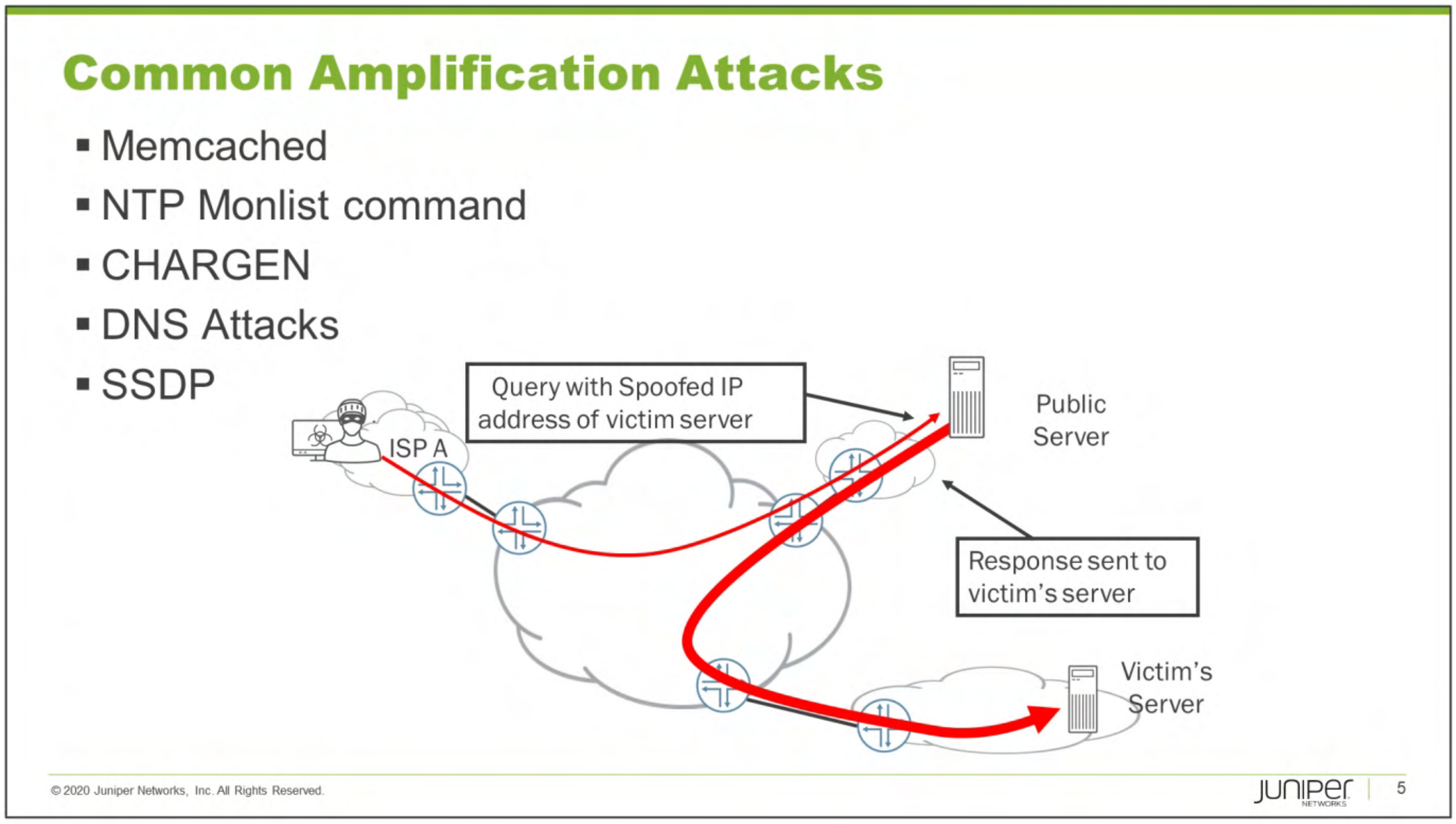
DDoS attacks are commonly launched from devices which are under the control of a command and control system (known as a botnet) that instructs the devices where and how to attack a targeted network. When a botnet, comprised of thousands or tens of thousands of compromised servers, is focused at a single resource, it is easy to overwhelm that resource.

Smaller botnets can achieve similar results through the use of a type of DDoS attack called an amplification attack. An amplification attack occurs when a small UDP-based request is made to a public DNS, or NTP system, etc. using the source address of the victim. This results in a much larger response directed to the victim's device in an attempt to overwhelm it's resources. The size of the request is small compared to the size of the response, which makes this a common DDoS attack tool.

Another newer style of DDoS attack is described as "low and slow" in which the attack is directed against the applications that are being run by the victim. These attacks are not so much about a brute force attempt to overwhelm available resources, but are meant to quietly consume resources over time to avoid detection until they consume sufficient resources to prevent legitimate requests. These are not as easy to detect but the results are the same.

Motivations for using DDoS attacks against an organization span a wide range of reasons including, cybercrime, extortion, activism, and more. Regardless of the reason, DDoS attacks are very disruptive and require immediate attention.

While DDoS attacks are normally short in duration, being measured in hours as opposed to days, weeks, or months, just waiting until they dissipate is generally not an acceptable choice. Since these attacks fill the incoming connections, mitigation usually requires external help to resolve the issue, such as the ISP.



Amplification Attacks

All amplification attacks take advantage of a difference in the bandwidth usage between an attacker's request to a public device and the public device's response which is directed to the victim. This difference can be magnified by repeating the requests over and over until the resulting volume of traffic disrupts the victim's network infrastructure.

Common amplification attacks include:

- *Memcached attack:* Memcached is a database caching system for speeding up websites and networks. It works by an attacker implanting a large "payload" of data on an open memcached server. When the attacker sends an HTTP GET request with the victim's source address, the memcached server responds with a large response. The amplification factor of this attack can be anywhere from 10,000 to 50,000.
- *NTP monlist command:* The monlist command is used for debugging and allows the user to retrieve information from the Network Time Protocol (NTP) service. The command output contains the IP address of the last NTP clients or servers the service has talked to. While the list has a limit of 600 entries, the size of the initial request is small compared to the information returned, resulting in an amplification factor of 556.
- *CHARGEN:* A service of the Internet Protocol Suite that is intended for testing, debugging, and measurement purposes. In the UDP implementation of the protocol, the server sends a UDP datagram containing a random number of characters (0-512) everytime it receives a datagram from the sending host. The amplification factor of this attack is 358.
- *DNS Amplification:* A DDoS in which the attacker exploits vulnerabilities in Domain Name Service (DNS) servers to turn initially small queries into much larger payloads which can overwhelm a victim's servers. The amplification factor of this type of attack is between 28 to 54.
- *Simple Service Discovery Protocol:* A DDoS attack that exploits the Universal Plug and Play (UPnP) network protocols for sending an amplified traffic stream to a victim's server. The SSDP protocol allows a discovered plug and play device to respond to a request with a detailed description of its services. The amplification factor of this attack is 30.

Additional amplification is achieved by having each bot in a botnet make similar requests, further protecting the true identity of the attacker and reaping the benefits of greatly increased attack traffic.

There are various networking tools to help identify and define the attack that is in progress. This includes SNMP for tracking traffic variations and traffic sampling using JFlow, sFlow, and IPFIX.

Agenda: BGP FlowSpec

- DDoS Overview
- →DDoS Mitigation Techniques
- FlowSpec Overview
- Configuring and Monitoring FlowSpec

© 2020 Juniper Networks, Inc. All Rights Reserved.

DDoS Mitigation Techniques

The slide highlights the topic we discuss next.

DDoS Attack Mitigation

- Attack Mitigation
  - Unicast Reverse Path Forwarding (Anti-Spoofing)
  - Destination-based Remote Triggered Black Hole
  - Source-based Remote Triggered Black Hole
- A Better Solution
  - BGP FlowSpec

© 2020 Juniper Networks, Inc. All Rights Reserved.

DDoS Attack Mitigation

The slide lists some of the ways to mitigate DDoS attacks. These include:

- Unicast Reverse Path Forwarding (uRPF);
- Destination-based Remote Triggered Black Hole (RTBH);
- Source-based Remote Triggered Black Hole; and
- BGP FlowSpec.

We will cover these solutions over the next few pages.

Unicast Reverse Path Forwarding

- Internet Standard – Anti Spoofing
  - BCP 38 (Best Common Practice)
  - RFC 2827
  - RFC 3704
- Verifies that the source IP of the incoming packet has a resolvable route in the routing table
- Multiple operational modes
  - Strict mode
    - The interface that the packet is received on must be the best and active path back to the source prefix.
  - Loose mode
    - Source address must match a prefix in the routing table (accommodates asymmetric routing)

© 2020 Juniper Networks, Inc. All Rights Reserved.

Unicast Reverse Path Forwarding

IP spoofing is often used during a DDoS attack. IP spoofing allows an attacker to create IP packets with a source address that is not the actual address of the sender. This results in traffic being sent to a destination but the response being returned to somewhere else. This modified source address may be the actual victim of an attack or it used to hide the true identity of the attacker. If an attacker uses TCP the potential damage is limited, as a three way handshake is required before the TCP connection is complete and data can be exchanged. However, by using UDP, which is connectionless, an attacker can send or cause to be sent large volumes of traffic to overwhelm an unprotected device.

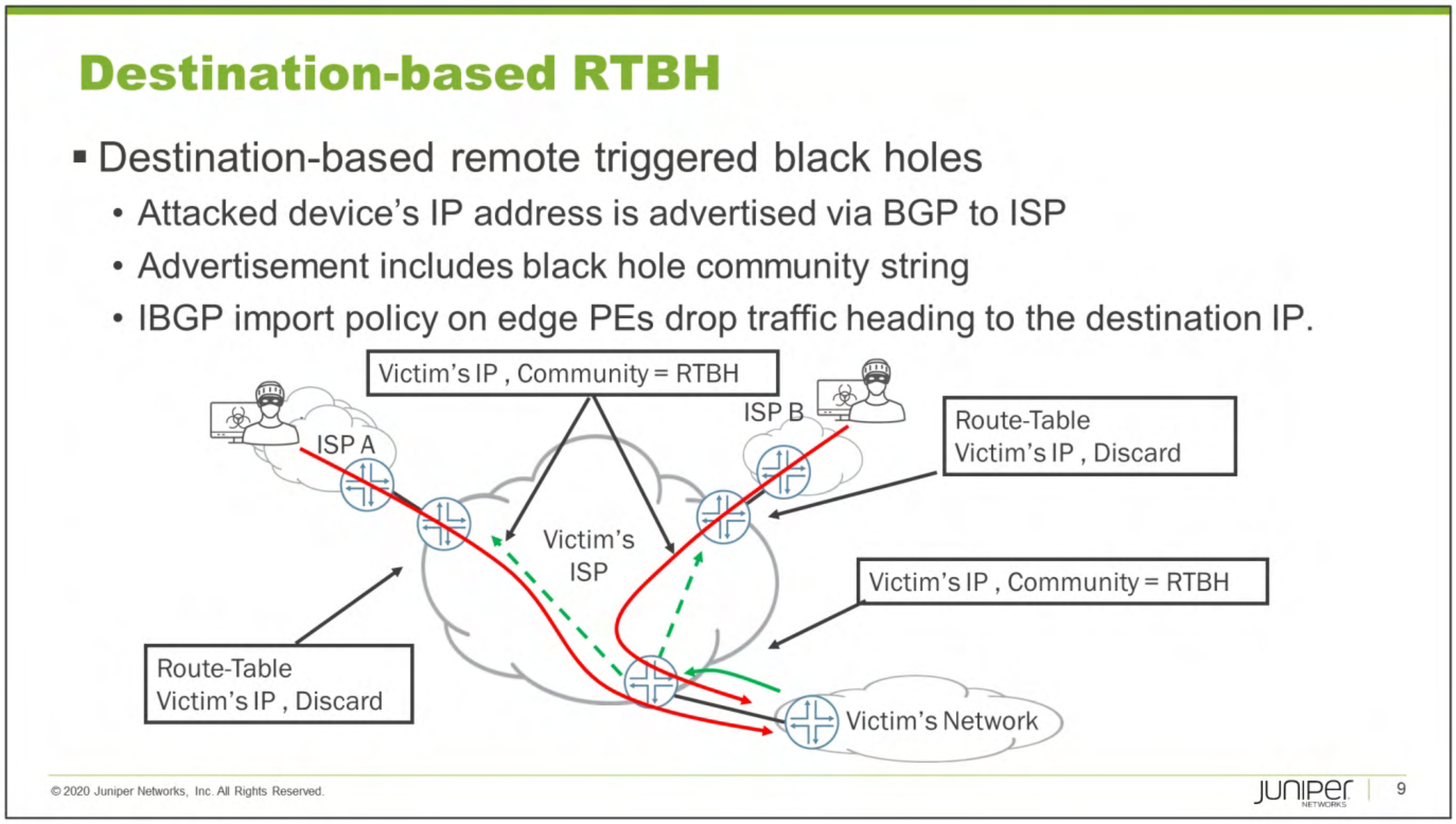
A unicast reverse-path forwarding (uRPF) check is a tool used to reduce the forwarding of IP packets that might be spoofing an address. A uRPF check performs a route table lookup on an IP packet's source address, and determines whether the packet is arriving along a path that the device would use to return traffic to that source address. If the packet passes the check, the packet is forwarded to the destination address. If it fails the check, the packet is discarded.

Multiple Modes

uRFP can work in one of two modes: Strict and Loose.

Strict Mode functions by requiring packets to arrive on an interface that is in the same subnet as the source address or the receiving interface is the best forwarding path to reach the source address. This mode is commonly used on access facing ports.

Loose Mode only requires the route to have a path to the source address in the Forwarding Information Base (FIB) for the packet to be allowed, ignoring the ingress interface that was used by the traffic. This allows for asymmetric routing in a multihomed environment.



Remote Triggered Black Holes

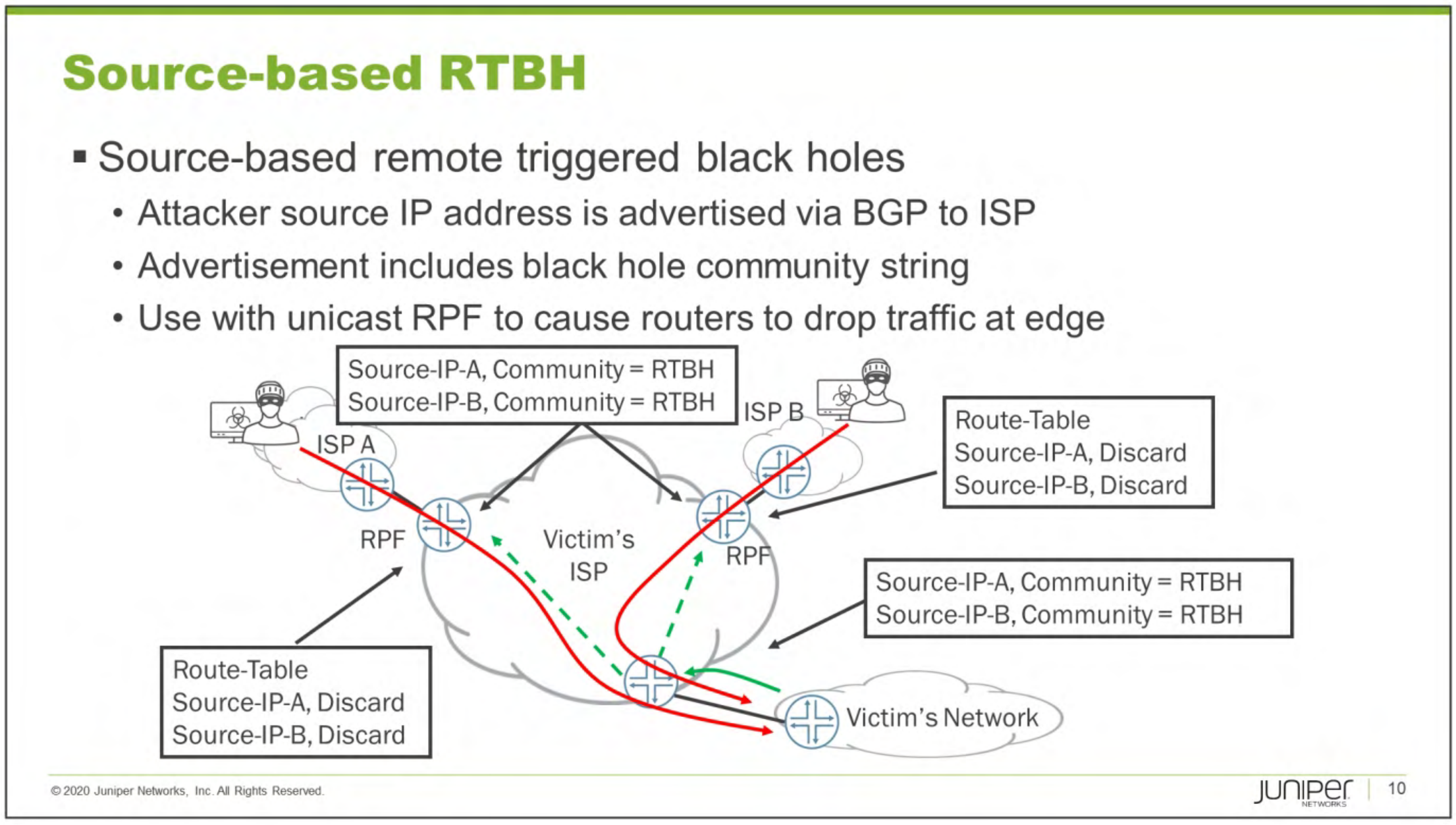
Remote triggered black holes (RTBH) are a vendor-agnostic solution to mitigate DDoS attacks. It relies on BGP and BGP policies for it's implementation. RTBH provides a way for traffic to be dropped based on an attacker's (source) or victim's (destination) address.

Once an attack is discovered, it is communicated to the service provider and RTBH is initiated. How the attack is communicated to the service provider depends on the relationship between the service provider and the customer. One customer may need to reach out by phone or through a web-site to report the issue. Another customer may be trusted and will be able to add a custom community to a route which will be recognized by the service provider as a RTBH notification.

RTBH is configured as either source-based or destination-based.

Destination-based RTBH

Destination-based RTBH was standardized in RFC 3882 in 2004. It provides a way to advertise a prefix with a RTBH community that will cause the network to block traffic at the edge destined to the advertised prefix. It is a heavy-handed approach, actually aiding the attacker in stopping traffic from reaching its destination, but it is an effective method of restoring bandwidth when the attack is targeted to a single IP address or prefix.



Source-based RTBH

Source-based RTBH was standardized in 5635 in 2009. It defines a process where a route with a RTBH community which matches the source address of the attacker is distributed to all edge devices. When the route is received at the edge, an import policy changes the routes next-hop to discard. As the attacker's traffic enters the service provider network, the source address is verified using the uRPF check. Since the best path back to the source address used by the attacker is an internal bit bucket (discard), the uRPF check will fail and the packet will be dropped.

RTBH Issues

- Problems with RTBH
  - Destination based RTBH knocks victim offline
  - Source address of attacker often not known or too numerous to block
  - Source addresses hidden by carrier-grade NAT (blocking legitimate users)
  - Source addresses may be public DNS, NTP, SNMP, or LDAP servers that provide needed services.
- Using just a source or destination address is not granular enough to effectively block just the attack traffic.

© 2020 Juniper Networks, Inc. All Rights Reserved.

RTBH Issues

RTBH has been around since 2004 and has been the main mechanism for DDoS attack mitigation, but it is not without its issues.

Destination-based RTBH takes down the victim's IP address, effectively completing the attack. While this will relieve congestion of the network infrastructure, it is obviously not ideal. This solution may force an emergency IP address change to allow the victim to continue to operate or cause the suspension of business for an indeterminate amount of time.

Source-based RTBH works if the attack is limited to just one or two known addresses where the source address has not been spoofed. This is not the case in most DDoS attacks which source from many different devices. Even if the IP address is known, that particular attacker could be sitting behind a Carrier-Grade NAT connection and blocking that IP address could impact thousands of legitimate users.

The real issue with RTBH is not the technique of blocking traffic, it is that the implementation is not very granular and can only block traffic based on the source or destination IP address. FlowSpec provides a more granular solution and is discussed next.

Agenda: BGP FlowSpec

- DDoS Overview
- DDoS Mitigation Techniques
- →FlowSpec Overview
- Configuring and Monitoring FlowSpec

© 2020 Juniper Networks, Inc. All Rights Reserved.

FlowSpec Overview

The slide highlights the topic we discuss next.

BGP FlowSpec

- BGP FlowSpec
  - Provides a method to remediate DoS or DDoS attacks
  - Defined in industry standard RFC 5575
  - Uses BGP NLRI to advertise a route and protocol details
  - Automatically configures firewall filter or policer
  - Uses BGP AFIs 1 and 2 and SAFIs of 133 and 134

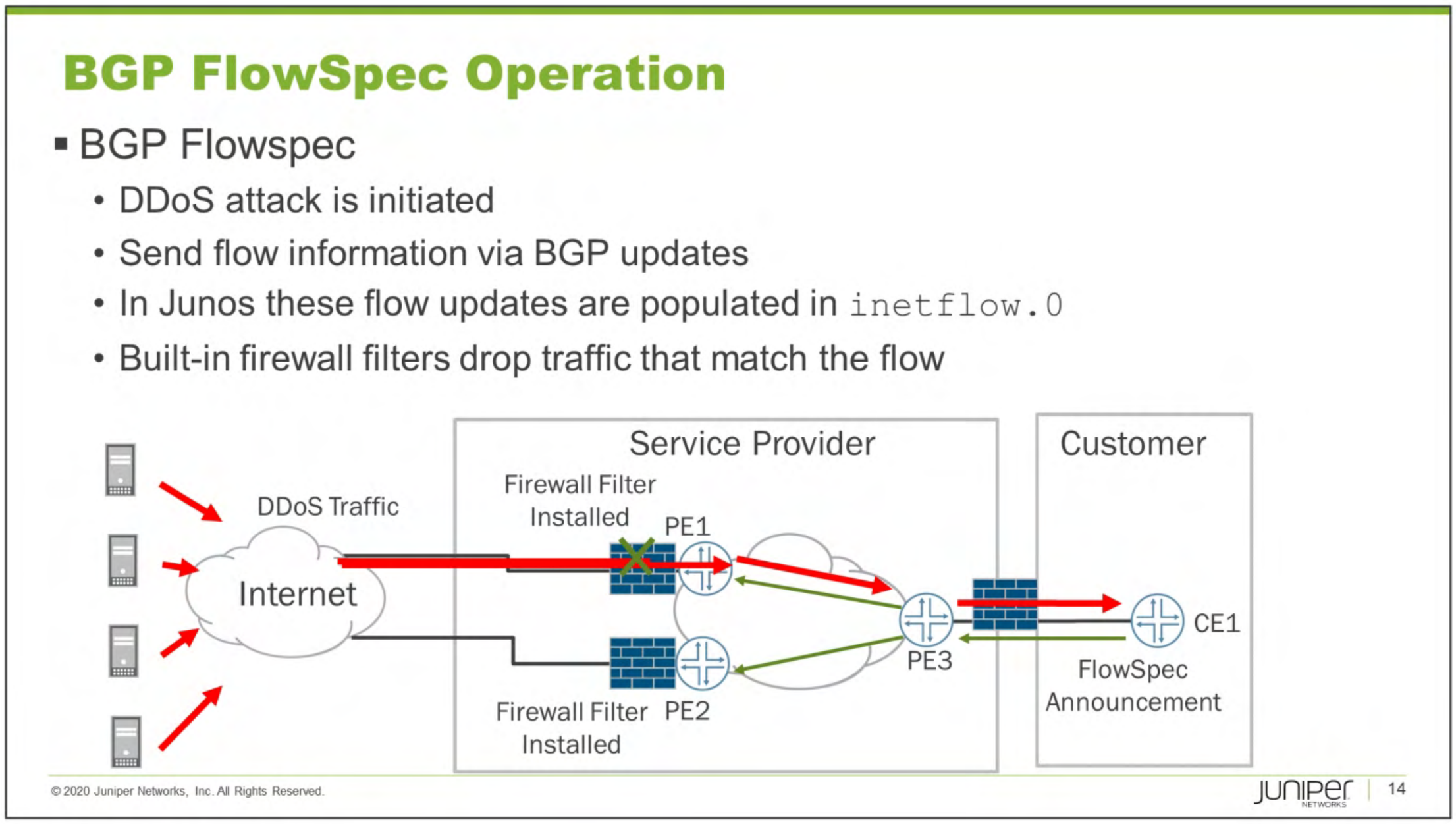
© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP FlowSpec

BGP FlowSpec, defined in RFC 5575, is designed to use BGP to propagate information, enabling edge devices to construct firewall filters to protect a network from denial-of-service attacks dynamically across autonomous systems. This is a powerful method to simplify the alleviation of a common problem.

BGP FlowSpec uses the following AFI/SAFI combinations:

- AFI 1/SAFI 133 – IPv4 dissemination of FlowSpec rules;
- AFI 1/SAFI 134 – VPNv4 dissemination of FlowSpec rules;
- AFI 2/SAFI 133 – IPv6 dissemination of FlowSpec rules; and
- AFI 2/SAFI 134 – VPNv6 dissemination of FlowSpec rules.



BGP FlowSpec Operation

An example of how FlowSpec operates is illustrated on this page.

1. DDoS traffic arrives across the Internet sourced from hundreds of hosts participating around the world. Each is implementing an amplification attack using DNS.
2. The customer's security appliance processes traffic data and determines that packets using the UDP protocol sourced from port 53 and targeted at a specific server comprise the majority of the attack traffic.
3. The customer creates and advertises a FlowSpec route using BGP to the service provider identifying the specific /32 route, the UDP protocol, and port 53. The route carries the desired action which is to discard matching packets.
4. The service provider edge routers receive the FlowSpec information which automatically creates a firewall filter to block the offending traffic.

BGP FlowSpec data can be received for propagation from various sources:

- An internal appliance monitoring traffic and automatically generating FlowSpec routes;
- Manually entered on an internal router;
- From a neighboring (customer) AS BGP speaker; and/or
- Entered through a service provider customer Web Console for collection, verification, and configuration of Flow specification information.

BGP FlowSpec NLRI Types

Type	Information	Type	Information
1	Destination Prefix (destination)	7	ICMP Type (icmp-type)
2	Source Prefix (source)	8	ICMP Code (icmp-code)
3	IP Protocol (protocol)	9	TCP flags (tcp-flags)
4	Source or Destination Port (port)	10	Packet Length (excluding L2 headers) (packet-length- value or range)
5	Destination Port (destination-port)	11	DSCP (match IP ToS octet (dscp)
6	Source Port (source-port)	12	Fragment Encoding -- DF bit, First Fragment, Last Fragment, etc. (fragment)

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP FlowSpec Types

RFC 5575 defines 12 different BGP FlowSpec NLRI component types. These types define the information that can be used to build the firewall filter matching options used by FlowSpec to mitigate an attack. Each NLRI component type is matched with a Junos option that can be used to build a flow specification.

A network operator can use any combination of these NLRI types to specify the attack vector they wish to act upon.

Traffic Filtering Actions

- Traffic Filtering options are extended communities added to a route
- Define the actions to be taken on the traffic

BGP Community	Action	Action Description
0x8006	Traffic-rate	Rate limiter value in bytes per second before dropping traffic. Set to 0 to drop all traffic.
0x8007	Traffic-action	sample traffic to get more info on the attack
0x8008	Redirect	Direct traffic into a VRF
0x8009	Traffic-marking	Change DSCP codes

© 2020 Juniper Networks, Inc. All Rights Reserved.

Traffic Filtering Actions

RFC 5575 also defines various actions, beyond simple filtering, that can be taken on the traffic that matches the FlowSpec route. This information is carried as an extended community added to the route which defines the actions that can be taken on the traffic. The following communities have been defined:

- *Traffic-rate community:* This community tells the receiving router what rate limiter, in bytes per second, to apply to the traffic. If the desire is to drop/discard all the traffic, a rate of 0 should be used.
- *Traffic-action community:* This community is used to sample traffic. This could be useful if the operator wants to sample the attack traffic to get better information about the attack itself.
- *Redirect community:* This community is used to redirect the specified traffic into a virtual routing and forwarding (VRF) instance. It uses the same community that would be used for identifying VPNv4 routes. Thus, the router can import the routes in the same manner for both applications.
- *Traffic-marking community:* This community is used to change the Differentiated Services (DiffServ) code point (DSCP) value in the IP header to the specified value. This could be used by the operator to set the traffic to highest discard probability available in their network. That way they do not drop the traffic until congestion is taking place.

Junos provides options when defining the flow action which will automatically set the correct values for the associated FlowSpec communities.

Traffic Filter Rule Order Processing

- RFC uses a deterministic algorithm
  - Starts by comparing the left-most-components of each NLRI
  - If the types differ, the lowest type by numeric value is used. If they are the same, then the values within that component are compared
  - For IP prefix values (types 1,2, and 3), the lowest IP is chosen. If the IP addresses are the same, the most specific prefix is used.
  - For all other types, the binary string of the contents is compared to determine the order.
- Junos by default does not support the RFC deterministic algorithm
- Best Practice is to have Junos follow the RFC route selection
  - Requires `routing-options flow term-order` set to `standard`

© 2020 Juniper Networks, Inc. All Rights Reserved.

Traffic Filter Rule Order Processing

When processing firewall filters, the common practice is to perform a top-down analysis of terms and evaluate multiple filters in the order they are listed in the filter chain. When a match is made, processing sJunostops. This works fine when firewall filters are designed and implemented using that formula. However, when firewall filters are dynamically added and removed using automation, filter order cannot be guaranteed so this method of evaluation may not result in the desired outcome.

To alleviate the concern, care was taken to define the order in which FlowSpec rules are processed thereby ensuring that conflicts are minimized. The defined processing order is:

- Compare the left-most components of each NLRI.
- If the types differ, the lowest type by numeric value is used. If they are the same, then the values within that component are compared.
- For IP prefix values (types 1, 2, and 3), the lowest IP is chosen. If the IP addresses are the same, the most specific prefix is used.
- For all other types, the binary string of the contents is compared to determine the order.

This is referred to in Junos configuration as a `term-order` of `standard`.

Since Juniper was an early adopter of BGP FlowSpec, its implementation of FlowSpec was released while the specifications for FlowSpec were still in draft form. When configuring Junos, it is considered best practice to configure Junos to use the RFC processing rules. If you do not configure this option, the Juniper default processing method is used.

FlowSpec Validation

- A flow specification received from a BGP peer will need to be validated against the associated routing table before being accepted
- A route is only considered valid if:
  1. The originator of the flow specification matches the originator of the best-match unicast route for the destination prefix embedded in the flow specification
  2. There are no more-specific unicast routes, when compared with the flow destination prefix, that have been received from a different neighboring AS than the best-match unicast route determined in #1.
- By default JUNOS validates using the above rules. This validation can be disabled, and custom policies can used to validate the FlowSpec routes.

© 2020 Juniper Networks, Inc. All Rights Reserved.

FlowSpec Validation

Just because a user or customer submits a FlowSpec specification, doesn't mean it should be accepted without some sort of verification. RFC 5575 defines a set of rules that should be used to validate a received specification before it is implemented. A route is only considered valid if:

1. The originator of the flow specification matches the originator of the best-match unicast route for the destination prefix embedded in the flow specification.
2. There are no more-specific unicast routes, when compared with the flow destination prefix, that have been received from a different neighboring AS than the best-match unicast route determined in #1.

By default, Junos validates any received FlowSpec routes using these rules. The one issue with this rule, is that it will prevent a service provider from implementing a FlowSpec specification unless they originated the most specific route that includes the destination prefix. This is unlikely to be the case if the prefix is originated by the customer. To overcome this issue, validation can be disabled and custom policies can be added to perform the validation.

Agenda: BGP FlowSpec

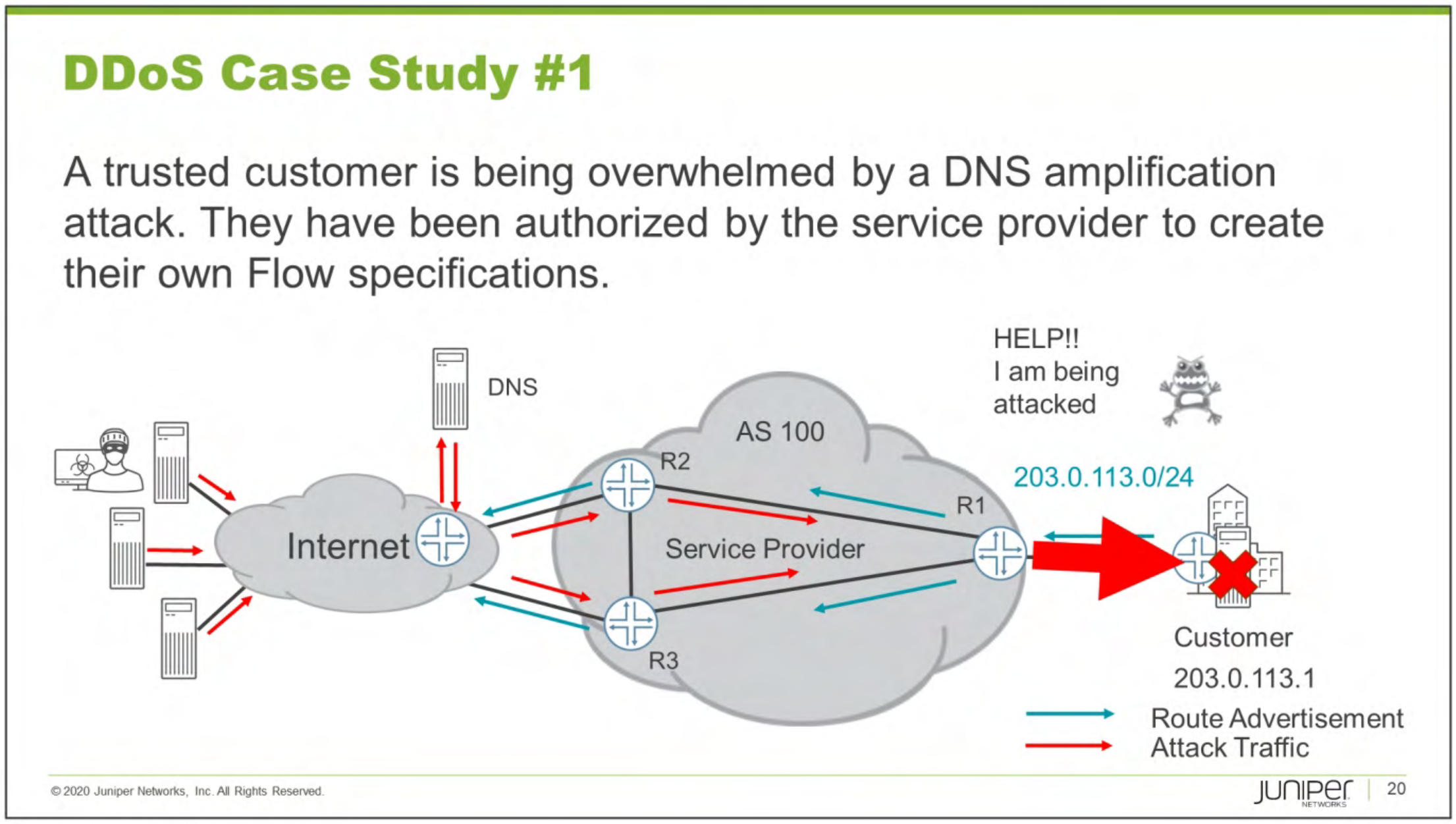
- DDoS Overview
- DDoS Mitigation Techniques
- FlowSpec Overview

→Configuring and Monitoring FlowSpec

© 2020 Juniper Networks, Inc. All Rights Reserved.

Configuring and Monitoring FlowSpec

The slide highlights the topic we discuss next.



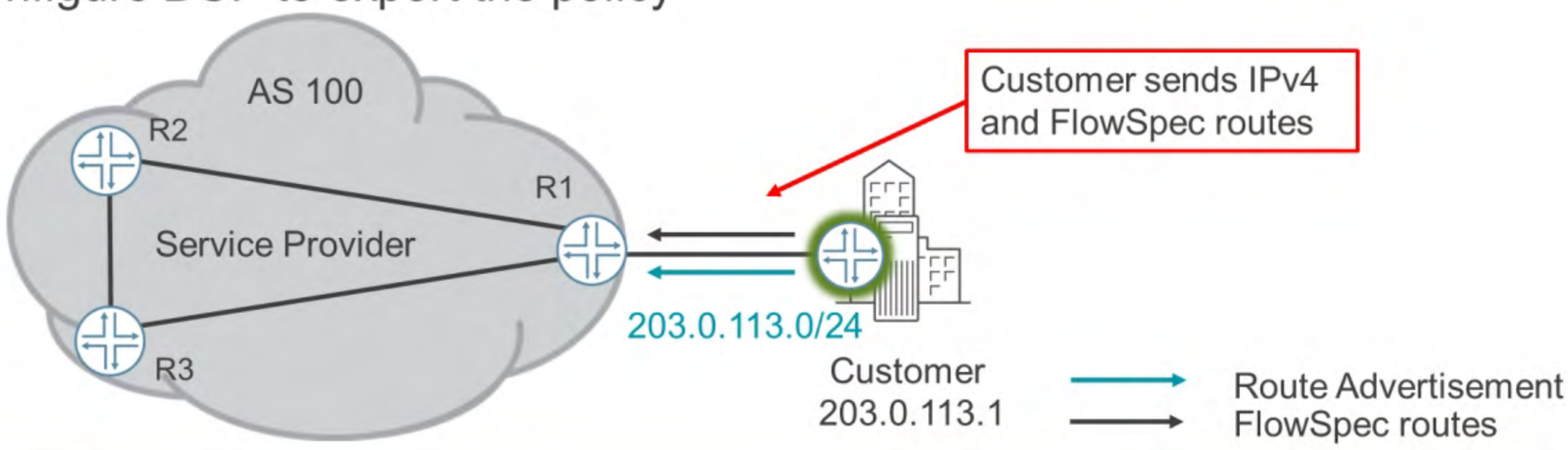
DDoS Case Study #1

In this case study, a service provider has a trusted customer that is being attacked. This customer has been given permission to create FlowSpec rules that the service provider will implement on their PE routers. The customer has determined that they are a victim of a DNS amplification attack directed at their server. They need to take immediate action to block this traffic. The specification they implement will need to drop UDP port 53 packets that are over 99 bytes in size and have a destination address of 203.0.113.1.

Details of how this is accomplished is covered over the next few pages.

DDoS Case Study #1 – Customer Requirements

- To respond to this attack the customer will need:
  - An established `family inet unicast` and `flow` peering with the SP
  - Advertise their internal network prefix to the SP
  - To configure the flow specification under `routing-options`
  - To create a policy to export routes (as needed)
  - To configure BGP to export the policy

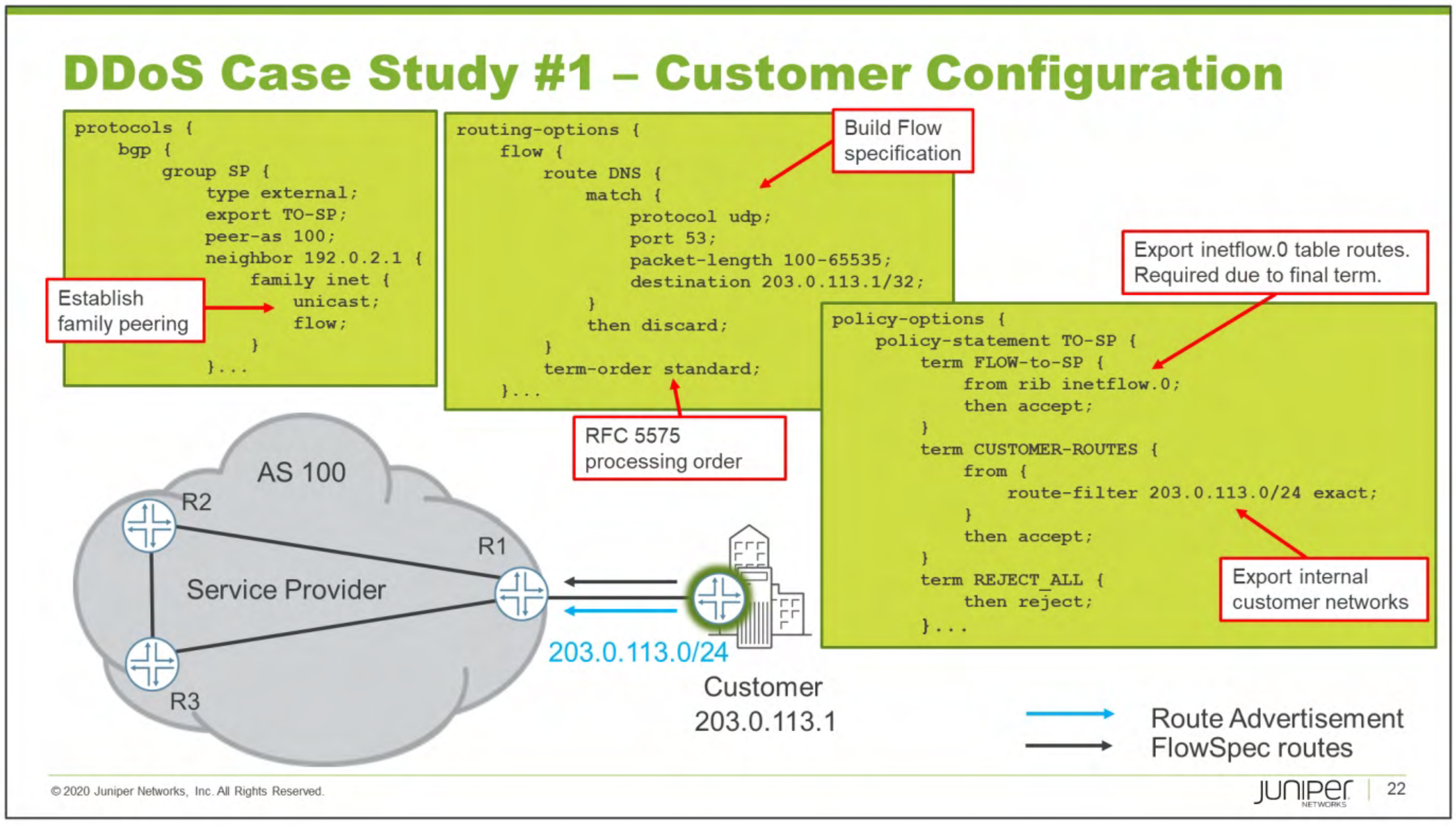


© 2020 Juniper Networks, Inc. All Rights Reserved.

DDoS Remediation—Customer

To respond to this attack, the customer will need the following:

- An established `family inet unicast` and `flow` peering session with the service provider. This provides the ability to exchange routes from both the inet.0 and inetflow.0 tables.
- The local route representing their internal network advertised to the service provider. This is needed to allow the flow specification to be validated by the service provider.
- A flow specification that defines the traffic to be blocked. In this case, the flow specification should include the prefix of 203.0.113.1/32, the protocol of UDP, the port of 53, and an action of drop.
- A policy to advertise the IPv4 cutomer network and possibly the inetflow.0 table.
- BGP will need an export statement to apply the policy.

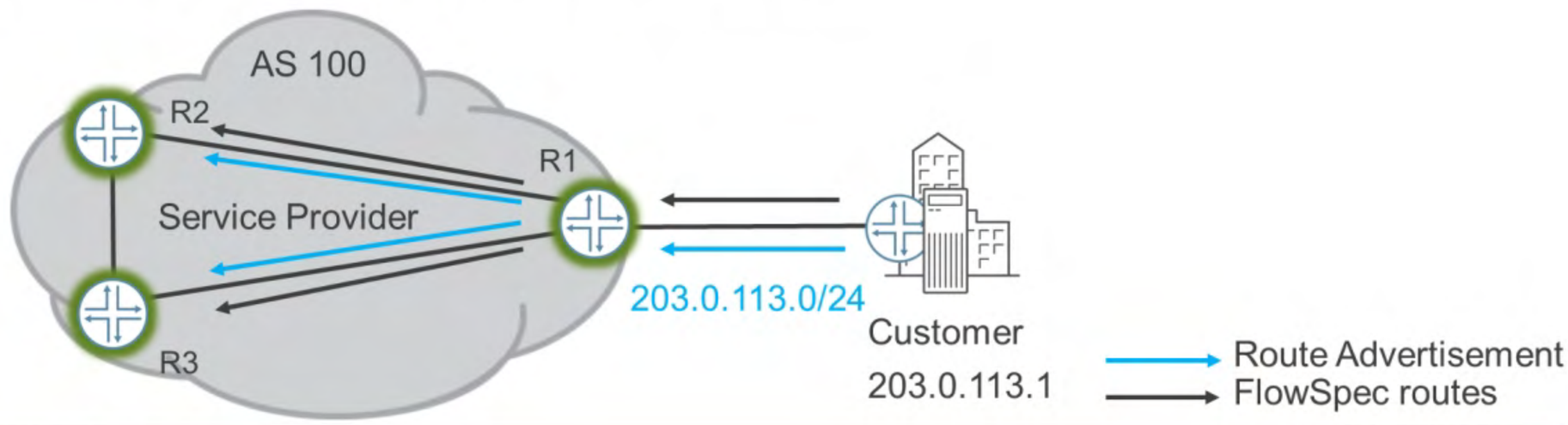


Customer Side Configuration

This slide displays the configurations needed to forward flow specifications to the service provider.

DDoS Case Study #1 – Provider Edge Requirements

- To receive flow specification data from a customer the PE needs:
  - An EBGP `family inet unicast` and `flow` peering with the Customer
  - An IBGP `family inet unicast` and `flow` peering with all other PEs
  - A policy to accept `family inet flow` routes from the customer
  - A policy to send `inetflow.0` routes to PE peers
  - The flow `term-order` option set to `standard`

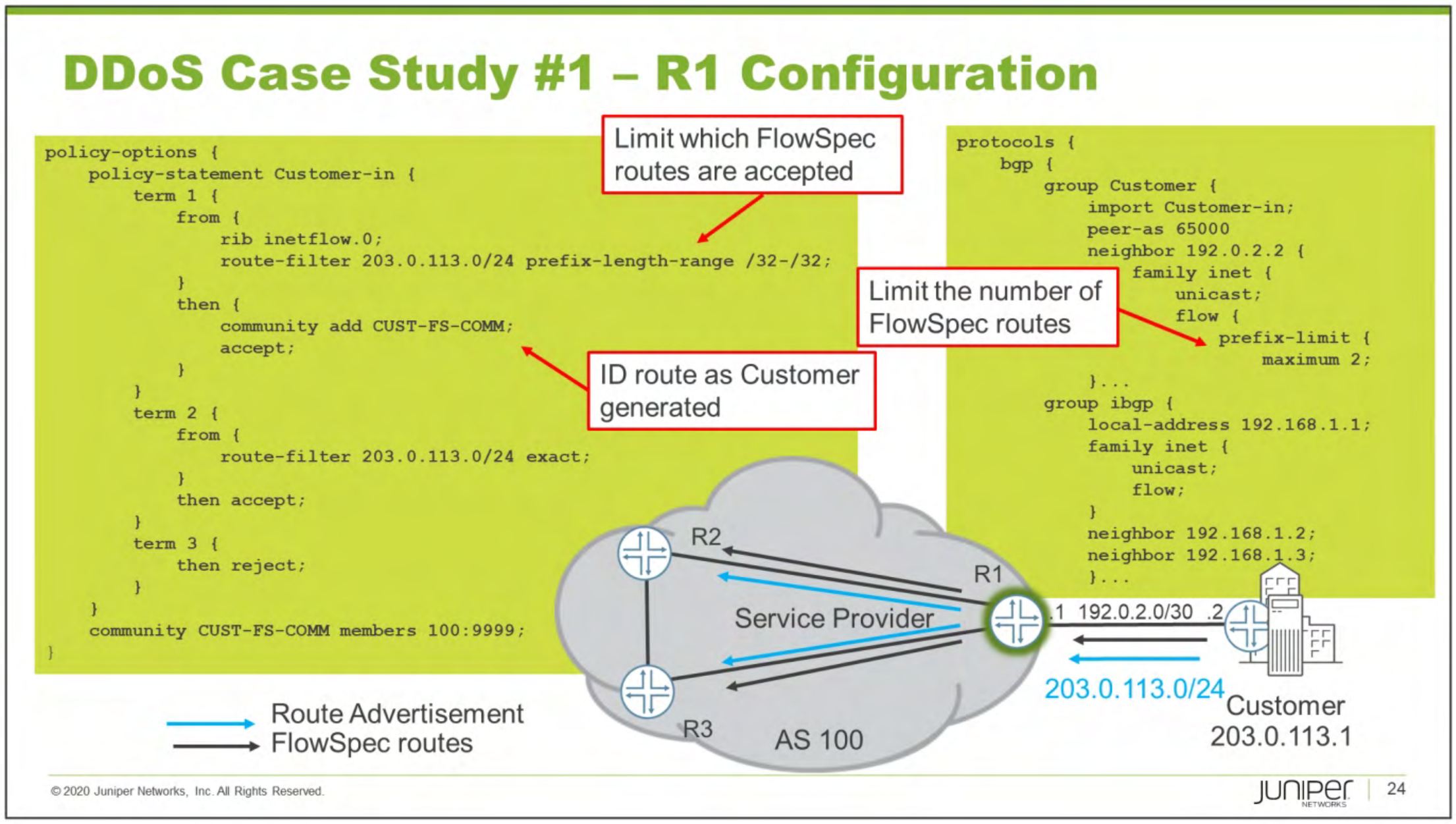


© 2020 Juniper Networks, Inc. All Rights Reserved.

Provider Edge Requirements

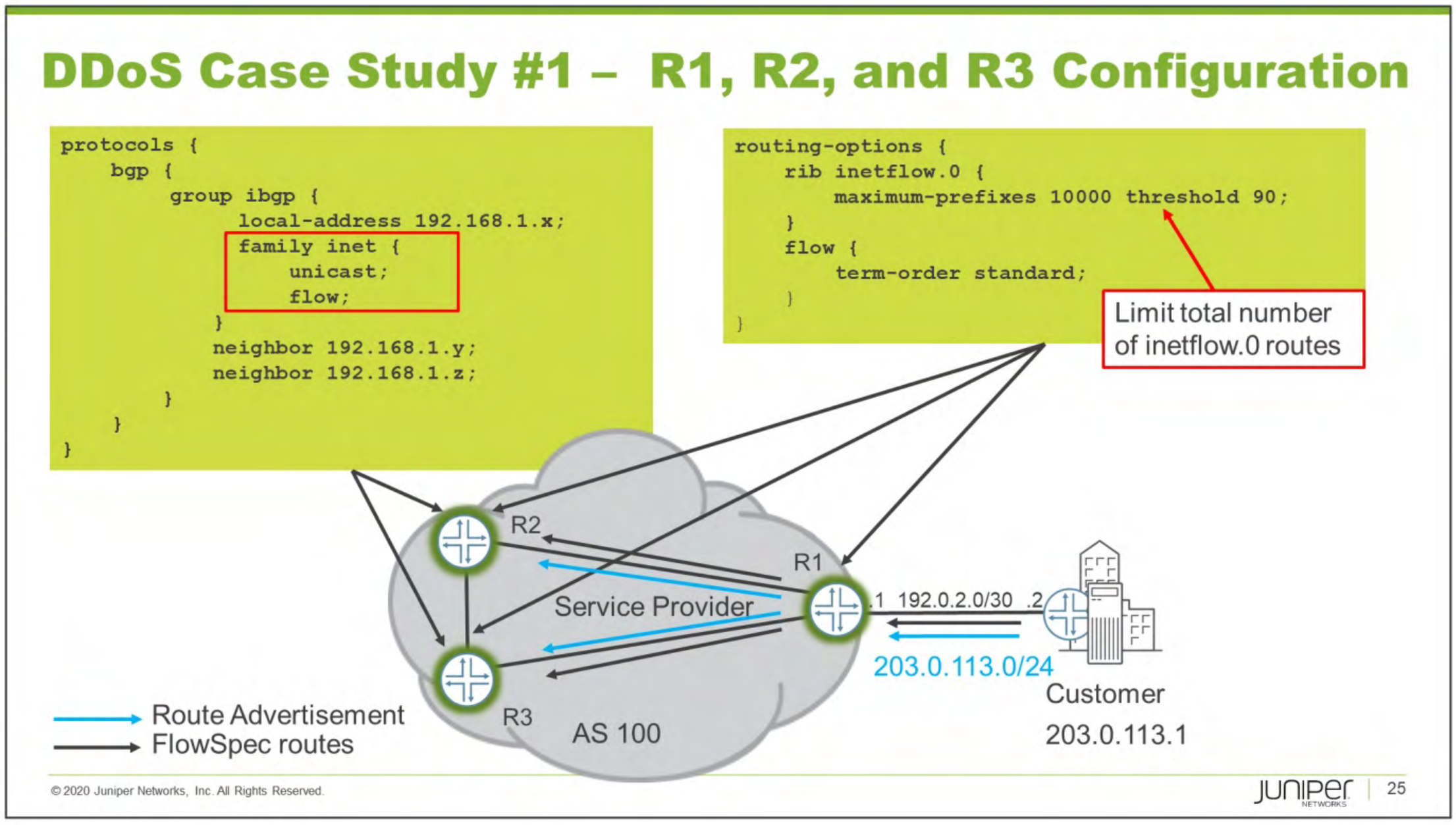
The Edge device that receives the flow specification needs the following:

- An established EBGP `family inet unicast` and `flow` peering with the customer. This will ensure the provider is able to receive the routes the customer controls and the flow specifications created by the customer. The number of flow specification routes accepted from a customer can be limited. Any time the limit is exceed the session will be torn down.
- An established IBGP `family inet unicast` and `flow` peering with the other PE devices in the service provider network. This will allow the edge device to forward the learned routes and flow specifications to its internal peers.
- A policy to accept flow specifications from the Customer. The main concern here is; how many addresses within a single flow specification should the Customer be able to affect? In our case study, the customer will only be allowed to submit /32 routes within the customer's network range. This same policy should also accept the Customer routes to forward to the IBGP peers. Another consideration for the policy would be to add a user-defined community to identify the route as being Customer generated as opposed to service provider generated. In this case study, the service provider will be marking the incoming FlowSpec routes with a community to identify it as customer generated.
- A policy to send the inetflow.0 routes to the IBGP peers. Customer inet.0 routes should be forwarded by the default IBGP forwarding policy to all IBGP peers.
- The flow `term-order` should be set to `standard`. This is considered a best practice. It is under the flow options that a limit can be placed on the total number of routes that can be in the inetflow.0 table. In addition, a percentage value can be configured so that a syslog entry will be created when the number of flow specifications being received is close to its maximum value.



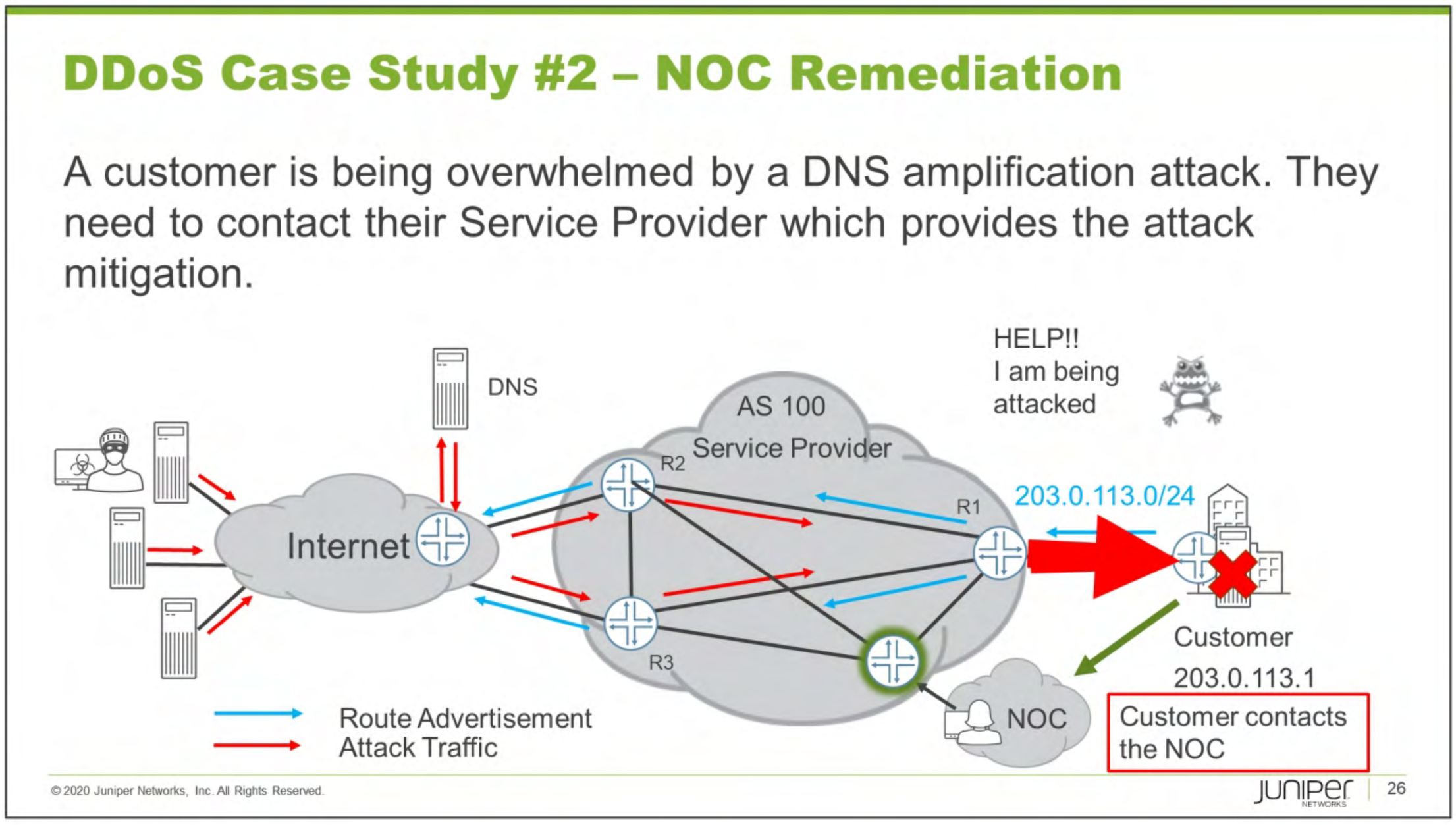
R1 Configuration

This slide depicts the BGP and policy configuration that would be needed on the R1 device to accept and forward flow specifications.



R1, R2, and R3 Configuration

This slide depicts the configuration that would be needed on all PE devices to meet the case study requirements.



DDoS Case Study #2

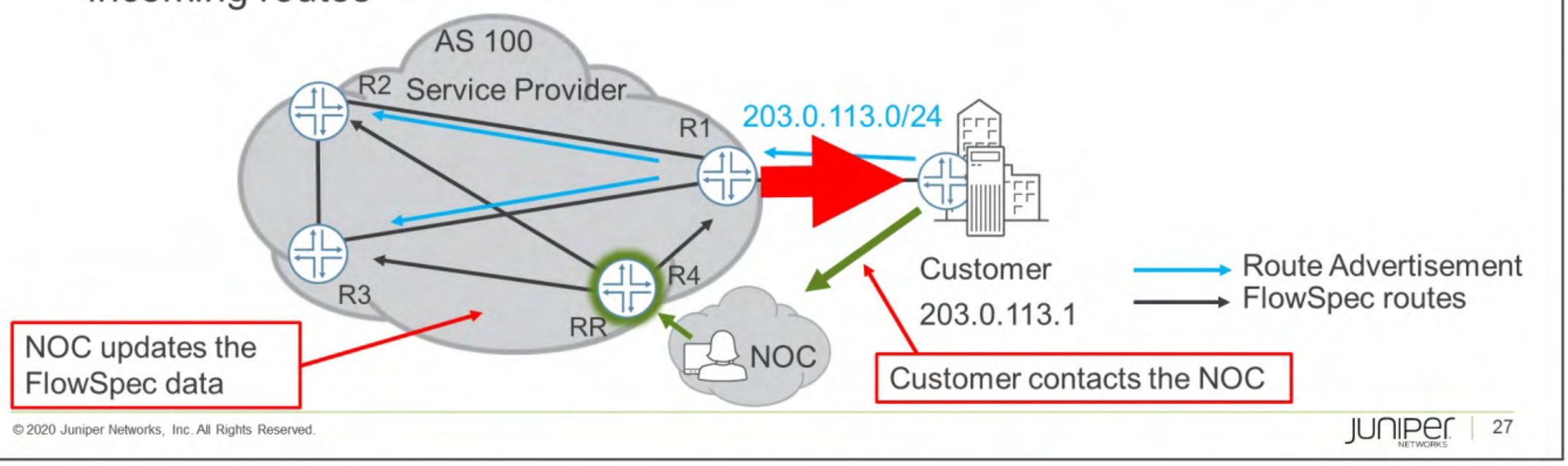
This case study is similar to the previous case study. In this case, the customer is not able to create and forward a specification to the service provider. Instead, when the customer is under attack, they either need to contact the provider via a phone call or access a Web console hosted by the provider to enter attack information from which the provider will create the flow specification data.

This method of entering flow specification breaks FlowSpec's validation rule which says the originator of the FlowSpec prefix must also be the originator of the route within which the prefix resides. This will require some extra configuration to ensure the flow route is accepted.

The case study will be discussed over the next few pages.

DDoS Case Study #2 – RR Requirements

- To implement NOC remediation, the following is needed on the RR:
  - Configure the FlowSpec details under `routing-options` and use the RFC `standard` processing order
  - Configure the RR to peer with all edge PE routers using `family inet flow`
  - Configure a policy to advertise inetflow.0 routes to IBGP peers and reject all incoming routes



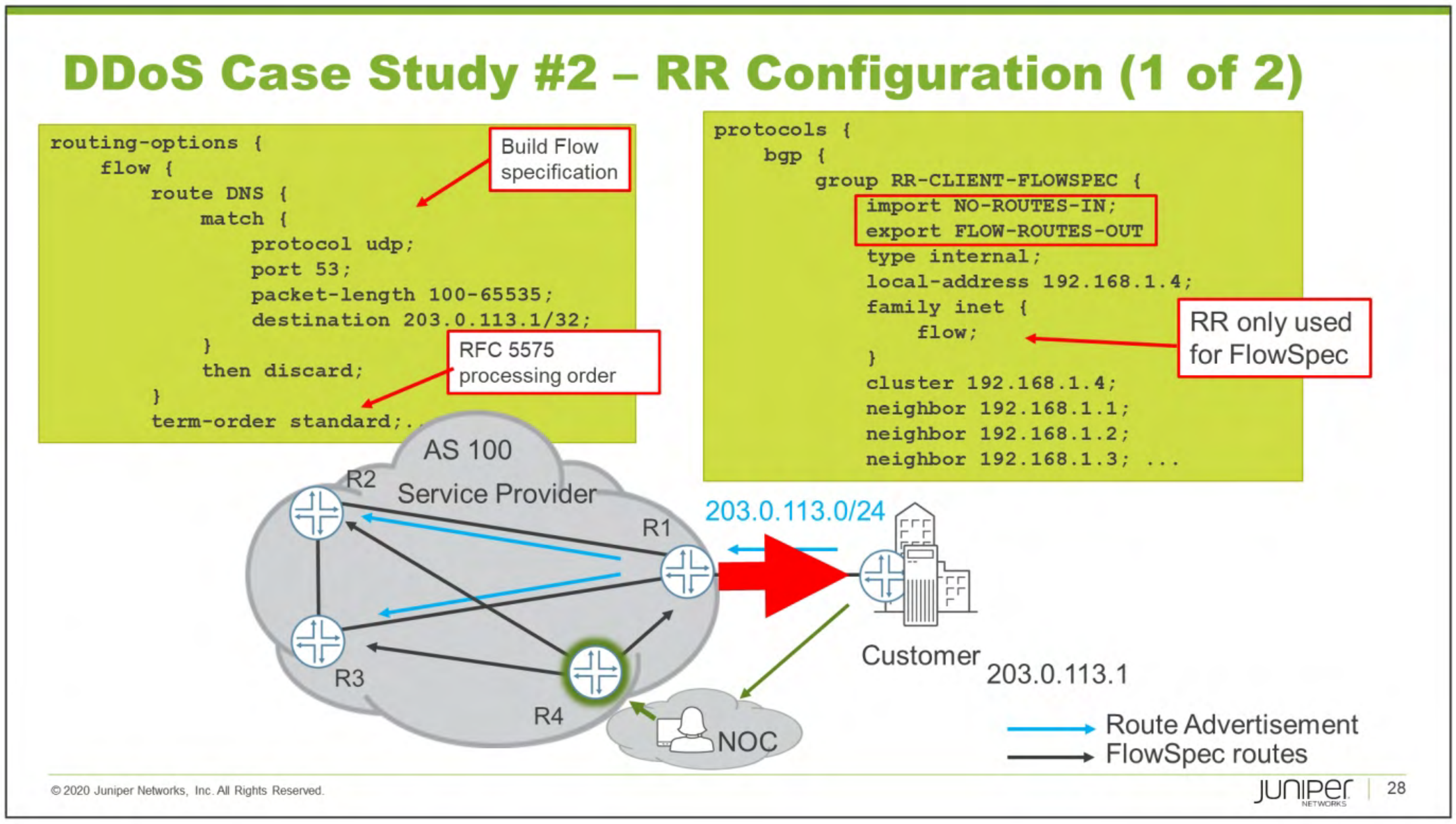
© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Reflector Requirements

In this case study, the most common way a service provider will implement FlowSpec is to use a RR. The specification will be generated on the RR and sent to all the IBGP peers.

To implement NOC remediation, the RR will need the following:

- A flow specification will need to be configured for the DNS attack. As previously mentioned, the flow `term-order` should be set to `standard`.
- The RR needs to peer with all other PE devices using `family inet flow`.
- The RR advertises the inetflow.0 routes to the IBGP peers. Since the RR will be the only source of FlowSpec routes, an import policy will be created to prevent any inetflow.0 routes from being received. The policy can optionally mark the advertised FlowSpec routes with a community to indicate the routes that have been added by the NOC.



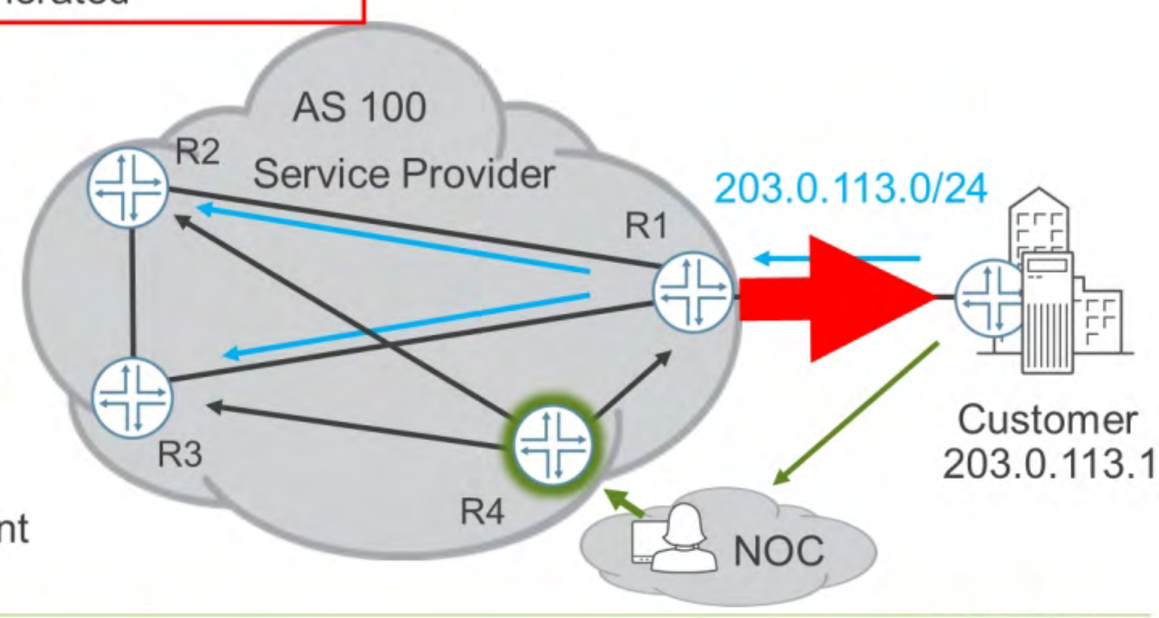
Route Reflector Configuration – Part 1

The slide shows a configuration that implements FlowSpec peerings on the route reflector. The flow route is defined under the `routing-options` hierarchy.

DDoS Case Study #2 – RR Configuration (2 of 2)

```
policy-options {
    policy-statement FLOW-ROUTES-OUT {
        term 1 {
            from rib inetflow.0;
            then {
                community add INTERNAL_FS;
                accept;
        }
        term 2 {
            then reject;
        }
    }
    policy-statement NO-ROUTES-IN {
        term 1 {
            then reject;
        }
    }
    community INTERNAL_FS members 100:1000
...
```

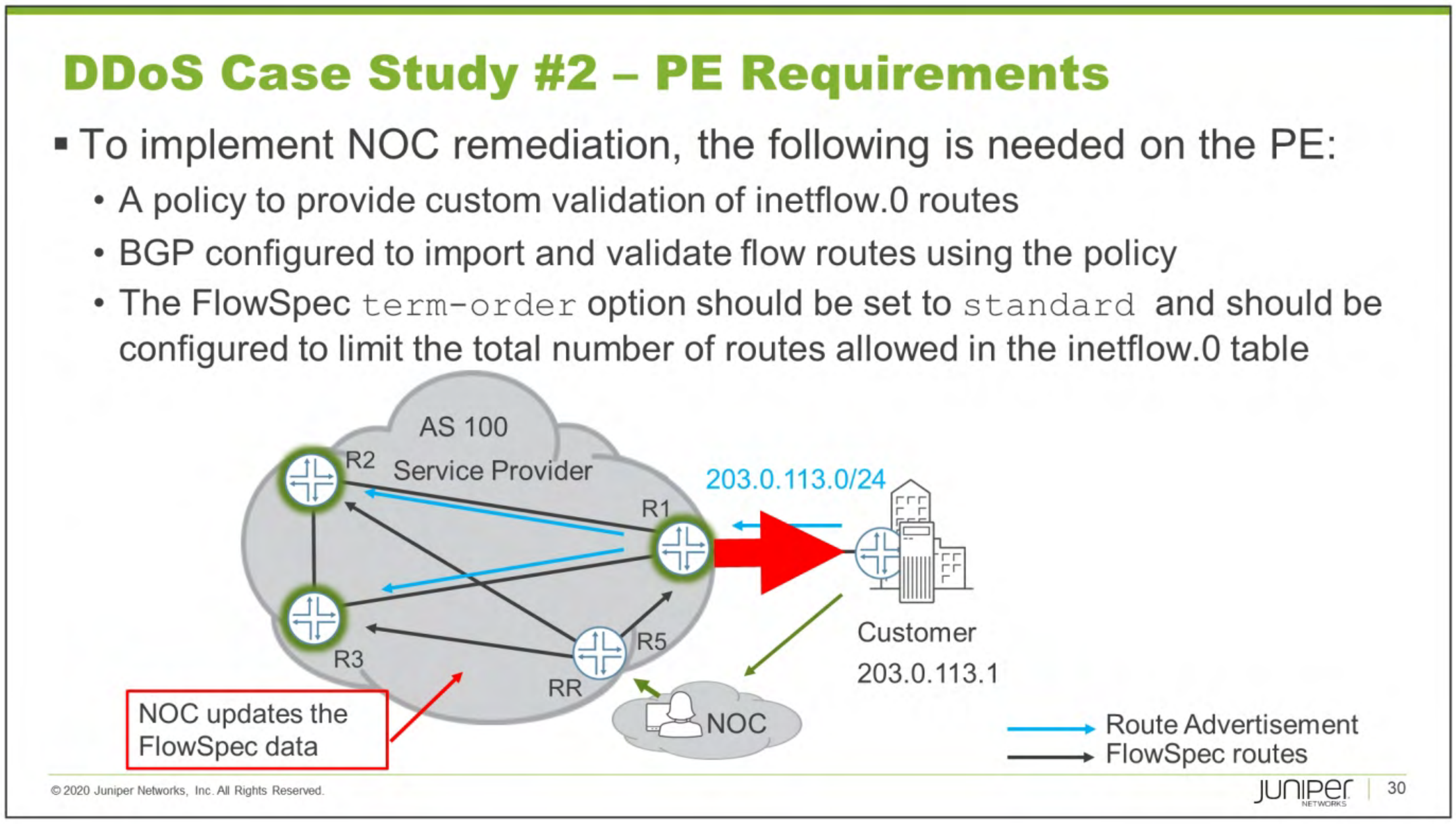
ID route as internally generated



© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Reflection Configuration – Part 2

IBGP is configured for `family inet flow` with an import policy (NO-ROUTES-IN) to reject all routes and an export policy (FLOW-ROUTES-OUT) to advertise routes that are added to the inetflow.0 route table.



PE Requirements

To ensure the PE devices are able to accept the flow specifications, the following is needed:

- A policy to accept specific inetflow.0 routes that are being sent.
- BGP configured to import the inetflow.0 routes based on the policy. Since the incoming routes are not being advertised by the originator of the network prefix, FlowSpec route verification will need to be disabled and a new policy used to validate the route.
- Finally, the flow specification should be configured to use the `standard term-order` which is considered best practice.

Configuration examples follow on the next pages.

DDoS Case Study #2 PE Configuration

```
policy-options {
    policy-statement FLOWSPEC-RR-IN {
        term 1 {
            from {
                rib inetflow.0;
                route-filter 0.0.0.0/0 prefix-length-range /32-/32;
            }
            then accept;
        }
        term 2 {
            then reject;
        }...
```

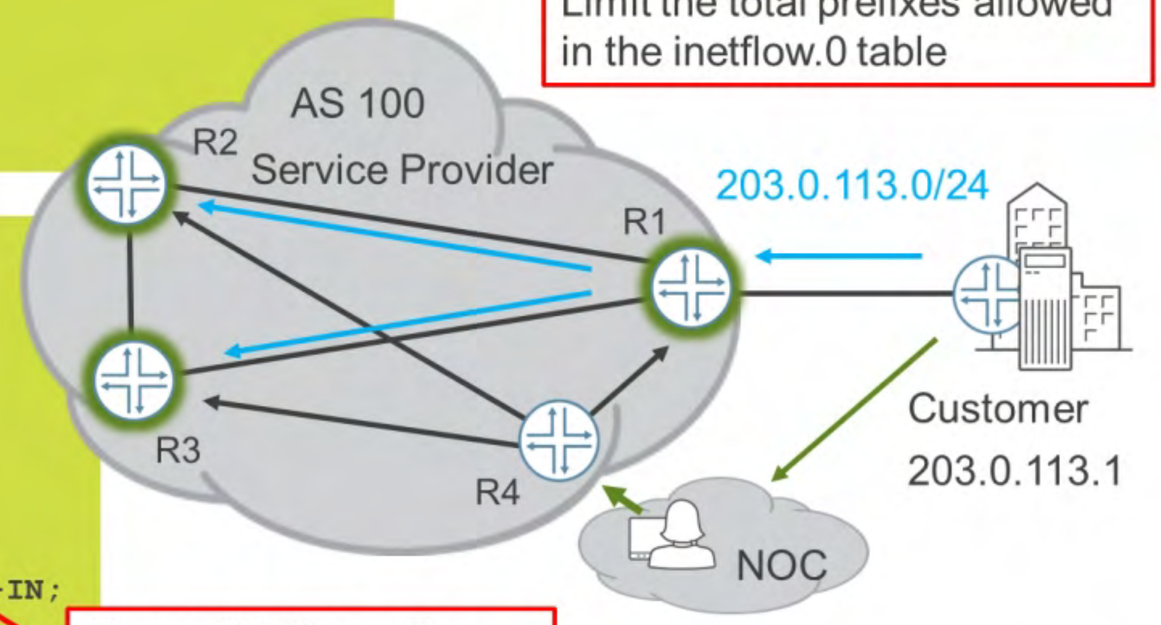
Limit FlowSpec routes accepted to /32s only

```
routing-options {
    rib inetflow.0 {
        maximum-prefixes 1000;
    }
    flow {
        term-order standard;
```

Limit the total prefixes allowed in the inetflow.0 table

```
protocols {
    bgp {
        group RR_CLIENT-FLOWSPEC {
            import FLOWSPEC-RR-IN
            type internal;
            local-address 192.168.1.x;
            neighbor 192.168.1.4 {
                family inet {
                    flow {
                        no-validate FLOWSPEC-RR-IN;
                    }...
```

New validation policy



© 2020 Juniper Networks, Inc. All Rights Reserved.

R1, R2, and R3 Example Configurations

The slide shows the policy that can be used to accept and validate the flow routes. In this example, the routes accepted are limited to /32s. While this is not an absolute requirement for flow specifications, it does prevent the accidental inclusion of a large block of addresses that could be detrimental if a malformed prefix is configured.

View and Verify BGP Peerings

- Verifing FlowSpec is configured between BGP peers

```
lab@mxA> show bgp summary
Threading mode: BGP I/O
Groups: 2 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                       1          1          0          0          0          0
inetflow.0
                       1          1          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
172.22.131.37         65000       2199       2182       0       5    16:22:28 Establ
  inet.0: 1/1/1/0
  inetflow.0: 1/1/1/0
192.168.71.2          64512          7          7       0       0        2:00 Establ
  inet.0: 0/0/0/0
```

FlowSpec peering established

No FlowSpec peering established

© 2020 Juniper Networks, Inc. All Rights Reserved.

Verifying BGP Flow Peering

The `show bgp summary` command is useful to verify the current state of the BGP peering. From here you can validate if the Established state has been achieved and what address families have been negotiated between both sides.

In the example above, the 172.22.131.37 peer has achieved both family inet unicast and family inet flow peering. This is visible from the route tables associated with the peering display. In addition, one IPv4 route and one Flow-route has been received from the peer.

The 192.168.71.2 peering only has achieved an IPv4 peering. If Flowspec was to be a peering option, then it was not successful for this peering. A check of the address families configured on both sides should reveal the issue.

View and Verify Flow Routes

- View inetflow.0 table entries

```
lab@router> show route table inetflow.0 extensive

inetflow.0: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
203.0.113.1,*,proto=17,port=53/term:1 (1 entry, 1 announced)
TSI:
KRT in dfwd;
Action(s): discard,count
Page 0 idx 0, (group SP type External) Type 1 val 0xd7e7eb0 (adv_entry)
   Advertised metrics:
     Flags: NoNexthop
     AS path: [65000] I
     Communities: traffic-rate:65000:0
Path 203.0.113.1,*,proto=17,port=53
Vector len 4.  Val: 0
        *Flow    Preference: 5
                 Next hop type: Fictitious, Next hop index: 0
                 Address: 0xc0f1528
                 Next-hop reference count: 12
                 Next hop:
                 State: <Active SendNhToPFE>
                 Local AS: 65000
                 Age: 3:54
                 Validation State: unverified
                 Task: RT Flow
                 Announcement bits (2): 0-Flow 1-BGP_RT_Background
                 AS path: I
                 Communities: traffic-rate:65000:0
```

FlowSpec route

FlowSpec action

Flow created community-string value of 0 to drop the traffic

Next hop set to Fictitious as there is no next-hop

© 2020 Juniper Networks, Inc. All Rights Reserved.

Verifying Flow Routes

To view and verify flow routes, use the `show route table inetflow.0 extensive` command. Besides viewing the details of the flow specification, the actions taken are visible. The community added to the flow reflects the traffic rate option used to police or drop the traffic. In the example on the slide, the value of 0 indicates that the traffic should be dropped.

The Fictitious attribute indicates there is no BGP-NH in the route as traffic forwarding is not the purpose of this route.

Hidden Flow-routes

- Example of a Flow-route that failed FlowSpec validation

```
lab@router> show route table inetflow.0 extensive hidden

inetflow.0: 1 destinations, 1 routes (0 active, 0 holddown, 1 hidden)
203.0.113.1,*,proto=17,port=53/term:N/A (1 entry, 0 announced)
        BGP                 /-101
                Next hop type: Fictitious, Next hop index: 0
                Address: 0xc0f1528
                Next-hop reference count: 1
                Next hop:
                State: <Hidden Ext SendNhToPFE>
                Local AS: 64512 Peer AS: 65000
                Age: 1:53
                Validation State: unverified
                Task: BGP_65000.172.22.131.37
                AS path: 65000 I
                Communities: traffic-rate:65000:0
                Validation state: Not resolved, Originator: 172.22.131.37, Nbr AS: 65000
                Localpref: 100
                Router ID: 172.22.131.37
                Hidden reason: Flow-route fails validation
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Hidden Flow-routes

The flow routes are hidden if the route fails validation. The rules of flow route validation are:

- The originator of the flow specification matches the originator of the best-match unicast route for the destination prefix embedded in the flow specification.
- There are no more-specific unicast routes, when compared with the flow destination prefix, that have been received from a different neighboring AS than the best-match unicast route determined in #1.

In this case, the originator did not advertise a route that encompassed the FlowSpec destination address.

View Flow Validation

- Flows must be validated before conversion to firewall filters

```
lab@router> show route flow validation detail

inet.0:
203.0.113.1/32
                Flow destination (1 entries, 0 match origin, next-as)
```

Validation Fails

```
lab@router> show route flow validation detail

inet.0:
203.0.113.0/24
                Internal node: no match, consistent, next-as: 65000
                Active unicast route
                    Dependent flow destinations: 1
                    Origin: 172.22.131.37, Neighbor AS: 65000
203.0.113.1/32
                Flow destination (1 entries, 1 match origin, next-as)
                    Unicast best match: 203.0.113.0/24
                    Flags: Consistent
```

Validation Succeeds

© 2020 Juniper Networks, Inc. All Rights Reserved.

View Flow Validation

The `show route flow validation detail` command can be used to see if the Flow-route was verified using the default rules. If the route fails the validation or you have configured custom verification rules, the command results will display the Flow-route with 0 matches.

If the Flow-route is validated by the default rules, then both the Flow-route and the validated route will display.

Verify Firewall Filters

- Firewall filter automatically created

```
lab@mxC-R3> show firewall

Filter: __default_bpdu_filter__

Filter: __flowspec_default_inet__
Counters:
Name                                                Bytes              Packets
203.0.113.1,*,icmp-type=8,len>=128&<=65535            128                    1
203.0.113.1,*,proto=17,port=53                         52                    1
```

NOTE: Current implementation applies the Firewall Filter to all interfaces

© 2020 Juniper Networks, Inc. All Rights Reserved.

Verify Firewall Filters

A firewall filter will automatically be created on every device that has a Flow-route in its inetflow.0 table. The firewall filter and its associated counters can be viewed by using the `show firewall filter __flowspec_default_inet__` command. This firewall filter is automatically applied to all interfaces.

This command can be used to verify if any matches are found by the firewall filter. These counters will remain until cleared using the `clear firewall counter <counter-name> filter <filter-name>` command.

Summary

- In this content, we:
  - Described DDoS attacks
  - Described DDoS mitigation techniques
  - Described FlowSpec Operations
  - Configured and Monitored FlowSpec

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- DDoS attacks;
- DDoS mitigation techniques;
- FlowSpec Operations; and
- How to configure and monitor FlowSpec.

Review Questions

1. What address family is required to implement FlowSpec?
2. Under which Junos hierarchy is the attack definition configured that FlowSpec will use to block traffic?
3. Which processing order is recommended when implementing Junos FlowSpec?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

3.

Lab: BGP FlowSpec

- Configure and monitor FlowSpec

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: BGP FlowSpec

The slide provides the objective for this lab.

Answers to Review Questions

1.

The address family for FlowSpec is `family inet flow`.

2.

The Junos hierarchy where the attack definition is configured is under `routing-options flow route <name>`.

3.

The recommended FlowSpec term processing order is `standard`.

Education Services

Advanced Junos Service Provider Routing

Chapter 15: Troubleshooting BGP

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - Review common BGP troubleshooting procedures
  - List common BGP troubleshooting commands
  - Identify issues with BGP peering

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- Review common BGP troubleshooting procedures;
- List common BGP troubleshooting commands; and
- Identify issues with BGP peering.

Agenda: Troubleshooting BGP

→BGP Troubleshooting

- Case Study: Neighborship Issues

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Troubleshooting

The slide lists the topics we will discuss. We discuss the highlighted topic first.

BGP Overview

- BGP is the routing protocol of the Internet
- BGP reliably exchanges routes with defined peers using TCP port 179
- A full internet download consists of over 800,000 routes just for the IPv4 address family
- BGP supports many address families
  - IPv4, IPv6, L2vpns, L3vpns, EVPN, MVPN, FlowSpec etc.
- BGP supports 2 methods of route redistribution
  - IBGP
  - EBGP
- More routes of more types using multiple methods of delivery creates more potential for problems

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Overview

BGP is the protocol responsible for routing across the Internet. It reliably exchanges routes with defined peers using TCP port 179. BGP is very extensible, and can be adapted to transport different types of prefixes for different applications.

At the time of this writing, a full Internet download of IPv4-based routes is over 800,000 routes. The number of IPv6 routes is over 70,000 and growing. Not only is BGP used to transport IPv4 and IPv6 routes, BGP now carries information used by many different routing applications. These include Layer 2 VPNs, Layer 3 VPNs, EVPNs, MVPNs, and Flowspec.

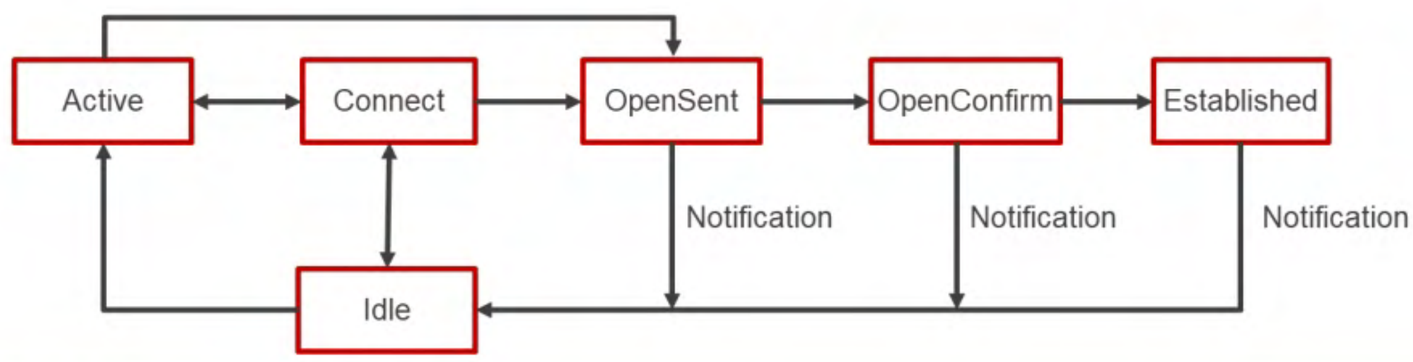
Each of these various applications embed their information in the Network Layer Reachability Information (NLRI) field of the BGP update message. These messages are exchanged between peers using either IBGP or EBGP. EBGP and IBGP have different rules for forwarding traffic which must also be managed.

BGP has to perform many roles to support all of these applications and as such there are more places where BGP issues can occur. This makes troubleshooting more complex than other protocols.

This module will help you understand some of the tools that can be used to troubleshoot this powerful protocol.

Initial BGP Troubleshooting

- Ensure BGP protocol traffic is being allowed into the local device
  - Firewall Filters
  - Security Policies (host-inbound-traffic)
- Watch for BGP TCP session issues
  - BGP messages use TCP transport
  - Often BGP peering problems occur because the TCP connection fails



© 2020 Juniper Networks, Inc. All Rights Reserved.

Initial BGP Troubleshooting

BGP troubleshooting generally falls into one of two categories:

- Peering session establishment troubleshooting
- Routing Troubleshooting

In order for BGP to attempt to form an adjacency, BGP messages must be able to reach its peer. If firewall filters or security policies are in place when configuring BGP, they could interfere with the peering process. It is important to ensure firewall filters allow TCP port 179. If your BGP router is also a security device then don't forget the `host-inbound-traffic` configuration requirement of the SRX device.

TCP Session Establishment

You must establish a TCP connection to exchange traffic using BGP routers. The possible states for the TCP connections are as follows:

- *Idle:* The initial neighbor state, BGP waits for start event;
- *Connect:* BGP waits for TCP connection to be completed; and
- *Active:* TCP timeout has occurred and the TCP session is not established yet. System continues to listen for completion of TCP session.

When the BGP state field in the **show bgp summary** command is `Idle`, it indicates that the outgoing interface may be down or there is no route in the routing table to forward a TCP SYN message.

When the BGP state field in the **show bgp summary** command is `Connect`, it indicates that TCP messages are being sent but no response has been received. This can be caused by a firewall filter blocking the BGP port or a misconfigured or nonexistent neighbor.

When the BGP state field in the **show bgp summary** command is `Active`, it can indicate that the BGP session is having an issue. Normal issues that can cause this state are an incorrect AS number or authentication issue.

IBGP Peering Issues

- IBGP peering session establishment issues
  - Misconfigured peer IP address
  - Missing local source IP address
  - No route to neighbor
  - Large TCP MSS that causes BGP packet fragmentation and a firewall filter dropping the fragments
  - Passive neighbor setting at both ends
  - Authentication parameters mismatch

© 2020 Juniper Networks, Inc. All Rights Reserved.

IBGP Peering Troubleshooting

Several reasons can explain why an IBGP session does not come up. Mostly they are related to TCP transport issues. For example, if you misconfigure an IBGP peer address your router will not be able to deliver the BGP messages to the remote peer.

IBGP usually uses indirect peering using the routers' loopback interfaces. Hence, BGP relies on the underlying IGP protocol for the remote peer loopback reachability. If the loopback route is missing in the IGP, the TCP connection and BGP session will fail to establish.

When defining a BGP loopback peering session, you need to correctly match the source address used by the local peer to ensure that it matches the session definition at the remote peer. By default, the router will source traffic from its egress interface, which will not be the loopback interface, and this can make the incoming connection request appear to come from an undefined peer. Note that the Junos OS is able to successfully establish a BGP session even if you have one of the two peers configured without the `local-address` setting but generally you should avoid these situations.

A good way to test IP reachability between the BGP peers is to ping the loopback of the peer, sourcing the ping from the local loopback. This tests both the ability to reach the remote peer and its ability to route traffic back to the source device. It is not enough to have a route to the neighbor, but the neighbor must be able to reach the originating BGP peer as well.

You can configure a BGP neighbor with a passive setting. This configuration will cause the local BGP speaker to not try to initiate the BGP TCP session but instead wait for its peer to send the first TCP syn packet. If the two peers are both configured with the passive setting the session will not come up.

By default, a BGP TCP session transmits 576 bytes in a single packet to minimize the chances that the packet will be fragmented at a device along the path to the destination. You can configure the TCP maximum segment size (MSS) to a higher value in an attempt to optimize delivery of large BGP packets, typically BGP Update messages. In this scenario if a link on the path of the BGP messages has an MTU that cannot handle the large TCP segment, the packet will be fragmented. An issue might arise if you have a firewall filter rejecting TCP fragments somewhere on the path.

One workaround is to enable path MTU discovery in the BGP configuration. Path MTU discovery, which is disabled by default in the Junos OS, allows BGP to dynamically determine how large the packets can be in a TCP session without being fragmented. Note however, that enabling TCP path MTU discovery might create an ICMP vulnerability. You can configure BGP authentication to guarantee that only trusted routers participate in the AS's routing. The mismatched authentication parameters will prevent the BGP session establishment. Note that authentication data is included in TCP segments and processed by the Junos kernel. The mismatched authentication is reported by the kernel in the system log file.

EBGP Peering Issues

- EBGP peering session establishment issues
  - Misconfigured peer IP address
  - Misconfigured peer or local AS number
  - MTU mismatch
  - One hop (default) or multiple hops
  - Passive setting at both ends
  - Authentication parameters mismatch

© 2020 Juniper Networks, Inc. All Rights Reserved.

EBGP Peering Troubleshooting

EBGP normally peers to a neighbor using the addresses configured on the directly connected link between the routers. A misconfigured peer address will prevent a router from establishing a session.

If you configure a multihop EBGP session there are three extra configuration steps needed:

1. Since each router will establish the peering session with the other's loopback address, you have to use the `local-address` option, so the BGP open message will be sourced appropriately.
2. Use the `multihop` BGP option to alter the default TTL of 1 on EBGP sessions.
3. Each router must have IP routing capability to the remote router's loopback address. This normally requires the configuration of a static route.

The failure to configure any of these parameters results in a BGP session that does not come up.

A typical BGP protocol specific session failure occurs if the neighboring autonomous system number does not match the configuration at the remote peer. BGP detects this problem when it starts exchanging BGP open messages. The BGP speaker then sends the BGP notification message to its peer reporting the issue.

With directly connected EBGP sessions, TCP uses MTU-sized packets. If there is an MTU mismatch between the two sides of the TCP connection, the BGP session can be dropped if a large BGP packet, typically a BGP update message, does not fit the MTU size. This failure typically occurs in multi-vendor environments where different vendors might handle the interface MTU differently.

A mismatched authentication parameter will prevent BGP session establishment as will the configuration of the `passive` option on both connected peers.

Routing Issues

- BGP routing issues
  - Received routes
    - BGP next hop is not reachable
    - Import filtering policy
    - Prefix limit is reached
    - Recursive routing failure
  - Advertised routes
    - Export filtering policy
    - Configured aggregation with missing contributing routes
    - Inactive routes
    - Well-known route advertisement controlling communities are attached to a route

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Routing Troubleshooting

Once BGP peering sessions are established successfully BGP neighbors exchange BGP update messages advertising NLRIs. In the process of troubleshooting BGP routing you generally monitor the routing updates flowing in either receive or send direction.

When a BGP router receives a BGP update it first checks that it can reach the BGP next hop. It uses a recursive route lookup to resolve the loopback peering address to an IP forwarding next hop. This is not an issue in a typical EBGP session using the physical interface address for the peering session. It might become an issue in an IBGP environment or on the multihop EBGP sessions.

The BGP default import policy accepts all BGP received routes that pass the sanity checks. If an incorrect filtering import policy is applied you may end up filtering good routes. Verify which routes have been filtered by viewing hidden routes.

You can set a limit on the amount of prefixes received from a peer. The prefix limit configuration provides an option of tearing the session down once the limit is reached along with the holddown timer. If the received prefixes hit the limit, the router ceases the session and sends a BGP notification message. The router then waits for the specified period before it attempts to re-establish the BGP session.

The recursive routing failure occurs if a BGP router receives a BGP route and the recursive lookup of this route's BGP next hop results in using the route itself. This failure would result in a BGP session failure and constant deletion and reinsertion of BGP routes into the routing table. The Junos OS performs a sanity check for this kind of failure and if it is detected, it marks the offending route as hidden, and ignores it.

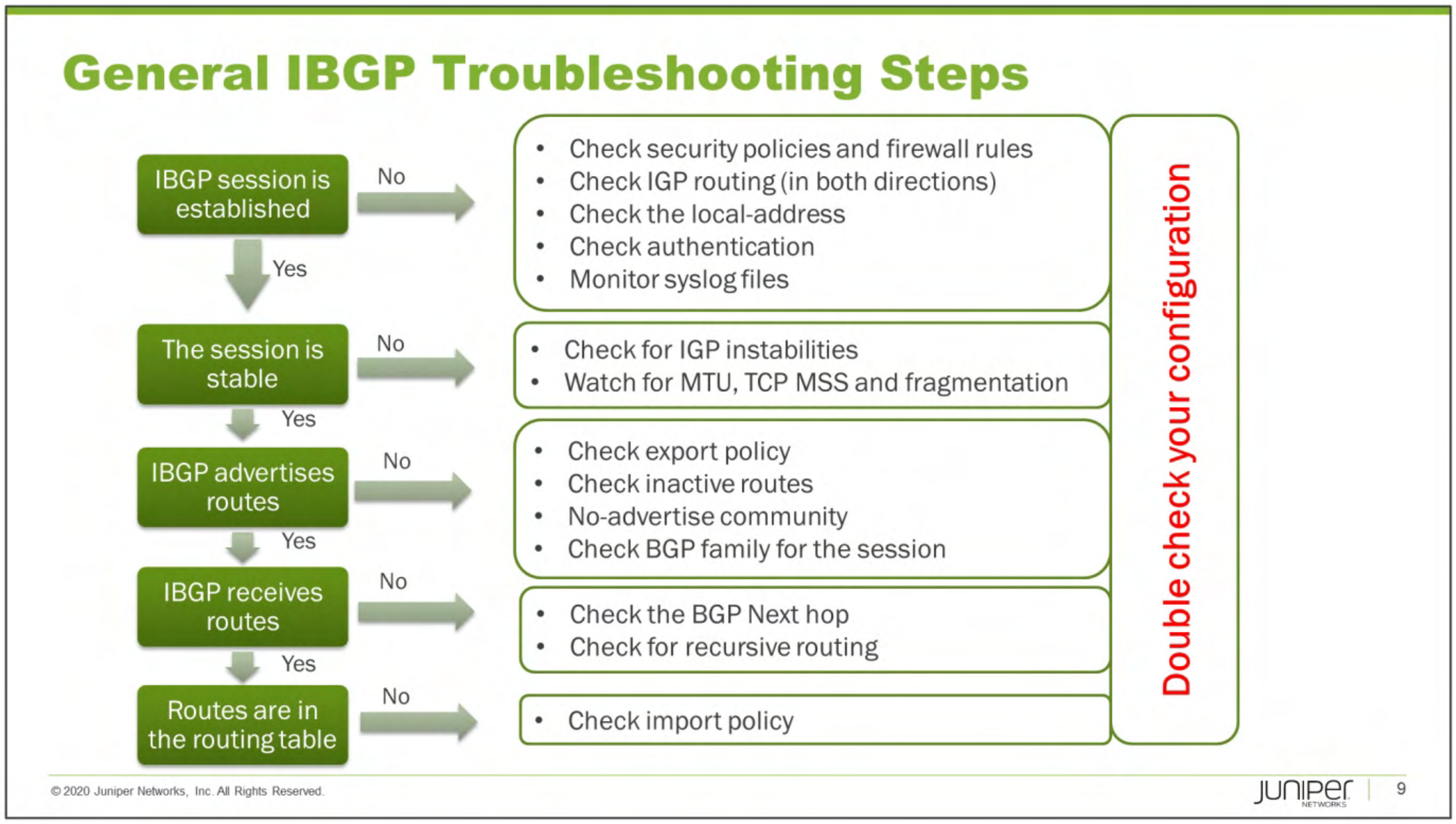
If routes are not being advertised, you might suspect an incorrect export filtering policy. By default BGP advertises all active BGP routes to its peers. An incorrect export policy can filter the routes that you want to advertise.

Often BGP ASBRs employ route aggregation by configuring aggregate or generated routes and exporting these routes to BGP. Recall that for an aggregate route to be active it must have at least one active contributing route in the routing table.

A BGP route can become inactive if it is shadowed by the same route learned from a protocol with a lower preference. Recall that you can make BGP advertise the inactive routes by configuring the `advertise-inactive` option.

Some routes might be tagged with one of the well-known communities regulating the route propagation. No-export, no-export-subconfed, and no-advertise are the three well-known community values. If mistakenly configured these communities will set a scope on the BGP route propagation, thus preventing the route advertisement.

A next-hop self policy applied to a RR can cause suboptimal routing by changing the BGP next-hop to the RR's address.



General IBGP Troubleshooting Steps

The purpose of the IBGP troubleshooting flowchart shown on the slide is to provide a set of high-level steps designed to get you started with IBGP fault analysis.

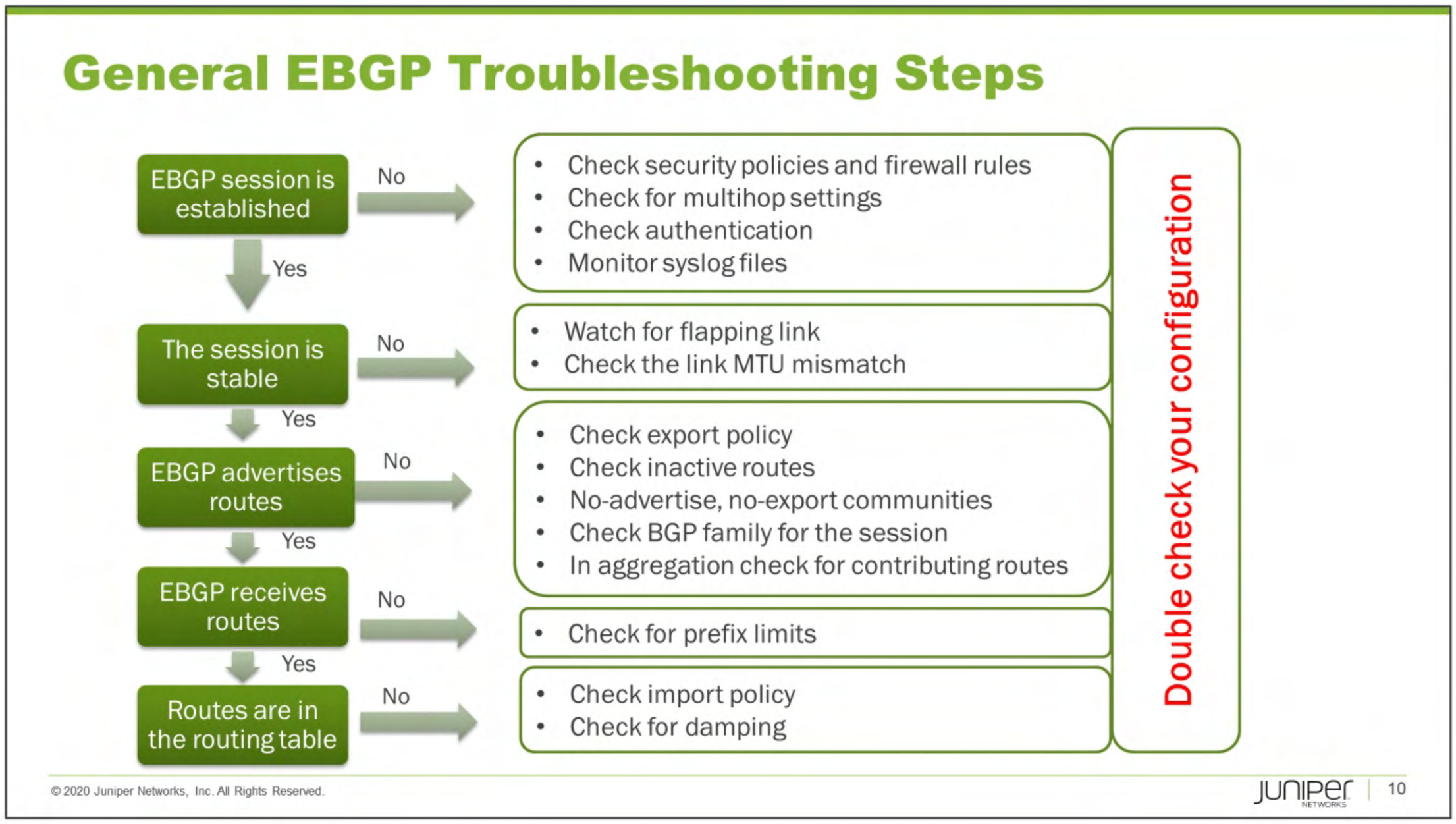
You begin with verifying theIBGP peering sessions state. If a session does not come up you start examining various parameters that we have discussed in the previous section that can affect the session establishment. First, you want to make sure that your security policies or firewall filters allow BGP protocol messages. For IBGP sessions you verify the remote peer loopback reachability by inspecting the IGP routing table. You may want to check the update source IP address, which is typically the loopback address. If BGP uses authentication, you have to make sure that authentication parameters match at both ends.

For more difficult issues, you may want to check the system TCP sockets, monitor the system log files or debug BGP with traceoptions.

Once the IBGP sessions come up you monitor the session state to ensure that they are stable. For flapping sessions watch for IGP instabilities, or TCP MSS related issues.

If the IBGP sessions are up and running, you proceed to examining BGP routes advertised and received by the router. Lack of advertised routes may be the indication of incorrect export policy action, presence of inactive routes, lack of configured BGP family support, or presence of routes tagged with the no-advertise community.

If some routes appear to be hidden, you can suspect the BGP next-hop resolution problem, or the recursive routing failure. The absence of expected BGP routes in the routing table may be also an indication of incorrect import policy filtering the routes.



General EBGP Troubleshooting Steps

The purpose of the EBGP troubleshooting flowchart shown on the slide is to provide a set of high-level steps designed to get you started with EBGP fault analysis.

You begin with verifying of EBGP peering sessions state. You generally follow the same procedures as for IBGP session troubleshooting.

Once the EBGP sessions come up, you monitor the session state to ensure that they are stable. For flapping sessions, watch for link behavior or MTU related issues.

If the EBGP sessions are up and running, you proceed to examining BGP routes advertised and received by the router. You generally follow the same procedures as for IBGP routing troubleshooting. Lack of advertised routes can be the indication of incorrect export policy action, presence of inactive routes, lack of configured BGP family support, or presence of routes tagged with the no-advertise or no-export community.

If a router is configured with the BGP prefix limit, it might not drop the BGP session depending on the prefix limit settings. By default, the prefix limit setting does not tear the session down if the limit is reached but just sends a warning message to the system log.

If some routes appear to be hidden, you can suspect an incorrect import policy filtering the routes or the recursive routing failure.

If route damping is enabled on a BGP ASBR, then some routes can be suppressed by the damping algorithm in cases of route oscillation.

Useful BGP Operational Commands (1 of 4)

- Check summary information about BGP peers

```
lab@srx> show bgp summary
Groups: 3 Peers: 3 Down peers: 2
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0               889        889          0          0          0          0
inet6.0                0          0          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|...
172.27.255.4     3895077211        417        417       0       1        1:54 889/889/889/0
172.27.255.7     3895077211          0          0       0       0       20:16 Active
201.201.0.1      2087403078          0          0       0       0          13 Idle
```

  - A session stays in `Idle` state:
    - BGP cannot even attempt to establish the session
  - A session bounces between `Connect` and `Active` states:
    - BGP is trying to initiate a TCP session or is waiting for the session to be initiated by a peer
    - BGP is sending open messages out but is not receiving required information in response
- Use `show system connections` to view TCP connection data

© 2020 Juniper Networks, Inc. All Rights Reserved.

Checking BGP Session State: Part 1

Use the **show bgp summary** command to view basic information about all BGP neighbors. This command is normally the first command that is run during BGP troubleshooting to verify the neighbor states of your BGP configuration.

This command displays the list of peers along with the peer AS number, the peer statistics and the peer state. The state column displays either the BGP state or the amount of prefixes received from the neighbor. If prefixes are displayed, it indicates that the peering has reached the established state.

If a state is displayed it indicates a failure of BGP to successfully connect to the peer. As covered earlier: a state of `Idle` indicates that the outgoing interface may be down or there is no route in the routing table to forward a TCP SYN message. A state of `Connect` indicates that TCP messages are being sent but no response has been received. This can be caused by a firewall filter blocking the BGP port or a misconfigured or nonexistent neighbor. Finally a state of `Active` can indicate that the BGP session is having an issue. Normal issues that can cause this state are an incorrect AS number or authentication issue.

It is important to remember that Idle, Connect, and Active are valid transitional states. There is an `Up/Dwn` timer in the output of this command. It can take time for a BGP connection to move to the established state. If the amount of time elapsed since the session began initializing is less than 90 seconds, you may want to wait before starting troubleshooting procedures.

A good command to use during initial startup of a session is **`show bgp summary | refresh 2`**. This will cause the bgp summary command to refresh every two seconds, so you don't have to rerun the command over and over. You use a ctrl C to stop the refresh.

Another valuable piece of information available by using this command is the `InPkt` and `OutPkt` counts. These numbers show the number of messages sent and received and can sometimes be helpful in determining if the issue is local or remote.

Useful BGP Operational Commands (2 of 4)

- Check summary information about local BGP groups

```
user@srx> show bgp group
Group Type: Internal    AS: 65000                    Local AS: 65000
  Name: ibgp            Index: 0                     Flags: <>
  Holdtime: 0
  Total peers: 4        Established: 4
  10.222.1.1+51717
  10.222.1.3+59820
  10.222.1.4+57550
  10.222.1.5+56737
  inet.0: 0/0/0/0

Group Type: External                               Local AS: 65000
  Name: as100           Index: 1                     Flags: <Export Eval>
  Export: [ ebgp-export ]
  Holdtime: 0
  Total peers: 1        Established: 1
  10.1.254.1+50692
  inet.0: 0/0/0/0
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Viewing BGP Group Parameters: Part 2

Use the **show bgp group** command to view the BGP group information. This command displays the established peers summary per BGP group. It is particularly useful in checking the group applied export policies.

Useful BGP Operational Commands (3 of 4)

- Check BGP neighbor session details

```
user@srx> show bgp neighbor 172.22.138.37
Peer: 172.22.138.37+179 AS 65530 Local: 172.22.138.38+60316 AS 65501
  Type: External    State: Established   Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: Cease
  Options: <Preference AddressFamily PeerAS Refresh>
  Address families configured: inet-unicast inet6-unicast
..
  NLRI for restart configured on peer: inet-unicast inet6-unicast
  NLRI advertised by peer: inet-unicast inet6-unicast
  NLRI for this session: inet-unicast inet6-unicast
  Peer supports Refresh capability (2)
  Stale routes from peer are kept for: 300
  Peer does not support Restarter functionality
  NLRI that restart is negotiated for: inet-unicast inet6-unicast
  NLRI of received end-of-rib markers: inet-unicast inet6-unicast
  NLRI of all end-of-rib markers sent: inet-unicast inet6-unicast
  Peer supports 4 byte AS extension (peer-as 65530)
..
Last traffic (seconds): Received 2    Sent 2    Checked 2
  Input messages:  Total 3094   Updates 18      Refreshes 0     Octets 59544
  Output messages: Total 3263   Updates 51      Refreshes 0     Octets 64854
..
```

Establish is the desired state

Last Error

Families negotiated

Messages sent and received

© 2020 Juniper Networks, Inc. All Rights Reserved.

Checking BGP Neighbor Details: Part 3

Use the **show bgp neighbor** command to display the neighbor information.

This command provides detailed information about the operational state of BGP on the local device. Through this command you can retrieve the port numbers used, the current state, the last or current error, and what address families were negotiated. You can also see the number of messages sent and received.

Useful BGP Operational Commands (4 of 4)

- Examine log files
  - Default log file is messages
  - For authentication errors use | match auth (short for authentication)
  - For other errors use | match noti (short for notification)

```
user@srx> show log messages | match noti
Nov 30 18:22:29  vSRX-1 rpd[7394]: bgp_process_open:4185: NOTIFICATION sent to 192.168.1.2 (Internal
AS 65000): code 2 (Open Message Error) subcode 2 (bad peer AS number), Reason: peer 192.168.1.2
(Internal AS 65000) claims 65100, 65000 configured
Nov 30 18:22:33  vSRX-1 rpd[7394]: bgp_pp_recv:4798: NOTIFICATION sent to 192.168.1.2+56774 (proto):
code 2 (Open Message Error) subcode 2 (bad peer AS number), Reason: no group for 192.168.1.2+56774
(proto) from AS 65100 found (peer as mismatch) in master(ge-0/0/1.0), dropping him
Nov 30 18:23:29  vSRX-1 kernel: tcp_auth_ok: Packet from 192.168.1.5:64047 missing MD5 digest
Nov 30 18:23:30  vSRX-1 kernel: tcp_auth_ok: Packet from 192.168.1.6:56201 missing MD5 digest
---(more)---
```

Can you spot any issues?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Examine Log Files: Part 4

Various BGP session status messages are reported to the system syslog daemon, which stores them in a system log file, typically /var/log/messages. Monitor the log file for BGP related issues.

The sample output on the slide shows an example of log entries that capture error conditions. One message indicates that there is a misconfigured AS number. The other indicates that there is an authentication mismatch.

Use Traceoptions

- For difficult problems, use traceoptions

```
[edit protocols bgp]
user@srx# show traceoptions
file bgp_trace.log size 10m files 2;
flag packets detail;
flag general;
flag open;
flag update;
flag all
```

Set the log file name and other information

Flags define what type of information to capture in log file

Can be used to capture everything

```
user@srx> show log bgp_trace.log
...
Nov 30 17:23:10.524959 BGP RECV 192.168.1.3+179 -> 192.168.1.1+63525
Nov 30 17:23:10.525000 BGP RECV message type 4 (KeepAlive) length 19
Nov 30 17:23:11.219817 bgp_connect_complete: error connecting to 192.168.1.2 (Internal AS
65000): Socket is not connected
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Use Traceoptions

Traceoptions provide the ability to view what is happening within the internal routing process. Traceoption flags define the category of data to capture in the file. Some of the common options are displayed in the slide. The `all` option will capture all output associated with the BGP process. Using this option will result in a large amount of data being captured.

The trace output is sent to a named log file in `/var/log` directory. You can view the log file by using the **show log *<filename>*** command or you can monitor the updates in the log file dynamically with the **monitor start *<filename>*** command. Remember to stop monitoring when you are done with the **monitor stop** command.

Since Junos has a clean separation of the forwarding and control plane, traceoptions has minimal impact on traffic forwarding. Even with limited impact, traceoptions should not be left running for long periods of time.

It is recommended that you deactivate rather that delete your traceoption configuration. Deactivation leaves the code in place but does not include it in the operational code. A common problem when users delete the traceoption configuration, is that each new traceoption ends up creating new files. Over time this can fill the local storage medium. By activating and deactivating the traceoptions configuration, the same file is reused each time, conserving disk space used.

Monitor in Real Time

- View real-time protocol traffic exchanges
  - **monitor traffic interface**

```
user@srx> monitor traffic interface ge-0/0/4.303 no-resolve detail matching tcp
Address resolution is OFF.
Listening on ge-0/0/4.303, capture size 1514 bytes

04:28:22.887679  In IP (tos 0xc0, ttl   1, id 34525, offset 0, flags [none], proto:
TCP (6), length: 170) 10.1.254.1.179 > 10.1.254.2.57891: P 79:197(118) ack 98 win
16365 <nop,nop,timestamp 834133740 836758381>: BGP, length: 118
       Keepalive Message (4), length: 19
       Update Message (2), length: 76
         Origin (1), length: 1, Flags [T]: IGP
         AS Path (2), length: 6, Flags [T]: 100
         Next Hop (3), length: 4, Flags [T]: 10.1.254.1
         Updated routes:
           10.1.100.0/24
           10.1.101.0/24
   ---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Monitor in Real Time

When the troubleshooting goal is just to see the protocol traffic that is flowing on the wire to see if the issue can be spotted without having to go through the process of setting up traceoptions, the **monitor traffic interface** command can be used.

This command provides you with a real time capture of the traffic. Many times this can provide deeper insight into the traffic flow to narrow down and identify the problem. You can use various filtering options in this command. For example, to capture only BGP messages you can use the following filter:

```
user@srx> monitor traffic interface ge-0/0/4.303 matching "tcp and port 179"
```

Verify Routing (1 of 4)

- Check which routes are being advertised

```
user@srx> show route advertising-protocol bgp 10.1.254.1

inet.0: 35 destinations, 35 routes (35 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.222.0.0/30            Self                                       I
* 10.222.0.4/30            Self                                       I
---(more)---
```

Shows that you are sending routes. Does not mean they were received

- Check which routes are being received

```
user@srx> show route receive-protocol bgp 10.1.254.1 <hidden|all>

inet.0: 52 destinations, 53 routes (52 active, 0 holddown, 1 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 0.0.0.0/0                10.1.254.1                                 100 I
* 10.1.100.0/24            10.1.254.1                                 100 I
* 10.1.101.0/24            10.1.254.1                                 100 I
---(more)---
```

Note: Hidden routes are routes dropped by policy

This command show routes received before Import policy modifications have been applied

© 2020 Juniper Networks, Inc. All Rights Reserved.

Verifying Routes: Part 1

Once the BGP peering has reached the established state, the issues that need to be addressed deal with routing. This means troubleshooting policies to see why routes are or are not appearing as expected in the routing table. The next few slides deal with commands that can be used to monitor and evaluate the routes being sent and received.

Displaying BGP Advertised Routes

The **show route advertising-protocol bqp** <*neighbor*> command displays routes that have been advertised to the specified BGP neighbor. The information reflects the routes after export policy processing. When validating that the export policy is working, you may need the extensive option to see all of the applied changes. For example the base command does not show the prepended local AS number nor does it include any communities. The `detail` option will include that information, which may aid in troubleshooting.

Displaying BGP Received Routes

The **show route receive protocol bqp** <*neighbor*> command displays the BGP routes received from the specified BGP neighbor. The information reflects the routes before import policy processing, with the exception of filtered routes which are hidden. These filtered routes can be viewed by adding the `all` or `hidden` option to the end of the command.

Verify Routing (2 of 4)

- Verify Import Policy Changes

View routes from a specific peer. Useful to verify if import changes have been applied

```
user@srx> show route protocol bgp source-gateway 10.1.254.1

inet.0: 78 destinations, 79 routes (78 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.16.0/24    *[BGP/170] 00:35:54, localpref 100, from 192.168.1.5
                      AS path: I, validation-state: unverified
                    >  to 172.22.0.10 via ge-0/0/3.0
192.168.17.0/24    *[BGP/170] 00:35:54, localpref 100, from 192.168.1.5
                      AS path: I, validation-state: unverified
                    >  to 172.22.0.10 via ge-0/0/3.0
192.168.18.0/24    *[BGP/170] 00:35:54, localpref 100, from 192.168.1.5
                      AS path: I, validation-state: unverified
                    >  to 172.22.0.10 via ge-0/0/3.0
192.168.19.0/24    *[BGP/170] 00:35:54, localpref 100, from 192.168.1.5
                      AS path: I, validation-state: unverified
                    >  to 172.22.0.10 via ge-0/0/3.0
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Verify Import Policies: Part 2

Since import polices are applied after the RIB-IN table, using the **`show route advertising-protocol bgp`** *<neighbor>* does not reflect the changes made by the import policy. If you want to see that your import policy is working as expected, you must look in the inet.0 routing table. While manually displaying individual routes and checking for the change is possible, there is a simple way to view all the routes received from a bgp peer. This is possible by using the `source-gateway` option with the previous command. This command can also be used with the `detail` option to view communities and other information that is not included in the standard output.

Verify Routing (3 of 4)

- Verify BGP routes using display options

```
user@srx> show route protocol bgp

0.0.0.0/0          *[BGP/170] 00:22:01, localpref 100, from 10.1.1.1
                     AS path: 100 I, validation-state: unverified
                    >  to 10.1.254.1 via ge-0/0/0.0
                   [BGP/170] 00:19:11, localpref 100, from 192.168.1.2
                     AS path: 200 I, validation-state: unverified
                    >  to 172.22.0.2 via ge-0/0/1.0
44.0.0.0/8         *[BGP/170] 00:22:01, localpref 100, from 10.1.1.1
                     AS path: 100 8918 11423 2152 7377 I, validation-state: unverified
                    >  to 172.22.0.2 via ge-0/0/1.0

user@srx> show route protocol bgp active-path

inet.0: 78 destinations, 79 routes (78 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[BGP/170] 00:22:29, localpref 100, from 10.1.1.1
                     AS path: 100 I, validation-state: unverified
                    >  to 10.1.254.1 via lt-0/0/0.0
44.0.0.0/8         *[BGP/170] 00:22:29, localpref 100, from 10.1.1.1
                     AS path: 100 8918 11423 2152 7377 I, validation-state: unverified
```

View all routes

View active routes only

© 2020 Juniper Networks, Inc. All Rights Reserved.

Verify BGP Routes Using Display Options

The `show route protocol bgp` command will display all of the BGP routes in your route table. This is not an issue when the number of BGP routes is small. However, as stated earlier there are over 800,000 IPv4 routes in a full Internet download and using such a generic command can make finding and troubleshooting BGP routes difficult.

Using one of the many display options can limit the number of routes you will need to see to verify if any policy issues are present. It is possible to receive BGP routes from more than one source. When looking at the route table, all known, valid routes are included in the output. If needed to evaluate only routes that are selected as active the `active-path` option can be included in the command.

Verify Routing (4 of 4)

- Using the detail option

```
user@srx> show route protocol bgp detail
```

View route details

```
inet.0: 78 destinations, 79 routes (78 active, 0 holddown, 0 hidden)
0.0.0.0/0 (2 entries, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect, Next hop index: 0
                Address: 0xc1de338
                Next-hop reference count: 34
                Source: 10.1.1.1
                Next hop type: Router, Next hop index: 598
                Next hop: 10.1.254.1 via lt-0/0/0.0, selected
                Session Id: 0x0
                Protocol next hop: 10.1.1.1
                Indirect next hop: 0xbf68584 262144 INH Session ID: 0x0
                State: <Active Ext>
                Local AS: 65000 Peer AS:   100
                Age: 25:56      Metric2: 0
                Validation State: unverified
                Task: BGP_100.10.1.1.1
                Announcement bits (3): 0-KRT 3-BGP_RT_Background 4-Resolve tree 1
                AS path: 100 I
                Accepted
                Localpref: 100
                Router ID: 10.1.1.1
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Using the Detail Option

Junos uses options to control the amount of information that appears in display output. When troubleshooting making sure you get the right amount of information to confirm or eliminate something as an issue is important. The five levels of detail available in many Junos show commands are:

1. terse
2. brief
3. <no option>
4. detail
5. extensive

For a quick one line response use the terse option. The brief and <no option> returns the same output. Once you add the detail option, all of the BGP attributes that are part of the route are displayed. Any attributes that were previously added to a route and changes made by policy will be visible in this output.

While extensive provides additional information such as the protocol header information, it generally is not used for routing policy troubleshooting.

Agenda: Troubleshooting BGP

- BGP Troubleshooting

→Case Study: Neighborship Issues

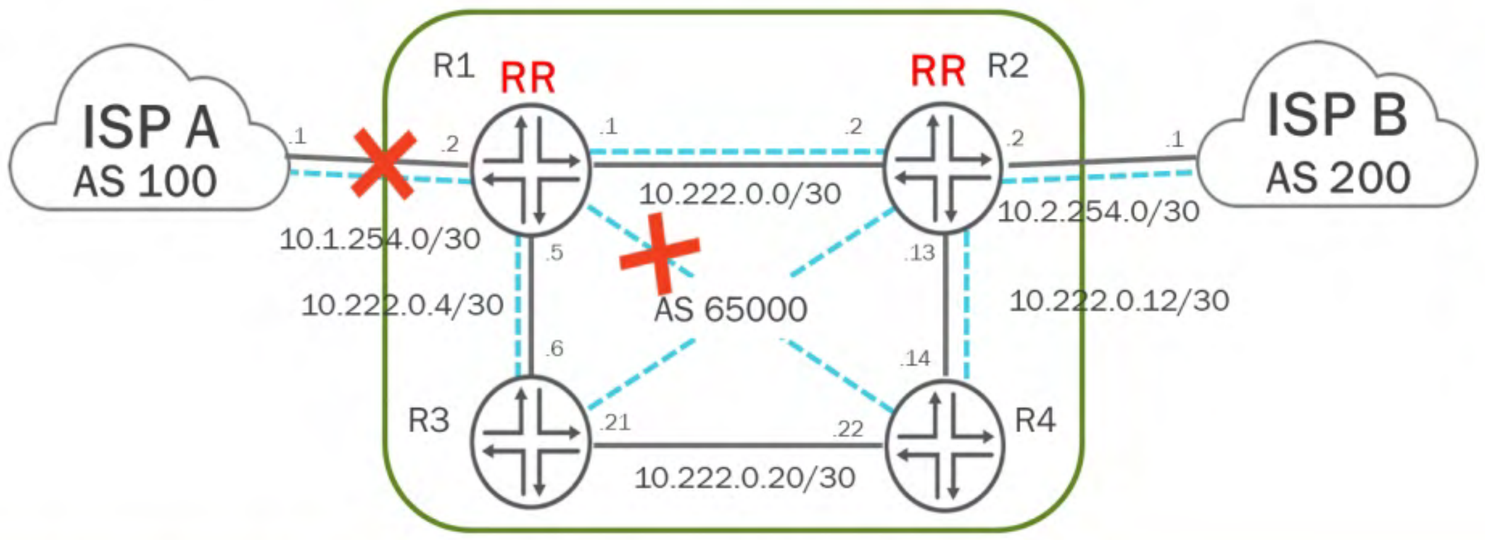
© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Case Study

The slide lists the topic we will discuss next.

BGP Case Study: Neighborship Issues

- Topology, symptoms, and relevant information:
  - Your company established a new branch office that requires redundant Internet connections using BGP. When BGP is configured and activated on the routers, you discover that the R1 EBGP session does not come up and the R1 IBGP session with R4 is flapping. Physical connectivity and IGP routing have been verified and they are working fine. The routers' security policies have also been checked and they permit BGP messages.



Router	Loopback
R1	10.222.1.2
R2	10.222.1.3
R3	10.222.1.4
R4	10.222.1.5

© 2020 Juniper Networks, Inc. All Rights Reserved.

BGP Case Study: Neighborship Issues

The slide presents a network topology, reported symptoms and description of a case related to BGP peering session issues. In this case study your company established a new branch office that requires BGP routing. When you configured and activated BGP on the routers the verification showed that R1 EBGP session does not come up and R1 BGP session with R4 is flapping. You verified all physical connections and they are working fine. You have also checked the security policies and have not found any issues.

Troubleshooting R1 EBGP Session (1 of 2)

- Check the session state

```
user@R1> show bgp summary
...
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
10.1.254.1              100       1799       1881       0      15          24 Active
10.222.1.3            65000       1905       1936       0       3    14:23:07 0/9/9/0              0/0/0/0
---(more)---
```

- Check the neighbor details

```
user@R1> show bgp neighbor 10.1.254.1
Peer: 10.1.254.1 AS 100        Local: 10.1.254.2 AS 65000
  Type: External    State: Active         Flags: <>
  Last State: Idle          Last Event: Start
  Last Error: Open Message Error
  Export: [ ebgp-export ]
  Options: <Preference AddressFamily PeerAS Refresh>
  Address families configured: inet-unicast inet6-unicast
  Holdtime: 90 Preference: 170
  Number of flaps: 15
  Last flap event: RecvNotify
  Error: 'Open Message Error' Sent: 6 Recv: 0
  Error: 'Cease' Sent: 13 Recv: 2
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Check R1 BGP Sessions

You begin with inspecting R1 BGP sessions. You use the **show bgp summary** command to get a brief summary of all the sessions. The command shows that R1 EBGP session with the neighbor 10.1.254.1 is in the Active state.

Check the Neighbor Details

You use the **show bgp neiqhbor** command to view the EBGP neighbor details. The output of the command indicates that there is an active error (Open message error).

Troubleshooting R1 EBGP Session (2 of 2)

- Look up the syslog file for BGP errors

```
user@R1> show log messages | match "open message"
Mar 19 00:07:31  R1 rpd[1325]: bgp_pp_recv:3124: NOTIFICATION sent to 10.1.254.1+52788
(proto): code 2 (Open Message Error) subcode 2 (bad peer AS number), Reason: no group for
10.1.254.1+52788 (proto) from AS 1000 found (peer as mismatch), dropping him
---(more)---
```

- Solution:
  - ISP A is AS 100 and they are trying to connect to your site using the wrong AS number (1000). They need to change their configuration.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Check the RI Syslog File

Because the neighbor session signal the Open message error you decide to look in the syslog file for the *open message* entries. The search presents a message pointing out to the source of the problem.

Solution

The neighboring autonomous system number is misconfigured. This needs to be corrected.

Troubleshooting R1-R4 IBGP Session (1 of 10)

- The session state monitoring indicate a flap every 1.5 minutes

```
user@R1> show bgp summary
...
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
...
10.222.1.4            65000        210        223       0       0     1:34:20 0/0/0/0              0/0/0/0
10.222.1.5            65000        216        294       0      26        1:29 0/0/0/0              0/0/0/0
```

```
user@R1> show bgp summary
...
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#Active/Received/Accepted/Damped...
...
10.222.1.4            65000        201        215       0       0     1:30:21 0/0/0/0              0/0/0/0
10.222.1.5            65000        207        280       0      27           4 Active
```

```
user@R1> show bgp summary
...
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
...
10.222.1.4            65000        210        223       0       0     1:34:20 0/0/0/0              0/0/0/0
10.222.1.5            65000        216        294       0      26           9 0/0/0/0              0/0/0/0
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Troubleshooting R1 BGP Session with R4

You proceed to troubleshooting the R1 BGP session with R4. A periodic issue of the **show bgp summary** command indicates that the session bounces every minute and a half.

Troubleshooting R1-R4 IBGP Session (2 of 10)

- View the syslog messages

```
user@R1> show log messages | match notification | match 10.222.1.5
Mar 19 00:38:26  R1 rpd[1325]: bgp_read_v4_message:10258: NOTIFICATION received from
10.222.1.5 (Internal AS 65000): code 4 (Hold Timer Expired Error), socket buffer sndcc: 259
rcvcc: 0 TCP state: 4, snd_una: 567635780 snd_nxt: 567636020 snd_wnd: 16384 rcv_nxt:
3933071576 rcv_adv: 3933087939, hold timer out 90s, hold timer remain 0.002139s
Mar 19 00:40:28  R1 rpd[1325]: bgp_read_v4_message:10258: NOTIFICATION received from
10.222.1.5 (Internal AS 65000): code 4 (Hold Timer Expired Error), socket buffer sndcc: 259
rcvcc: 0 TCP state: 4, snd_una: 1364380205 snd_nxt: 1364380464 snd_wnd: 16384 rcv_nxt:
2614298917 rcv_adv: 2614315280, hold timer out 90s, hold timer remain 1:29.999825s
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Examine the Syslog File

You decide to examine the syslog file for BGP notification messages. The lookup reveals that the messages file contains entries for BGP notifications received from R4. The reason for this problem is that the neighbor sends notification messages indicating the BGP hold timer is expiring.

Troubleshooting R1-R4 IBGP Session (3 of 10)

- Check the failed neighbor details

```
user@R1> show bgp neighbor 10.222.1.5
Peer: 10.222.1.5 AS 65000      Local: 10.222.1.2 AS 65000
  Type: Internal    State: Active       (route reflector client)Flags: <>
  Last State: Idle          Last Event: Start
  Last Error: Cease
  Options: <Preference LocalAddress Cluster Refresh>
  Local Address: 10.222.1.2 Holdtime: 90 Preference: 170
  Number of flaps: 34
  Last flap event: RecvNotify
  Error: 'Hold Timer Expired Error' Sent: 0 Recv: 33
  Error: 'Cease' Sent: 1 Recv: 0
```

- Is the neighbor flapping?

```
user@R1> ping 10.222.1.5 source 10.222.1.2
PING 10.222.1.5 (10.222.1.5): 56 data bytes
64 bytes from 10.222.1.5: icmp_seq=0 ttl=63 time=2.237 ms
64 bytes from 10.222.1.5: icmp_seq=1 ttl=63 time=1.850 ms
^C
--- 10.222.1.5 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.850/2.043/2.237/0.194 ms
```

Remember to source the ping from the local loopback

Based on several ping tests the peer is reachable and stable

© 2020 Juniper Networks, Inc. All Rights Reserved.

Check the R4 Neighbor Details

The **show bgp neighbor** command also shows the Hold Timer Expired Error.

Check the Connectivity

You proceed to monitoring end-to-end connectivity between the two IBGP neighbors. The ping tests and interface checks have not revealed any issues.

Troubleshooting R1-R4 IBGP Session (4 of 10)

- Validate that BGP messages are being exchanged between the peers
- Set up traceoptions to capture the BGP message exchange

```
user@R1> show log bgp.log | match "(BGP SEND)|(BGP RECV)"
Mar 19 02:03:57.497394 BGP SEND 10.222.1.2+57161 -> 10.222.1.5+179
Mar 19 02:03:57.497491 BGP SEND message type 1 (Open) length 59
Mar 19 02:03:57.503939 BGP RECV 10.222.1.5+179 -> 10.222.1.2+57161
Mar 19 02:03:57.504067 BGP RECV message type 1 (Open) length 59
Mar 19 02:03:57.504236 BGP SEND 10.222.1.2+57161 -> 10.222.1.5+179
Mar 19 02:03:57.504330 BGP SEND message type 4 (KeepAlive) length 19
Mar 19 02:03:57.507059 BGP RECV 10.222.1.5+179 -> 10.222.1.2+57161
Mar 19 02:03:57.507235 BGP RECV message type 4 (KeepAlive) length 19
Mar 19 02:03:57.512312 BGP SEND 10.222.1.2+57161 -> 10.222.1.5+179
Mar 19 02:03:57.512393 BGP SEND message type 2 (Update) length 83
Mar 19 02:03:57.512939 BGP SEND 10.222.1.2+57161 -> 10.222.1.5+179
---(more)---
```

- The message exchange is occurring regularly

© 2020 Juniper Networks, Inc. All Rights Reserved.

Debugging BGP at R1

You then enable the traceoptions at RI to debug BGP operations. You examine the trace log file for BGP message flow between the two IBGP neighbors. The debugging shows that the neighbors follow the standard BGP session establishment rules by exchanging open messages, then keepalive messages, then update messages.

Troubleshooting R1-R4 IBGP Session (5 of 10)

- Is R1 able to send routes to R4 when the connection is up?

```
user@R1> show route advertising-protocol bgp 10.222.1.5

inet.0: 53 destinations, 53 routes (53 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 0.0.0.0/0                Self                            100        100 I
* 10.1.100.0/24            Self                            100        100 I
* 10.1.101.0/24            Self                            100        100 I
---(more)---
```

- Does R4 receive routes from R1 when the connection is up?

```
user@R4> show route receive-protocol bgp 10.222.1.2


user@R4> show route hidden


user@R4> show route resolution unresolved
Tree Index 1
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Check BGP Advertised Routes at R1

You use the **show route advertising-protocol bgp <*neighbor*>** command on R1 to determine what is being advertised to R4. The command output shows that R1 advertises certain prefixes it received from the neighboring AS 100.

Check BGP Received Routes at R4

You use the **show route receive-protocol bgp <*neighbor*>** command on R4 to view the routes being received from R1. The command output shows that R4 does not receive any routes. You check the hidden routes and the routes which BGP next hop was not resolved but there are none.

Troubleshooting R1-R4 IBGP Session (6 of 10)

- Enable traceoptions at R4 and watch for BGP Update messages
  - There are none

```
user@R4> show log bgp.log | match "BGP RECV message type 2"
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Debugging BGP at R4

You proceed to debug BGP at R4. You enable traceoptions to trace only BGP update messages. The trace log file lookup does not show any received BGP update messages.

Troubleshooting R1-R4 IBGP Session (7 of 10)

- Determine the path the R1 Update messages to R4 would take

```
user@R1> traceroute 10.222.1.5 source 10.222.1.2
traceroute to 10.222.1.5 (10.222.1.5) from 10.222.1.2, 30 hops max, 40 byte packets
 1  10.222.0.6 (10.222.0.6)  2.113 ms  1.645 ms  1.784 ms
 2  10.222.1.5 (10.222.1.5)  1.762 ms  1.858 ms  1.608 ms
```

- Check the TCP MSS for the session

```
user@R1> show system connections inet extensive | find 10.222.1.5
tcp4       0    202  10.222.1.2.58691                          10.222.1.5.179
ESTABLISHED
...
    rttmin:        1000  mss:         500
     flags: REQ_SCALE RCVD_SCALE REQ_TSTMP RCVD_TSTMP SACK_PERMIT [0x100003e0]
---(more)---
```

- Check the path MTU

```
user@R1> ping 10.222.1.5 source 10.222.1.2 size 512 do-not-fragment
PING 10.222.1.5 (10.222.1.5): 512 data bytes
36 bytes from 10.222.0.6: frag needed and DF set (MTU 128)
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  00 021c 8ad2   2 0000  3f  01 974c 10.222.1.2  10.222.1.5
---(more)---
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Check the Route from R1 to R4

You decide to inspect the hop-by-hop BGP update message propagation. You check the path the BGP messages take with the traceroute command.

Check the TCP Parameters

Then you look up the TCP parameters of the IBGP session by using the show system connection extensive command. You find out that R1 is using TCP MSS of 500 for this IBGP session.

Check the Path MTU

You can test the end-to-end path MTU by generating ping requests with different payload sizes and **do-not-fragment** option set. The test revealed that the next hop router (R3) has an interface MTU of 128 so it cannot transmit the larger packets without fragmenting them.

Note that you rarely encounter an MTU of this size in a real network, but the lab simulation is intended to show you the generic approach to this kind of problem.

Troubleshooting R1-R4 IBGP Session (8 of 10)

- At this point, what would you check next?
  - Are there any firewall filters?

```
user@R4> show interfaces ge-* detail | match "input filter"

user@R4> show interfaces lo0.0 detail | match "input filter"
      Input Filters: block-frags
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Check for Firewall Filters

You then decide to thoroughly examine the firewall filters on the routers along the IGP path from R1 to R4. You find out that R4 has a firewall filter applied to its loopback interface.

Troubleshooting R1-R4 IBGP Session (9 of 10)

- Evaluate the firewall filter. Fragments are being rejected by R4

```
user@R4> show configuration firewall family inet filter block-frags
term 1 {
    from {
        is-fragment;
    }
    then {
        count fragments;
        discard;
    }
}
term 2 {
    then accept;
}
user@R4> show firewall

Filter: block-frags
Counters:
Name                                                Bytes              Packets
fragments                                          376755                 2899
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Examine the Firewall Filter

You examine the firewall filter and see that the filter rejects IP fragments.

Troubleshooting R1-R4 IBGP Session (10 of 10)

- Review the data that you have collected:
  - Large BGP packets (Updates) being are fragmented due to a low MTU setting on the link between R4 and R5
  - The R4 firewall filter drops all fragments that are destined for the Routing Engine
- What options do you have to solve this problem?
  - Enable BGP path MTU discovery
  - Make sure that MTU is large enough to support the BGP negotiated TCP MSS
  - Allow the fragments

© 2020 Juniper Networks, Inc. All Rights Reserved.

Collected Information

You found out that the initial phase of the IBGP session between R1 and R4 is going flawlessly but then when R1 starts advertising the BGP NLRIs in BGP update messages, which are essentially bigger TCP packets, the R3 router fragments them due to low MTU on its link to R4. The firewall filter on R4 the drops the fragments.

Solution

You can either configure the BGP path MTU discovery option or set the IP MTU on all the links to a higher value to ensure that the BGP packets will not be fragmented, or you can modify the firewall filter settings to accept the fragments.

Summary

- In this content, we:
  - Review common BGP troubleshooting procedures
  - List common BGP troubleshooting commands
  - Isolate different issues with BGP communication and configuration

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- The common BGP troubleshooting procedures;
- The common BGP troubleshooting commands; and
- How to Identify issues with BGP peering.

Review Questions

1. What are the default BGP route advertising rules?
2. How does using route reflectors change the default BGP route advertising rules?
3. Why can a mismatched MTU break an EBGP session?
4. List 2 ways you can solve a BGP next hop reachability problem?
5. When you issue a **`show bgp summary`** command, what is indicated by the Active state?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

3.

4.

5.

Lab: Troubleshooting BGP

- Monitor and troubleshoot the operation of a BGP network.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: Troubleshooting BGP

The slide lists the objective for this lab.

Answers to Review Questions

1.

By default BGP advertises all active BGP routes received from IBGP peers to EBGP peers only, and all active BGP routes received from EBGP peers to both IBGP and EBGP peers.

2.

Route reflectors are allowed to advertise BGP routes learned from IBGP peers to other IBGP peers. Route reflectors advertise all active IBGP routes received from clients to other clients and non-clients, and all active IBGP routes received from non-clients to clients only.

3.

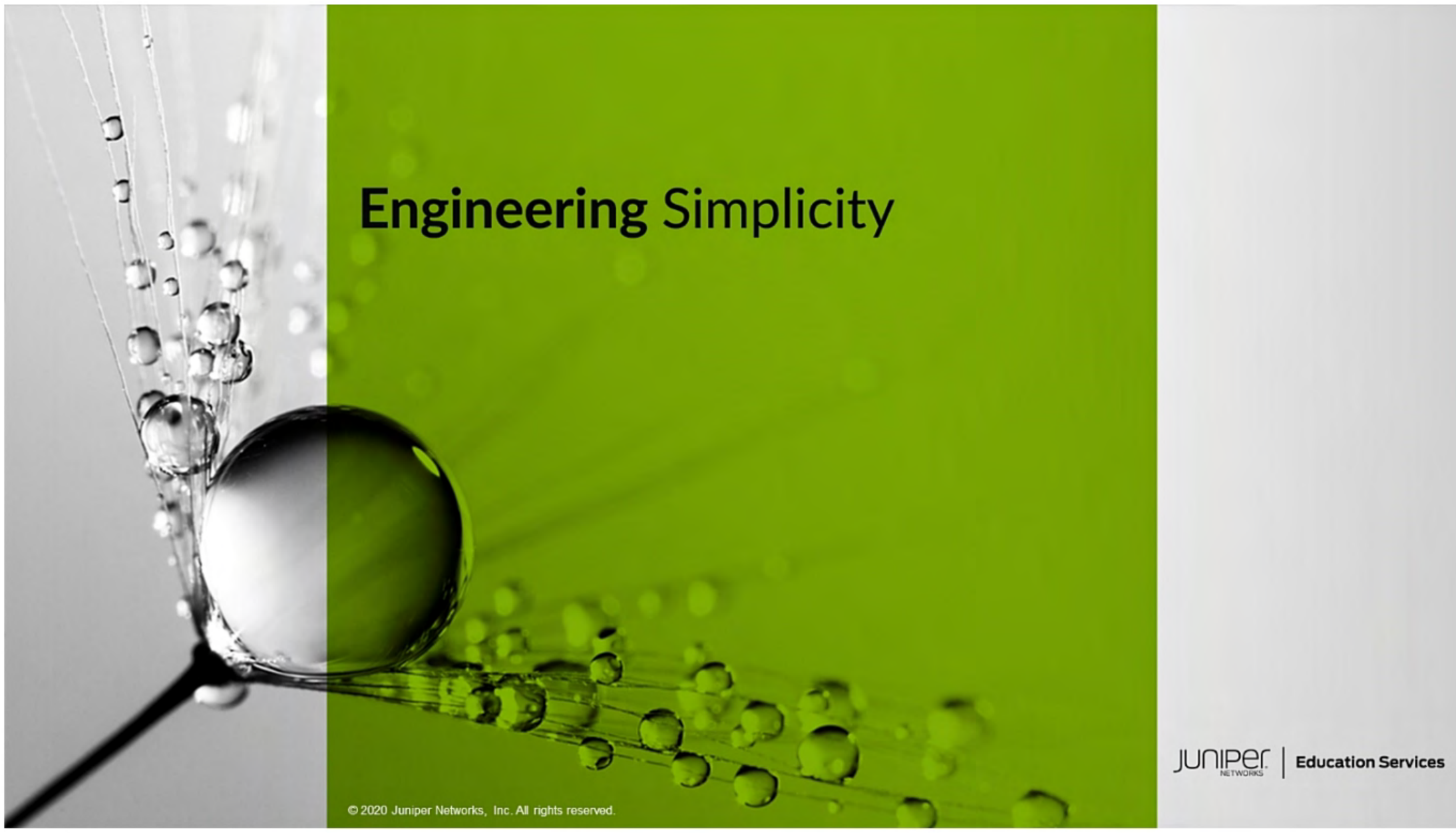
On directly connected EBGP sessions, TCP uses MTU-sized packets. If there is an MTU mismatch between the two sides of the TCP connection, large BGP packets cannot be delivered, which tears the BGP session down.

4.

The valid ways to solve the BGP next-hop problem are: next-hop-self policy, IGP passive interfaces, export direct routes into the IGP, static routes.

5.

The router is actively trying to form the BGP session


Engineering Simplicity
© 2020 Juniper Networks, Inc. All rights reserved.
Education Services

Education Services

Advanced Junos Service Provider Routing

Chapter 16: Policy Troubleshooting

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - Isolate problems in routing policy structure and configuration
  - Identify common commands for troubleshooting routing policy

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- Isolating problems in routing policy structure and configuration; and
- Identifying common commands for troubleshooting routing policy.

Agenda: Policy Troubleshooting

→Routing Policy Overview
- Policy Structure
- Using Regular Expressions
- Useful Commands
- Case Studies

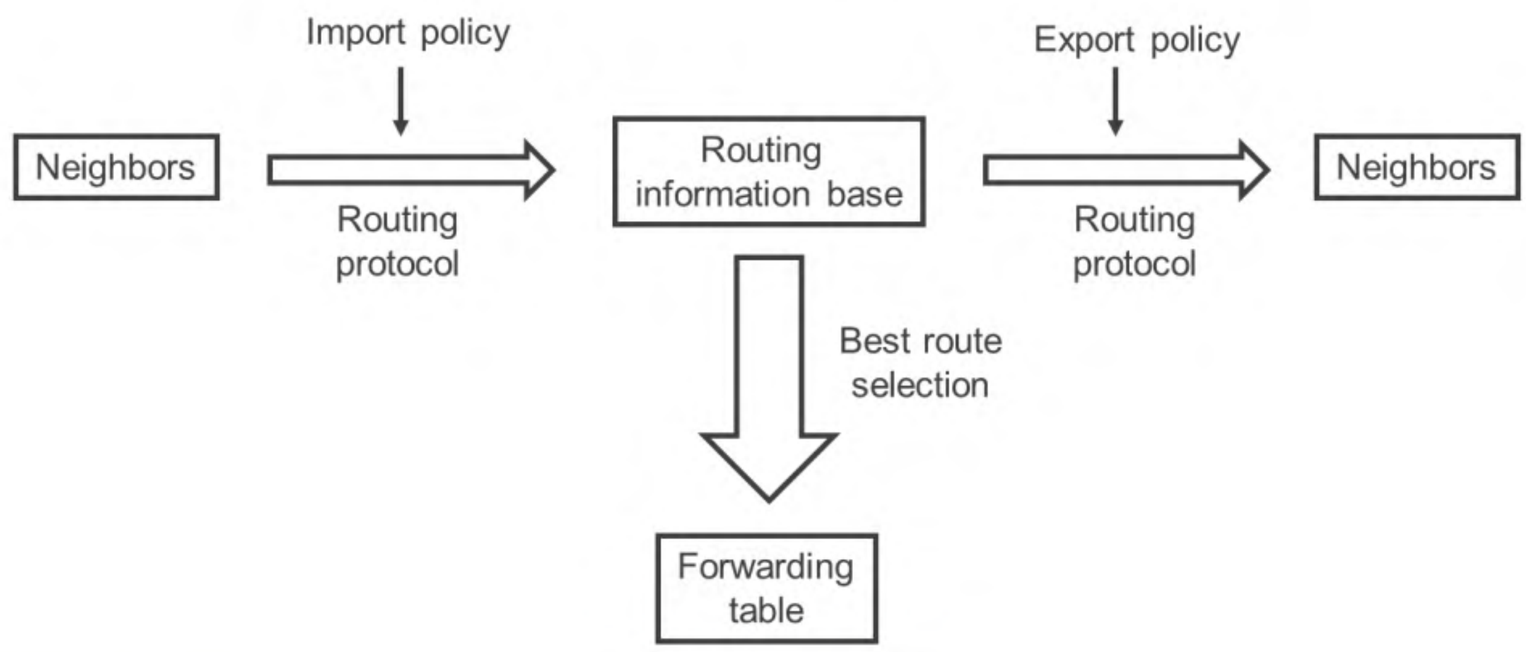
© 2020 Juniper Networks, Inc. All Rights Reserved.

Routing Policy Overview

The slide lists the topics we will discuss. We discuss the highlighted topic first.

Routing Policy Overview (1 of 3)

- Import and export protocol policies
  - The role of policies in the flow of routing information



© 2020 Juniper Networks, Inc. All Rights Reserved.

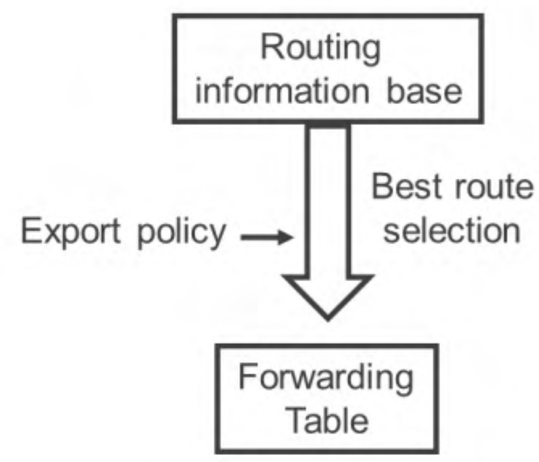
Import and Export Protocol Policies

The slide shows a high-level view of import and export policy processing. The import and export policies act as a control point in the exchange of routing information to and from neighbors. In both cases, on top of accepting (letting through) or rejecting (stopping) a route, policies can also modify its attributes.

Routing Policy Overview (2 of 3)

- Forwarding table export policy

```
user@router# show policy-options
policy-statement forwarding-policy {
    term premium-traffic {
        from {
            neighbor 192.168.78.5;
            community premium;
        }
        then {
            install-nexthop lsp premium-to-R5;
            accept;
        }
    }
    term non-premium-traffic {
        from neighbor 192.168.78.5;
        then {
            install-nexthop lsp nonpremium-to-R5;
            accept;
        }
    }
    term the-rest {
        then {
            load-balance per-packet;
        }
    }
```




```
user@router# show routing-options
autonomous-system 65512;
forwarding-table {
    export forwarding-policy;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Forwarding Table Export Policy

It is also possible to use policies to influence the installation of active routes in the forwarding table. The forwarding policies exist between the routing protocol process (rpd) and the Junos kernel, which then install them in the PFE. The consequences of a mistake in a forwarding policy can be dramatic—a good practice is never to reject anything. A 'reject' action in the last term will cause even interface routes to disappear from the forwarding table.

In the example, two BGP communities advertised by an IBGP speaker are used to decide which destination should take a *premium* label-switched path (LSP) and which a *non-premium* LSP. Typically, the non-premium LSP will include more hops, or links of worse quality in terms of delay. These applications are interesting, but add some complexity to the troubleshooting of forwarding issues.

Finally, note that the last term is used to enable load-balancing per-packet (Junos load balancing is actually per-flow). This configuration term is standard in many networks.

Routing Policy Overview (3 of 3)

- Routing policy maintenance
  - Policies are typically not static entities—they must be maintained
    - BGP update should be filtered to prevent rogue advertisements; failing to do this can cause worldwide outages (YouTube outage of February 2005)
    - Bogon lists should be kept up-to-date
    - Prefix lists for loopback filters need to be kept up-to-date
    - Temporary policies need to be cleaned up
  - Good operational procedures need to include a process to keep policies up-to-date
    - This includes regenerating BGP policies from routing registries—typically using automatic tools

© 2020 Juniper Networks, Inc. All Rights Reserved.

Routing Policy Maintenance

Routing policies are not static entities—and as such, they need to be maintained. Some typical examples are updating BGP route filters or prefix lists used in firewall filters; the BGP case is especially important, as carelessly accepting advertisements from peers can lead to worldwide outages.

In February 2005, YouTube vanished from the worldwide Internet for hours—a provider in Asia was trying to block access to its website, and ended up originating a route that was followed all over the world. This highlighted the vulnerability of the core Internet infrastructure—vulnerability that does not come from the protocols itself, but rather from lax operational procedures of core providers. Good operational procedures need to include processes to keep policies up to date.

Remote-Triggered Black Hole

- Another use of policies: DDoS mitigation
  - Routes to attack destinations are advertised with special communities
  - Traffic matching the tagged routes is then discarded

  Disadvantages:
  - Extremely coarse granularity—can stop volume attack traffic on the edge, but will also discard legitimate traffic
  - A better alternative: BGP FlowSpec
    - Allows to distribute firewall information through BGP
    - Can take actions, not only on the basis of source/destination, but also based on protocols
    - Can selectively drop amplified smurf attacks

© 2020 Juniper Networks, Inc. All Rights Reserved.

Remote-Triggered Black Hole

Another use of communities not strictly tied to routing is the creation of remotely-triggered blackhole routes to stop volume distributed denial-of-service (DDOS) attacks on the edge of a network, preserving bandwidth in the core. This usage was especially interesting as these attacks became popular, but in the last few years, new and better techniques have been developed to address the limitation of RTBH routes. On the Juniper platforms, BGP FlowSpec signaling allows you to drop attack traffic on the edge with a very fine granularity, while still protecting legitimate traffic.

Agenda: Policy Troubleshooting

- Routing Policy Overview
- →Policy Structure
- Using Regular Expressions
- Useful Commands
- Case Studies

© 2020 Juniper Networks, Inc. All Rights Reserved.

Policy Structure

The slide highlights the topic we discuss next.

Policy Structure (1 of 6)

- Policy syntax
  - A policy is a sequence of terms, each including a condition and an action

Term my-aggregate:
originates the BGP aggregate
which represents the backbone

Term next-hop-self:
change the next-hop
for EBGP routes

```
[edit policy-options]
user@router# show
policy-statement ibgp-export {
    term my-aggregate {
        from {
            protocol aggregate;
            route-filter 192.168.16.0/24 exact;
        }
        then accept;
    }
    term next-hop-self {
        from {
            protocol bgp;
            route-type external;
        }
        then {
            next-hop self;
        }
    }
```

Condition

Action

© 2020 Juniper Networks, Inc. All Rights Reserved.

Policy Syntax

Each policy is a sequence of terms that are typically composed of a condition and an action. Both the condition and actions are optional, all routes match terms that have no match conditions included, and the default action when not explicitly listed is next-term or next policy. Policy evaluation continues until routes encounter a terminating action like accept or reject. If neither is found, evaluation continues with the next policy. It is also possible to use flow-control statements like 'next policy' to stop evaluation of the current policy and continue to the next one, or to the protocol default policy (implicit final policy).

Policy Structure (2 of 6)

- Policies can be used without terms
  - For historical reasons, it is possible to have policies with no terms, just a condition and an action
    - Mixing the two styles can lead to serious mistakes

```
[edit policy-options]
user@router# show
policy-statement as65502-export {
    from {
        protocol bgp;
        community customers;
    }
    then accept;
}
```

=

```
[edit policy-options]
user@router# show
policy-statement as65502-export {
    term send-customer-routes {
        from {
            protocol bgp;
            community customers;
        }
        then accept;
    }
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Policies With and Without Terms

Mostly for historical reasons (backwards compatibility with earlier releases), it is still possible to define a policy which does not have terms, but just a condition and its action. Unfortunately, this makes disastrous mistakes possible when mixing the two styles. If an action is (by mistake) put outside any term of a policy, it will act as a catch-all action, and affect all routes which have neither been accepted nor rejected by previous terms. If a condition is put outside any term, the results are generally unpredictable. Be on the lookout for these trivial but potentially dangerous mistakes.

Policy Structure (3 of 6)

- The **to** statement
  - In some special cases, the policy includes a **to** statement to specify a destination (an IS-IS level or a routing instance)

IS-IS export policy
leak some prefixes from level 2 to level 1 to prevent suboptimal routing

Stop some specific routes from reaching level 2

```
[edit policy-options]
user@router# show
policy-statement isis-export {
    term leak-routes {
        from {
            route-filter 10.0.0.0/24 exact;
        }
        to level 1;
        then accept;
    }
    term stop-specifics {
        from {
            route-filter 192.168.17.0/24 longer;
        }
        to level 2;
        then reject;
    }
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

The to Statement

In some special cases, it is possible to use the **to** statement in a term. This is typically used to specify where matched routes are supposed to be exported to. Its use is mainly restricted to policies to distribute routes between instances, and IS-IS export policies for route leaking (Level 2 to Level 1) and aggregation (Level 1 to Level 2). The **to** statement is part of the match criteria (condition).

Policy Structure (4 of 6)

- Match conditions
  - Common match conditions
    - Protocol
    - Route filters and prefix lists
    - next-hop and interface
  - BGP-specific conditions
    - AS_PATH regular expressions
    - Communities regular expressions
    - Route type (internal or external)
    - Origin
    - Local preference
  - OSPF-specific conditions
    - Area
    - external type
    - route tag
  - ISIS-specific conditions
    - Level
    - route tag

© 2020 Juniper Networks, Inc. All Rights Reserved.

Match Conditions

The slide shows a summary of common match conditions. The list of all possible conditions is very long, and allows extensive checks on most (if not all) of the attributes of a route.

The full list is available in the Junos OS documentation, in the Policy Framework guide. For any attribute you can think of checking, chances are there is a condition that can be used to match based on it.

Policy Structure (5 of 6)

- Policy chains
  - When multiple import or export policies are configured, they are evaluated sequentially
    - As soon as a terminating condition is found and a decision is reached, evaluation stops
    - Order is important

```
export [ reject-private backbone-aggregate customers ];
```

    - In this example, the first policy will reject the private address space, the second policy advertises an aggregate for the backbone addressing space and the third policy advertises customer routes
    - If none of the policies comes to a terminating action, the default policy will be used to make a decision

© 2020 Juniper Networks, Inc. All Rights Reserved.

Policy Chains

When multiple import or export policies are configured, they are evaluated sequentially, from left to right. If a policy returns a terminating action (accept or reject), then evaluation stops. Otherwise, the route continues to the next policy. It is also possible to specify the flow-control action **`next-policy`**, even if this is seldom needed in real-world applications. If the last policy in the chain completes a route evaluation without issuing a terminating action, then the action will be decided by a protocol-specific default policy.

Policy Structure (6 of 6)

- Default import and export policies

Protocol	Import	Export
BGP	Accept all BGP IPv4 routes learned from configured neighbors.	Accept and export active BGP routes following BGP forwarding rules.
IS-IS	Accept all IS-IS LSPDUs and forward per IS-IS flooding rules. Import policies are not supported.	Used to redistribute routes into IS-IS. No routes are redistributed by default.
OSPF	Accept all OSPF LSAs and forward per OSPF flooding rules. Import policies are used to block external routes from being added to route tables.	Used to redistribute routes into OSPF. No routes are redistributed by default.
RIP	Accept all RIP routes learned from configured neighbors.	Reject everything.
RIPng	Accept all RIPng routes learned from configured neighbors.	Reject everything.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Default Import and Export Policies

The slide shows a summary of default policies for BGP and IGPs. The default behavior is mostly sensible in all cases, with the notable exception of RIP which by default accepts everything from peers, and advertises nothing. All routes that are neither accepted nor rejected by configured policies, will be evaluated by the protocol's default policy.

Policy Subroutines and Expressions (1 of 3)

- Policy subroutines
  - It is possible to use a policy as a condition in another policy

```
[edit policy-options]
user@router# show
policy-statement ibgp-export {
    term rip-routes {
        from policy is-rip-customer-
route;
        then accept;
    }
}
```

```
[edit policy-options]
user@router# show
policy-statement is-rip-customer-route {
    term 1 {
        from {
            protocol rip;
            route-filter 10.0.0.0/24 exact;
        }
        then accept;
    }
    term 2 {
        then reject;
    }
}
```

  - The policy subroutine should either accept or reject all routes. Otherwise, the behavior will be evaluated by the default policy of the protocol to which the calling subroutine is applied

© 2020 Juniper Networks, Inc. All Rights Reserved.

Policy Subroutines

It is possible to use a policy as a condition in another policy; in this case, a terminating action of 'accept' will cause the from policy condition to be true (verified), while an action of 'reject' will cause the condition to be false. In short, the two actions 'accept' and 'reject' are treated as logical values by the calling policy.

It is very important that the subroutine either accepts or rejects all possible routes because if a route does not meet any terminating action, it will be evaluated by the default policy of the protocol to which the calling policy has been applied. This behavior is typically not what the policy designer intended.

Policy Subroutines and Expressions (2 of 3)

- Policy expressions
  - It is possible to specify a logical expression as an export policy

    && logical AND
    || logical OR
    ! logical NOT

```
[edit protocols bgp]
user@router# show
group ibgp {
    type internal;
    export ( policy-one && !policy-two );
```

  - The results can be unpredictable if the policies only modify the route attributes and do not accept or reject the routes

© 2020 Juniper Networks, Inc. All Rights Reserved.

Policy Expressions

It is possible to use—as import or export policy—a logical expression composed by routing policies tied with the **&&**, **||**, or **!** operators. The expression is evaluated assigning the boolean value of false if a route is rejected by a policy, and a value of true if a route is accepted or subject to the 'next term' non-terminating action. While in theory this is an interesting idea, it is easy to get unpredictable results if the policies that comprise the expression change any of the route attributes—which could then be used by other policies in the expression.

Policy Subroutines and Expressions (3 of 3)

- Test your knowledge

A static route to 10.0.0.0/8 is in the routing table.
What will be the effect of these BGP export policies?

1. `(add-one || add-two)`
   Will return community one only
2. `(add-one && add-two)`
   Will return both communities
3. `(!add-one || add-two)`
   Will return both communities

Example for illustration only
It is against best practices to combine in an expression policies that alter route attributes

```
policy-statement add-one {
    term add-community {
        from {
            route-filter 10.0.0.0/8 exact;
        }
        then {
            community add one;
            accept;
        }
    }
}
policy-statement add-two {
    term add-community {
        from {
            route-filter 10.0.0.0/8 exact;
        }
        then {
            community add two;
            accept;
        }
    }
}
community one members 65501:1;
community two members 65501:2;
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Test Your Knowledge

This simple example shows why policy expression can be difficult to troubleshoot.

1. The route will be sent with only the community one; the first policy in the expression would return true, so the second policy is never evaluated as the result would not change. This is technically called 'short circuit logical operator evaluation'.
2. The route will be sent with both communities, because both policies need to be evaluated to work out the value of the expression.
3. Again, both policies need to be evaluated, so the route will be sent with both communities attached to it. Note that even if the first policy returned a 'reject' (so a Boolean false), the community is still added.

It is possible to get even more *interesting* results by combining in an expression policies that check on attributes and policies that set the same attributes. Even with a few policies, the intricacy of the expression evaluation should discourage any real-life application.

Agenda: Policy Troubleshooting

- Routing Policy Overview
- Policy Structure
- →Using Regular Expressions
- Useful Commands
- Case Studies

© 2020 Juniper Networks, Inc. All Rights Reserved.

Using Regular Expressions

The slide highlights the topic we discuss next.

Regular Expressions (1 of 6)

- What is a regular expression?
  - It is a formalism used to specify a string matching operation
  - Junos uses regular expressions in two main contexts, both related to BGP
    - AS_PATH matching: Selects routes which traversed a given sequence of autonomous systems
    - Community matching: Selects routes which are tagged with a given set of communities
  - The two main contexts differ in the atomic component of regular expression:
    - The full AS number for AS_PATH regex
    - A single character for communities
  - Regular expressions cannot be used directly in a policy. The regex is defined for an AS-path or community under `policy-options` as pre-defined objects.

© 2020 Juniper Networks, Inc. All Rights Reserved.

What is a Regular Expression?

A regular expression (regex) is a formalism used to specify a pattern matching operation. Their use is ubiquitous in computer systems: from programming languages, to queries, to command line shell syntax. In policies, Junos uses regular expressions in two contexts:

1. AS_PATH matching: to specify condition on a BGP route AS_PATH attribute; and
2. Community matching: to match on BGP communities attached to a route.

Even if the syntax is the same, the meaning of regular expressions in the two contexts is different: for AS_PATH regex use as atomic component the single autonomous system, while the community regex use the single character.

Regular Expressions (2 of 6)

Operator	Match Definition
{m,n}	At least m and at most n repetitions of term. Both m and n must be positive integers, and m must be smaller than n.
{m}	Exactly m repetitions of term. m must be a positive integer.
{m,}	m or more repetitions of term. m must be a positive integer.
*	Zero or more repetitions of term. This is equivalent to {0,}.
+	One or more repetitions of term. This is equivalent to {1,}.
?	Zero or one repetition of term. This is equivalent to {0,1}.
|	One of two terms on either side of the pipe.
–	Between a starting and ending range, inclusive.
^	A character at the beginning of a community attribute regular expression. An AS number at the beginning of an AS-path regular expression. This character is added implicitly; therefore, the use of it is optional.
$	A character at the end of a community attribute regular expression. An AS number at the end of an AS-path regular expression. This character is added implicitly; therefore, the use of it is optional.
( )	A group of terms that are enclosed in the parentheses. If enclosed in quotation marks with no intervening space ("()"), indicates a null. Intervening space between the parentheses and the terms is ignored.
[ ]	Set of characters. One character from the set can match. To specify the start and end of a range, use a hyphen (-).
!	The not operator.

© 2020 Juniper Networks, Inc. All Rights Reserved.

Regular Expression Syntax

The slide shows a small summary of regular expression syntax. The syntax mirrors closely the POSIX 1003.2 specifications, and it is equivalent to the UNIX extended regular expressions; the documentation (Policy Framework manual) provides examples of use of each operator.

Regular Expressions (3 of 6)

- The `as-path` regular expressions
  - The `as-path` uses the entire AS number as a single term within a regular expression:
    - 19{2,4} will match “19 19” “19 19 19” “19 19 19 19”
  - Character by character matches are not supported for AS-paths.

Pre-defined objects created

```
[edit policy-options]
user@router# show
...
as-path via-as300 ".* 300 .*";
as-path-group my-group {
    as-path alphanet "65001 .*";
    as-path beta_net "65002 .*";
    as-path gammanet "65003 .*";
    as-path internal "()";
}
```

```
policy-statement export-to-peer {
    term no-as300 {
        from as-path via-as300;
        then reject;
    }
    term my-group {
        from as-path-group my-group;
        then accept;
    }
}
```

Pre-defined objects used

© 2020 Juniper Networks, Inc. All Rights Reserved.

`as-path` and Community Expressions

The slide provides some examples of `as-path` regular expressions. It is also possible to group multiple expressions into a single as-path-group; the condition will be considered a match if at least one of the expressions matches the route.

Regular Expressions (4 of 6)

- Community regular expressions
  - Communities use each individual character as a regular expression term:
    - 65500:9{2,3} will match 65500:99 65500:999
  - The portion before the colon (:) usually identifies the originating AS. Both 2 and 4-byte AS numbers are supported

```
[edit policy-options]
user@router# show
...
community customer members 65501:200;
community my-own members 65501:.*;
```

Pre-defined objects created

```
policy-statement tag-customer {
    term remove-own-tags {
        from community my-own;
        then {
            community delete my-own;
        }
    }
    term tag-customer {
        then {
            community add customer;
        }
```

Pre-defined objects used

© 2020 Juniper Networks, Inc. All Rights Reserved.

Community Regular Expressions

Communities use the character as a regular expression atomic entity. The first part of the community, before the colon, typically contains the originating AS; Junos supports both 2 and 4-bytes autonomous system numbers since version 9.1.

Regular Expressions (5 of 6)

- More on communities and communities regex
  - When defining a community named-object with multiple members
    - Setting it will cause all communities to be set
    - Matching on it will return true only if the route has all the communities
  - Do not confuse **`community set`** and **`community add`**
    - **`community set`** will remove all pre-existing communities
    - **`community add`** will add new communities to a pre-existing list of communities
  - Some common communities are pre-configured
    - **`no-export`**
    - **`no-advertise`**
    - **`no-export-subconfed`**

© 2020 Juniper Networks, Inc. All Rights Reserved.

More on Communities and Community Regex

It is possible to define community identifiers with multiple members. In this case, when using the identifiers in match condition (from community), the condition will be considered verified only if the route is tagged by all member communities.

Remember that **`community set`** does not just add a community to a route, it also removes all pre-existing communities. This is very rarely done; more often, the **`community add`** action is used.

Finally, Junos recognizes three predefined communities:

- `no-export`: Routes tagged with this community should not leave the autonomous system;
- `no-advertise`: Routes tagged with this community should never be advertised (not even using IBGP); and
- `no-export-subconfed` (used in confederations only): Routes tagged with this community should never leave the confederation sub-as.

Regular Expressions (6 of 6)

- Tips on regular expressions
  - Troubleshooting regular expressions can be very difficult
    - The only approach is to try and break them into terms, and look at each term individually
  - The best strategy is to simplify whenever possible
    - When matching multiple AS_PATHs, consider avoiding the use of the logical or operator, in favor of the as-path-group statement

```
[edit policy-options]
user@router# show
as-path from-transit-provider "(65000 65001 .*)|(65002*)|(65003 65004)";
as-path-group my-group {
    as-path alphanet "(65000 65001 .*)";
    as-path betanet "(65003 65004)";
    as-path gammanet "(65002*)";
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Tips for Regular Expression Use

Troubleshooting regular expressions can be very difficult; the best strategy is to avoid the problem by keeping expressions as simple as possible.

Often, a BGP as-path expression will be composed of just a few terms—your peers' AS numbers, or possibly another AS which you would like to avoid using for transit.

Agenda: Policy Troubleshooting

- Routing Policy Overview
- Policy Structure
- Using Regular Expressions
- →Useful Commands
- Case Studies

© 2020 Juniper Networks, Inc. All Rights Reserved.

Useful Commands

The slide highlights the topic we discuss next.

Policies: Useful Commands (1 of 5)

- **`show route forwarding table`**
  - Allows you to verify the effects of forwarding table policies
  - Especially useful to see if load-balancing per-packet is working as expected

```
user@router> show route forwarding-table destination 0/0
Routing table: default.inet
Internet:
Destination        Type RtRef Next hop           Type Index NhRef Netif
default            user     0                    ulst 1048576     2
                              f8:c0:1:1d:4e:15   ucst  1122     9 ge-1/0/5.0
                              f8:c0:1:1d:4e:16   ucst  1100    12 ge-1/0/6.0
                              f8:c0:1:1d:4d:fc   ucst  1095    17 ge-1/1/4.0
...
```

  - `ulst` is the list of unicast next hops among which traffic is load-balanced

© 2020 Juniper Networks, Inc. All Rights Reserved.

The `show route forwarding table` Command

The **`show route forwarding table`** command allows you to check the effect of forwarding table policies. It is often used to check if load-balancing is working as expected; a destination which is being load-balanced will have a next-hop of type `ulst` (Unicast Next-Hop List), whose component will be the next-hops among which traffic is being load-balanced.

Policies: Useful Commands (2 of 5)

- **show policy**
  - Displays information about configured routing policies as they are parsed by the Junos policy engine
  - Can highlight stray actions outside terms and similar problems

```
user@router> show policy
Configured policies:
forwarding-policy
```

```
user@router> show policy forwarding-policy
Policy forwarding-policy:
    Term premium-traffic:
        from peer 192.168.78.5
         community premium [65001:10 ]
        then install-nexthop [ premium-to-R5 ] nh_sel: 25cc444 accept
    Term non-premium-traffic:
        from peer 192.168.78.5
        then install-nexthop [ nonpremium-to-R5 ] nh_sel: 25cc888 accept
    Term the-rest:
        then load-balance per-packet
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

The **show policy** Command

The **show policy** command displays information about configured routing policies as they are parsed by the Junos OS policy engine. The command in itself is not overly useful, but can catch stray actions outside terms and similar problems.

Policies: Useful Commands (3 of 5)

- The **`test policy`** command is very limited
  - Major limitations:
    - For some protocols (BGP), it can give different results from the ones you would get by applying the policy
    - Routes that are not explicitly rejected will be considered accepted. This can lead you to assume that routes will be accepted when they will in fact be rejected by the protocol's default policy
  - Can be used to check route filters, in some easy cases
    - A good example would be a policy that rejects bogons and invalid routes, and accept the rest

© 2020 Juniper Networks, Inc. All Rights Reserved.

The `test policy` Command Is Very Limited

The **`test policy`** command is the only command dedicated to testing policies, however, it is not very useful for two main reasons:

1. It behaves—in some conditions—differently from how a policy would behave when applied to the protocol; this makes the test rather pointless. Unfortunately, one of the affected protocols is BGP—which is typically making heavy use of complex policies.
2. Routes that are neither accepted nor rejected are considered as accepted by the policy. Because of this, the command output can be misleading; it can lead you to assume that routes will be accepted when the routes will in fact be rejected by the protocol's default policy.

Policies: Useful Commands (4 of 5)

- Tracing policies
  - For hard-to-debug policy problems, it is possible to trace policy evaluation
    - Configure **traceoptions flag policy** under routing-options
    - Use the **then trace** in the policy terms you want to trace

```
[edit routing-options]
user@router# show
traceoptions {
    file policy size 10m;
    flag policy;
}
```

```
policy-statement add-one {
    term add-community {
        from {
            route-filter 10.0.0.0/8 exact;
        }
        then {
            community add one;
            trace;
            accept;
        }
```

```
user@router> show log policy | match trace
Apr  3 11:46:16 trace_on: Tracing to "/var/log/R1/policy" started
Apr  3 11:46:16.366702 policy_export_trace: Prefix 10.0.0.0/8 term add-community --> accept
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Training Policies

It is possible to trace policy evaluation by configuring **traceoptions flag policy** under **routing-options**, and using the **then trace** action in your policies to find out which terms are accepting or rejecting routes. This is a very powerful tool, but needs to be used with a bit of care; enabling traceoptions flag policy, even without trace actions, tends to generate lots of information. Especially with full routing or route instability, traceoptions should only be kept for debugging, and removed as soon as possible. Otherwise, the hard disk will be constantly writing. Also because the term name is included in the traceoptions output, the term names in policies would need to be unique if tracing was enabled in multiple policies. The trace action is useful when trying to understand complex policies using boolean expressions.

Policies: Useful Commands (5 of 5)

- Use protocol-specific commands to test policy effect
  - For OSPF and IS-IS redistribution policies, check the link-state database
    - **`show ospf database external advertising-router`** ***`router`*** **`| match`** ***`prefix`***
    - **`show isis database`** ***`router`*** **`| match`** ***`prefix`***
  - For BGP, check received and advertised routes
    - **`show route advertise-protocol bgp`** ***`neighbor`***
    - **`show route receive-protocol bgp`** ***`neighbor`***
    - To display routes filtered by import policies, add **`hidden`**
  - For LDP, check the LDP database
    - **`show ldp database`**

© 2020 Juniper Networks, Inc. All Rights Reserved.

Use Protocol-Specific Commands to Test Policy Effect

Even if the **`test policy`** command has serious limitations, it is still possible to check policy operation by looking at their effect on routing protocols. These commands are protocol-specific; the slide shows some examples.

Agenda: Policy Troubleshooting

- Routing Policy Overview
- Policy Structure
- Using Regular Expressions
- Useful Commands

→Case Studies

© 2020 Juniper Networks, Inc. All Rights Reserved.

Case Studies

The slide highlights the topic we discuss next.

Case Study: IS-IS Major Outage

- A simple mistake
  - This policy—intended to redistribute a static route—caused a major outage for a valued customer

```
[edit policy-options]
user@router# show
policy-statement isis-export {
    term send-one-static {
        from {
            protocol static;
            route-filter 10.17.21.0/24 exact;
        }
    }
    then accept;
}
```

  - The problem should be easy to spot—but it is also an easy one to make

© 2020 Juniper Networks, Inc. All Rights Reserved.

A Common Mistake

Remember that if an action is—by mistake—put outside any term of a policy, it will act as a catch-all action, and affect all routes which have neither been accepted nor rejected by previous terms. In this example, one such mistake caused a massive outage on the network of a valued customer. The incorrect action 'then accept' causes the entire routing table to be redistributed into IS-IS, which in turn causes the router to assert the overload bit, and several of its (non-Juniper) neighbors to experience excessive CPU load and malfunction.

Watch for improperly applied actions in your policies. Best practice is to always use terms, even for short policies.

This is one of the few cases where the **show policy** command makes the problem immediately evident:

```
user@router> show policy isis-export
Policy isis-export:
    Term send-one-static:
        from proto Static
            route filter:
                10.17.21.0/24 exact
    Term unnamed:
        then accept
```

Case Study: OSPF Redistribution (1 of 3)

- A simple redistribution
  - A network must redistribute routes belonging to the 10.0.0.0/8 aggregate from IS-IS into OSPF
    - One of those routes, 10.0.0.0/24, also has two more specific routes. The more specific routes must be blocked

```
user@router> show route 10.0.0.0/8

inet.0: 35 destinations, 35 routes (35 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.0.0/20        *[IS-IS/160] 00:00:30, metric 10
                    > to 172.22.138.1 via ge-1/1/4.0
10.0.0.0/24        *[IS-IS/160] 00:00:30, metric 10
                    > to 172.22.138.1 via ge-1/1/4.0
10.0.0.0/25        *[IS-IS/160] 00:00:30, metric 10
                    > to 172.22.138.1 via ge-1/1/4.0
10.0.0.128/25      *[IS-IS/160] 00:00:30, metric 10
                    > to 172.22.138.1 via ge-1/1/4.0
```

Redistribute

Block

© 2020 Juniper Networks, Inc. All Rights Reserved.

A Simple Redistribution

The slide shows a very simple case, but illustrates an interesting aspect of Junos routing policies. A network needs to distribute several routes in the 10.0.0.0/8 range from IS-IS into OSPF. One of those routes, 10.0.0.0/24, happens to have two more specific routes that need to be blocked. Only the /24 prefix should be distributed. The problem is very simple, and there are many ways of writing a redistribution policy.

Case Study: OSPF Redistribution (2 of 3)

- The export policy was applied, but missed a route
  - 10.0.0.0/20 was redistributed, but 10.0.0.0/24 was not

```
user@router> show configuration policy-options
policy-statement 10 {
    term 10 {
        from {
            protocol isis;
            route-filter 10.0.0.0/8 longer;
            route-filter 10.0.0.0/24 longer reject;
        }
        then accept;
. . .
user@router> show ospf database external detail
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *10.0.15.255      192.168.78.2     0x80000001   725  0x22 0xb52f  36
  mask 255.255.240.0
```

- Can you see what the problem could be?

© 2020 Juniper Networks, Inc. All Rights Reserved.

The Export Policy Was Applied, But Missed a Route

Checking the content of the link-state database, only one external route was present—the one for the 10.0.0.0/20 IS-IS route.

Clearly there must be something wrong with the policy. Can you see what the problem could be?

Case Study: OSPF Redistribution (3 of 3)

- A solution
  - This is a normal effect of how Junos matches route filters
    - First, Junos performs a prefix search for the most specific route filter
    - Then, it checks the prefix-length operator
    - There is no re-evaluation of route-filters if the prefix-length operator fails

```
user@router> show configuration policy-options
policy-statement 10 {
    term 10 {
        from {
            protocol isis;
            route-filter 10.0.0.0/8 longer;
            route-filter 10.0.0.0/24 longer reject;
        }
        then accept;
```

Action defined after route filter

  - A solution could be to use a single filter route-filter **`route-filter 10.0.0.0/8 upto /24`**

© 2020 Juniper Networks, Inc. All Rights Reserved.

A Solution

By using match conditions carelessly like **`longer`**, **`orlonger`**, and so forth, policies can produce counterintuitive results. This is due to how Junos applies route filters—first doing a prefix search, and then afterwards checking the prefix-length operator. This is no other way in Junos to re-evaluate the route-filters, if the prefix-length operator fails to match.

It is also worth mentioning that by placing the actions to be taken after the route filters definition, can make policies more difficult to troubleshoot. Placing actions on the same line as the route-filter is not generally considered best practice.

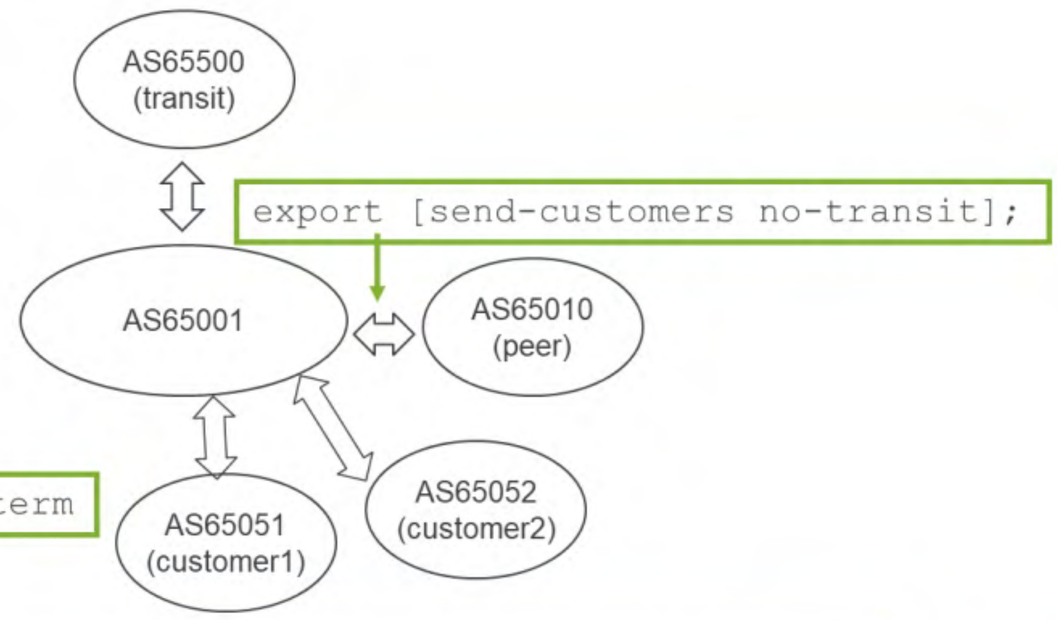
Case Study: Regex Confusion (1 of 2)

- Standard requirements in an ISP environment
  - Advertise customer routes to peers and transit providers
  - Advertise transit and peers routes to customer

```
[edit policy-options]
user@router# show
policy-statement send-customers {
    term customer-as-path {
        from as-path customers;
        then accept;
    }
}
policy-statement no-transit {
    term transit-as-path {
        from as-path transit;
        then reject;
    }
}

as-path customers "6505[1-2]";
as-path transit "65500";
```

AS path is one term

Only matches when there is one AS in the AS path



© 2020 Juniper Networks, Inc. All Rights Reserved.

Case Study: Regex Confusion

This is a very easy case study, but the problem it shows is sometimes found in the field—especially when operators are migrating from other vendors which use a different syntax. In a ISP environment, very often a standard requirement can be summarized this way:

- Advertise customer routes to transit (upstream) and peers; and
- Advertise transit and peer routes to customers.

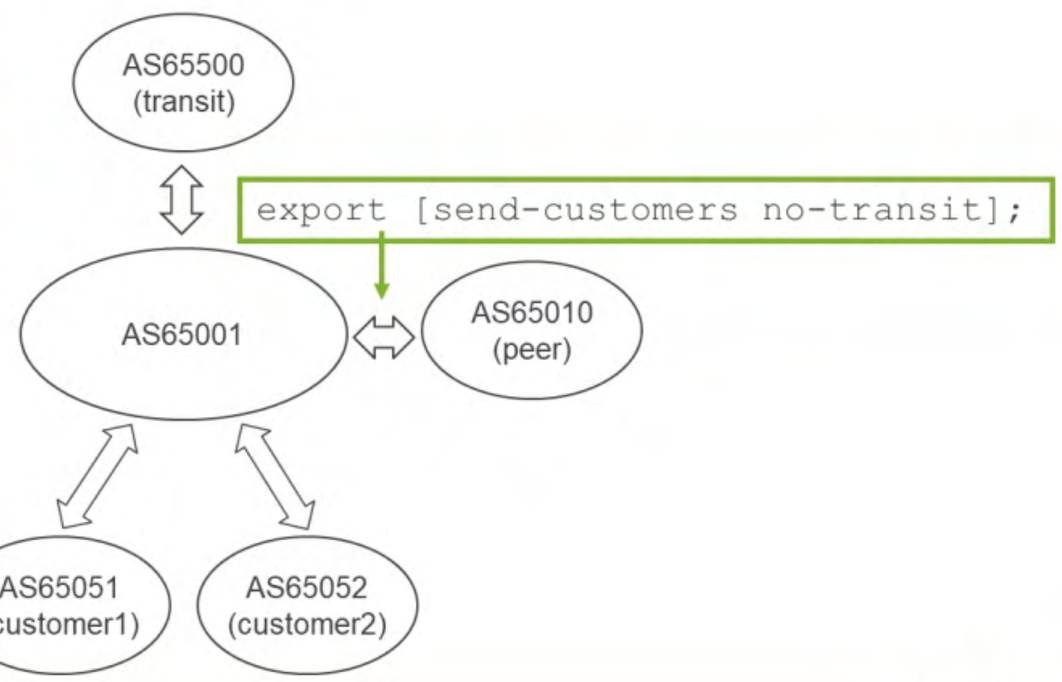
Advertising transit routes to peers should be avoided—differently from peer connectivity, transit connectivity is not free; you would end up paying for your peers' traffic.

The problems on the regular expressions shown on the slide should be evident. What would be the best solution?

Case Study: Regex Confusion (2 of 2)

- A possible solution
  - Use correct syntax in the regular expressions for customers
  - Add .* to the regular expression for transit routes

```
[edit policy-options]
user@router# show
policy-statement send-customers {
    term customer-as-path {
        from as-path-group customers;
        then accept;
    }
}
policy-statement no-transit {
    term transit-as-path {
        from as-path transit;
        then reject;
    }
}
as-path-group customers {
    as-path customer1 65051*;
    as-path customer2 65052*;
}
as-path transit "65500 .*";
```



© 2020 Juniper Networks, Inc. All Rights Reserved.

A Possible Solution

The policy failed to meet both of its requirements. The regular expressions were both wrong.

The first problem was the syntax of the regular expression used to match customers. As it was, it did not match any of the two customer autonomous systems. A possible way of solving this problem would be to use an **`as-path-group`** rather than a bigger as-path regular expression containing all customer ASs. The expressions for each customer have also been changed to take AS_PATH prepend into account. Each customer should be free to prepend their own AS as many times as they want.

The second problem was the regular expression used to match the transit routes. It would only match routes originating in AS65500, ignoring routes originating from an AS behind it. This resulted in AS65501 acting as a transit for AS65010, which is exactly what we wanted to prevent.

The use of regex for this application is neither very practical nor scalable, using BGP communities for this would be much simpler.

Summary

- In this content, we:
  - Discussed troubleshooting problems on routing policy structure and configuration
  - Identified common commands for troubleshooting routing policy

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- Isolating problems on routing policy structure and configuration; and
- Identifying common commands for troubleshooting routing policy.

Review Questions

1. What are some uses of a forwarding table export policy?
2. Why can actions outside terms be dangerous?
3. How can you trace a policy evaluation?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

3.

Lab: Troubleshooting Routing Policy

- Troubleshoot routing and forwarding policies

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: Troubleshooting Routing Policy

The slide provides the objective for this lab.

Answers to Review Questions

1.

The most common one is load-balancing across parallel path; it can also be used for other purposes for example to tie specific traffic to specific LSPs.

2.

Actions outside terms can be dangerous because they are difficult to spot, and apply to all routes not matching any of the terms; this overrides the protocol default policy, often with problematic results.

3.

Use the trace action in your policy, then configure traceoptions under routing-options.

Resources to Help You Learn More

Resource	URL
Pathfinder	http://pathfinder.juniper.net
Content Explorer	http://www.juniper.net/techpubs/content-applications/content-explorer
Feature Explorer	http://pathfinder.juniper.net/feature-explorer
Learning Bytes	www.juniper.net/learningbytes
Installation and configuration courses	www.juniper.net/courses
J-Net Forum	http://forums.juniper.net/t5/Training-Certification-and/bdp/ Training_and_Certification
Certification program	www.juniper.net/certification
Courses	http://www.juniper.net/training/technical_education
Translation tools	http://www.juniper.net/customers/support/#task

© 2018 Juniper Networks, Inc. All Rights Reserved.

Resources to Help You Learn More

The slide lists online resources available to learn more about Juniper Networks and technology. These resources include the following sites:

- *Pathfinder*: An information experience hub that provides centralized product details.
- *Content Explorer*: Junos OS and ScreenOS software feature information to find the right software release and hardware platform for your network.
- *Feature Explorer*: Technical documentation for Junos OS-based products by product, task, and software release, and downloadable documentation PDFs.
- *Learning Bytes*: Concise tips and instructions on specific features and functions of Juniper technologies.
- *Installation and configuration courses*: Over 60 free Web-based training courses on product installation and configuration (just choose eLearning under Delivery Modality).
- *J-Net Forum*: Training, certification, and career topics to discuss with your peers.
- *Juniper Networks Certification Program*: Complete details on the certification program, including tracks, exam details, promotions, and how to get started.
- *Technical courses*: A complete list of instructor-led, hands-on courses and self-paced, eLearning courses.
- *Translation tools*: Several online translation tools to help simplify migration tasks.

Education Services

Advanced Junos Service Provider Routing

Appendix A: BGP Route Damping

Engineering Simplicity

Objectives

- After successfully completing this content, you will be able to:
  - Explain the causes for route instability
  - Describe the effect of damping on BGP routing
  - Explain the default behavior of damping on links
  - Control damping using routing policy
  - View damped routes using CLI commands

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Will Discuss:

- The causes for route instability;
- The effect of damping on BGP routing;
- The default behavior of damping on links;
- How to control damping using routing policy; and
- How to view damped routes using CLI commands.

Agenda: BGP Route Damping

→Route Flap and Damping Overview
- Route Damping Parameters
- Configuring and Monitoring Route Damping

© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Flap and Damping Overview

The slide lists the topics we will discuss. We discuss the highlighted topic first.

Route Instabilities

- Routes can appear and disappear rapidly
  - Called *route flapping* or just *flapping*
- Rapid sequence of update and withdrawn BGP messages
  - Every BGP router that gets an update or withdrawn message must propagate it to its peers
    - Effects quickly cascade and affect router performance
    - Consumes processing power and bandwidth

© 2020 Juniper Networks, Inc. All Rights Reserved.

Routing Is Dynamic

In any real network, routes can appear and disappear rapidly if a link fails and restores itself repeatedly in a short period of time. This is because any routes (and there could be thousands) that use the failed interface as a next hop must respond to the failure, and the change in next hop must propagate to all other routers on the network. This rapid changing of routing next hops is called *route flapping* or just *flapping* as the link *flaps* up and down.

Update/Withdrawn Pairs

Flapping results in a rapid sequence of BGP update or withdrawn messages. Recall that BGP routers must maintain separate memory tables for inbound and outbound traffic on a per-peer basis. In addition, the BGP routing protocol propagates information on an as-needed basis. These two factors make BGP unstable in the face of a flapping link.

Every BGP router that receives one of these updates or withdrawn messages must send this information to all its BGP router peers. Much like the link-state IGPs of OSPF and IS-IS, BGP must also recalculate its routing tables and databases every time a new update is received. If the new information alters the path selection process, a new route is chosen for the RIB-LOCAL, and the new route must be sent downstream to all BGP peers.

The effect of route flapping quickly cascades and affects router performance. One intermittently failing link can adversely affect a whole network. If this type of update or withdrawal occurs on a very frequent basis, valuable resources in the router, such as processing power and bandwidth, are consumed.

Causes of Route Instability

- Faulty circuits (most common)
- IGP instability
  - Dynamic injection of IGP routes into BGP (static definition and aggregation helps)
- Human error
  - Incorrect routing policy
- Link congestion
  - Overloaded links drop BGP sessions
- In the past, some routers have had problems
  - Software problems (bugs, especially after upgrades)
  - Insufficient power (for example, busy CPU drops BGP sessions)
  - Insufficient memory (tables are kept in memory)
  - Network upgrades and maintenance (adding equipment)

© 2020 Juniper Networks, Inc. All Rights Reserved.

Bad Links

Routes in a network can flap for any number of reasons. Quite possibly, the most frequent reason for a link flap is because of a faulty circuit that is on the brink of outright failure. Any link that rapidly changes from seemingly operational to failing is a potential source of route flapping.

Unstable IGPs

Route flapping is not totally a BGP phenomenon. IGPs that are unstable because of faulty links can affect BGP when IGP routes are injected into BGP for advertising. BGP stability is always desirable and can be enhanced with careful use of static definitions and aggregates instead of injecting raw IGP routes into BGP.

Bad Routing Policy

Human error can cause route flaps as well. An incorrectly configured routing policy, causing routes to be first rejected, then accepted because of a change on the route, can cause flapping.

Congested Links

Link congestion can cause route flaps if the overloaded links drop the BGP sessions that keep the BGP routes *fresh*.

Flapping and History

In the past, sometimes the routers themselves contributed to the flapping problem. Older routers were filled with software bugs (mostly after an upgrade to a new release), had insufficient power (a busy CPU in a software-based router would drop BGP sessions), and often had insufficient memory (routing tables must be kept in memory). Sometimes, just adding equipment for routine network upgrades and maintenance caused route flaps.

BGP Stability Features

- Route damping is the main BGP mechanism to control the route flapping effects
  - Defined in RFC 2439, *BGP Route Flap Damping* (November 1998)
- Damping is ignored on IBGP sessions
- Damping applies to EBGP sessions
  - EBGP sessions can carry thousands of routes
  - EBGP must update or withdraw these routes as required

© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Damping

Because route flapping can be so harmful to BGP, the protocol was extended to support route damping. RFC 2439, *BGP Route Flap Damping* (November 1998), defines route damping. Sometimes the term dampening is seen and used.

No Damping for IBGP

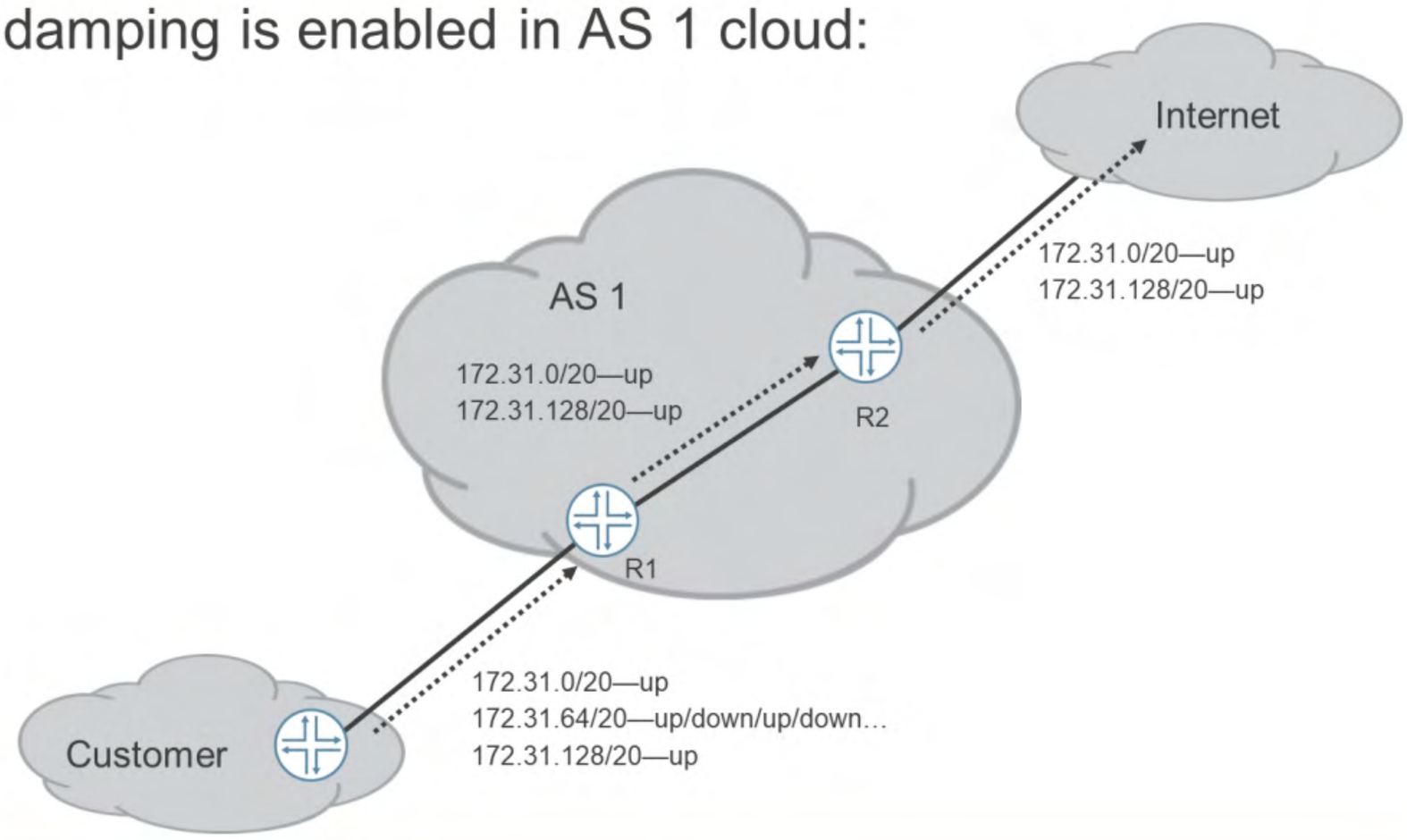
There is a difference between how damping is applied in BGP for internal and external peers. IBGP sessions ignore damping and flap as they please. There is a very good reason for this IBGP behavior. IBGP sessions usually peer to loopbacks, so IGP routes must be able to come and go so that IBGP sessions always have a way to reach the loopback. Recall that BGP has no reachability information of its own and relies on the IGP to resolve next hops.

EBGP Only

Route damping is only applied to routes received from an EBGP peer. EBGP sessions can carry information about thousands of routes. Each EBGP session must update or withdraw these routes as required. Route damping seeks to reward route stability while penalizing route flapping. Once damping is enabled, the router begins to maintain a database of instability. If an EBGP-received route experiences enough flaps, the local BGP process ignores information about that route. This reaction results in *not* including this information in the route selection process and not advertising route changes to downstream BGP peers. Note that some ISPs no longer use damping.

ISPs and Route Damping

- Route damping is enabled in AS 1 cloud:



© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Damping Use by ISPs

The slide shows an example of when BGP route damping is useful.

A customer of AS 1 is connected to the AS by a link running EBGP. The customer advertises three routes: 172.31.0.0/20, 172.31.64.0/20, and 172.31.128.0/20.

AS 1 provides transit service for this customer to the Internet, so the three routes are readvertised within AS 1 and further to the Internet.

However, look what happens when the 172.31.64.0/20 route starts to experience stability problems, causing multiple update and withdraw messages to be sent to AS 1 (up/down/up/down, and so on). Without route damping enabled, this flap action causes the router in AS 1 to send new update messages to other routers in AS 1. These IBGP peers then also send new update messages to their Internet peers.

Enabling route damping can halt this wave of instability at the edge of AS 1. Once enabled, the edge router in AS 1 starts maintaining statistics for the routes received from the customer. Once the 172.31.64.0/20 route is deemed unstable, the AS 1 router stops generating new update messages to its IBGP peers. The IBGP peers, in turn, also have no need to send update messages to the Internet. This makes the Internet, as a whole, more stable.

Agenda: BGP Route Damping

- Route Flap and Damping Overview
- →Route Damping Parameters
- Configuring and Monitoring Route Damping

© 2020 Juniper Networks, Inc. All Rights Reserved.

Route Damping Parameters

The slide highlights the topic we discuss next.

Figure of Merit

- Type of *point value* given to a route that becomes a penalty if it exceeds some value (**`suppress`**)
- New route gets a figure of merit of 0
- Point value of given incidents:
  - 1000 points when route is withdrawn
  - 1000 points when route is readvertised
  - 500 points when the path attribute is changed
- Points decay (reduce) at a certain rate (**`half-life`**)
- If points exceed cutoff (**`suppress`**) threshold, route is suppressed (must be less than or equal to *merit ceiling* value)
- When points fall below certain value (**`reuse`**), route is used again
- Maximum *penalty* imposed by **`max-suppress`** value

© 2020 Juniper Networks, Inc. All Rights Reserved.

Figure of Merit Is a Number

The point at which a route is deemed to be too unstable is calculated by the damping *figure of merit*. It might seem more like a *figure of demerit* but that is the term the RFC uses. In this context, the term *figure* means a *number*, not a picture. The figure of merit is a type of *point value* given to a route. The value becomes a *penalty* if the figure of merit exceeds some predetermined value (that is, the route is *suppressed*). It is often said that damping puts a route into a penalty box for a given period of time.

Default Value = 0

When a previously unknown (that is, new) route arrives at a BGP router that has damping enabled, the new route is assigned a figure of merit value of 0.

Event Point Values

Should the route experience any instability, the figure of merit is incremented according to the following:

- As the EBGP peer withdraws the route, the figure of merit is increased by a value of 1000;
- As the EBGP peer readvertises the route, the figure of merit is increased by a value of 1000; and
- As attributes for the route change through new update messages from the EBGP peer, the figure of merit is increased by a value of 500.

Point Reduction

The points given to a route *decay* (that is, reduce in value) at a certain rate, known as the `half-life`. As long as points decay faster than they accumulate, the route is not suppressed.

The Cutoff Threshold

Should the figure of merit value increase beyond a configured cutoff (`suppress`) threshold, the route is considered unusable, and new information about the route from the EBGP peer is ignored. The suppress value is configurable, but it must be less than or equal to the *merit ceiling* value explained further on this page.

Continued on the next page.

Figure of Merit (contd.)

The Reuse Threshold

The route can once again be considered usable after the figure of merit decreases below a configured threshold. The figure of merit is decreased on a time schedule you set. Should the figure of merit not decrease below the bottom threshold in a configured amount of time, the route can automatically be usable again (`reuse`).

Maximum Suppress Time

The configurable `max-suppress` parameter establishes the maximum time that a route can be suppressed. Also, the figure of merit can only increase to the maximum value, called the *merit ceiling*. The symbol used for the merit ceiling is ε_c. This maximum value is calculated from a combination of the components listed above and is determined by the following formula:

$$\varepsilon_c \leq \varepsilon_r \, e^{(t/\lambda)(\ln 2)}$$

where ε_r is the figure of merit reuse threshold, t is the maximum suppression (hold-down) time in minutes, and λ is the half-life in minutes.

Damping Parameters

- **`suppress`**: Cutoff (suppression) threshold
  - Value of *penalty* when damping is initiated
    - Default value 3000 (if changed, must be less than or equal to merit ceiling value ε_c or no damping occurs)
- **`reuse`**: Reuse threshold
  - Value to which *penalty* must decay for route to be used again
    - Default threshold is 750
- **`half-life`**: Decay half-life, in minutes
  - Time it takes to reduce damping *penalty* by half
    - Default is 15 minutes
- **`max-suppress`**: Maximum hold-down time, in minutes
  - Longest time to suppress the route activity
    - Default is 60 minutes

© 2020 Juniper Networks, Inc. All Rights Reserved.

Cutoff Threshold

The figure-of-merit value interacts with the damping parameters. The slide lists these parameters.

The **`suppress`** variable is the configured threshold where a BGP route is considered unstable and is not used. This variable represents the value of the penalty that establishes the point at which damping is initiated. When this value is reached, the route is cutoff (suppressed). The default value of **`suppress`** is 3000. Possible values range from 1–20000. If changed, this value must be less than or equal to the merit ceiling ε_c, or damping never occurs.

Reuse Threshold

The **`reuse`** variable is the configured threshold where a BGP route is considered usable once again. This variable is the value to which the penalty must decay before the router considers the route in its path selection. The default value of the **`reuse`** is 750. Possible values range from 1–20000.

Decay Half-Life

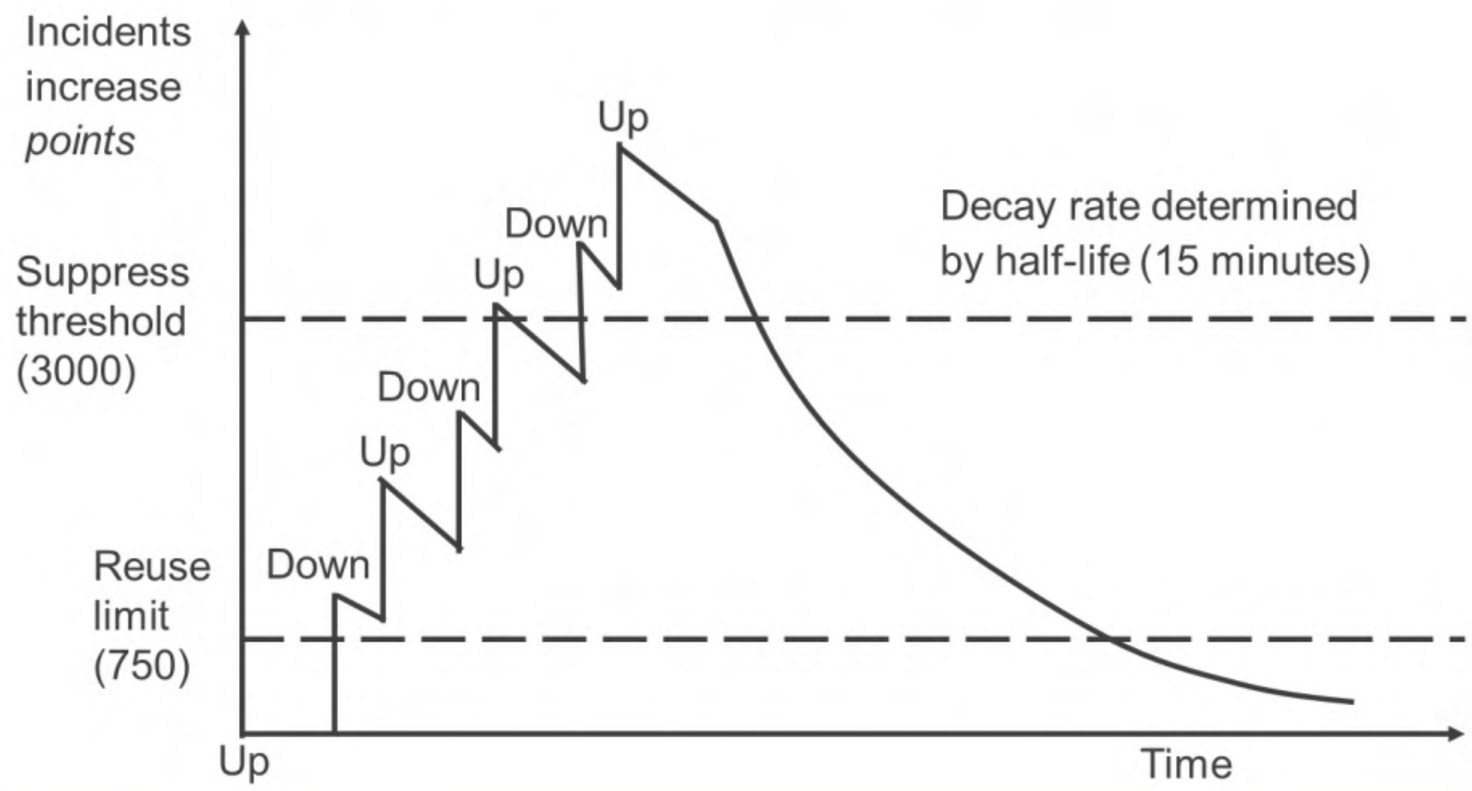
The **`half-life`** variable is the rate at which the figure of merit is decreased to half its value once the value is larger than 0. The default value of the **`half-life`** is 15 minutes. Possible values range from 1–45 minutes.

Maximum Hold-Down Time

The **`max-suppress`** variable is the configured maximum amount of time that a BGP route can be deemed unusable. This variable is the longest time that the route can be suppressed until the route is *given another chance to behave*. The default value of the **`max-suppress`** is 60 minutes. Possible values range from 1–720 minutes. At the end of the **`max-suppress`** interval, all is forgiven and the route becomes active again.

Figure of Merit Figure

- Exponential decay, and there is a fixed ceiling



© 2020 Juniper Networks, Inc. All Rights Reserved.

Sample Figure of Merit in Action

The slide shows a graphic representation of the figure of merit in use for a BGP route. The default values for **`suppress`** (3000), **`reuse`** (750), and **`half-life`** (15 minutes) are used. Note that an *exponential decay* occurs on the figure of merit, and a fixed ceiling exists on the figure-of-merit value (the merit ceiling).

After receiving a new BGP route, the figure of merit is 0 for some period of time. As soon as the route is withdrawn (or the link is down), the figure of merit increments to 1000. As long as the route stays down, the figure of merit decays somewhat. As the route is readvertised, the figure of merit is incremented by another 1000. Again, the figure of merit starts to decay. Now the route is withdrawn a second time, and again, the figure of merit is increased. Now, when the route is readvertised, the figure of merit is increased by another 1000. This time, because not enough time has elapsed between these events in this example, the route is over the **`suppress`** limit of 3000 and is considered unusable. In short order, the route is withdrawn and readvertised, yet again. Each time, the figure of merit increases 1000 for each action. Notice that the route is still damped and considered unusable, but the figure of merit still increases and decreases, even while the route is suppressed.

Whenever the figure of merit is greater than 0, the value is constantly and consistently decayed using the configured **`half-life`** value. The **`half-life`** is always configured in increments of minutes. The purpose of the **`half-life`** is to decay exponentially the figure of merit such that the value from any point in time is reduced by half at the end of the configured **`half-life`** (this half-life behavior is the essence of an exponential decay). The decay is an exponential process to allow individual routes to become active once their figure of merit falls below the **`reuse`** threshold. This prevents large numbers of routes becoming active at the same time. This has the effect of not overloading BGP routers with large amounts of updates at once, causing further route damping.

Agenda: BGP Route Damping

- Route Flap and Damping Overview
- Route Damping Parameters
- →Configuring and Monitoring Route Damping

© 2020 Juniper Networks, Inc. All Rights Reserved.

Configuring and Monitoring Route Damping

The slide highlights the topic we discuss next.

Configuring Damping

- Damping parameters are defined within `[policy options]` hierarchy
- Can use as a policy action
  - **`half-life`**: 1–45 minutes, default 15
  - **`max-suppress`**: 1–720 minutes, default 60
  - **`reuse`**: 1–20000, default 750
  - **`suppress`**: 1–20000, default 3000
  - **`disable`** = *do not damp*

```
[edit policy-options]
damping name {
    half-life minutes;
    max-suppress minutes;
    reuse number;
    suppress number;
    disable;
}
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Damping Parameters in a Routing Policy

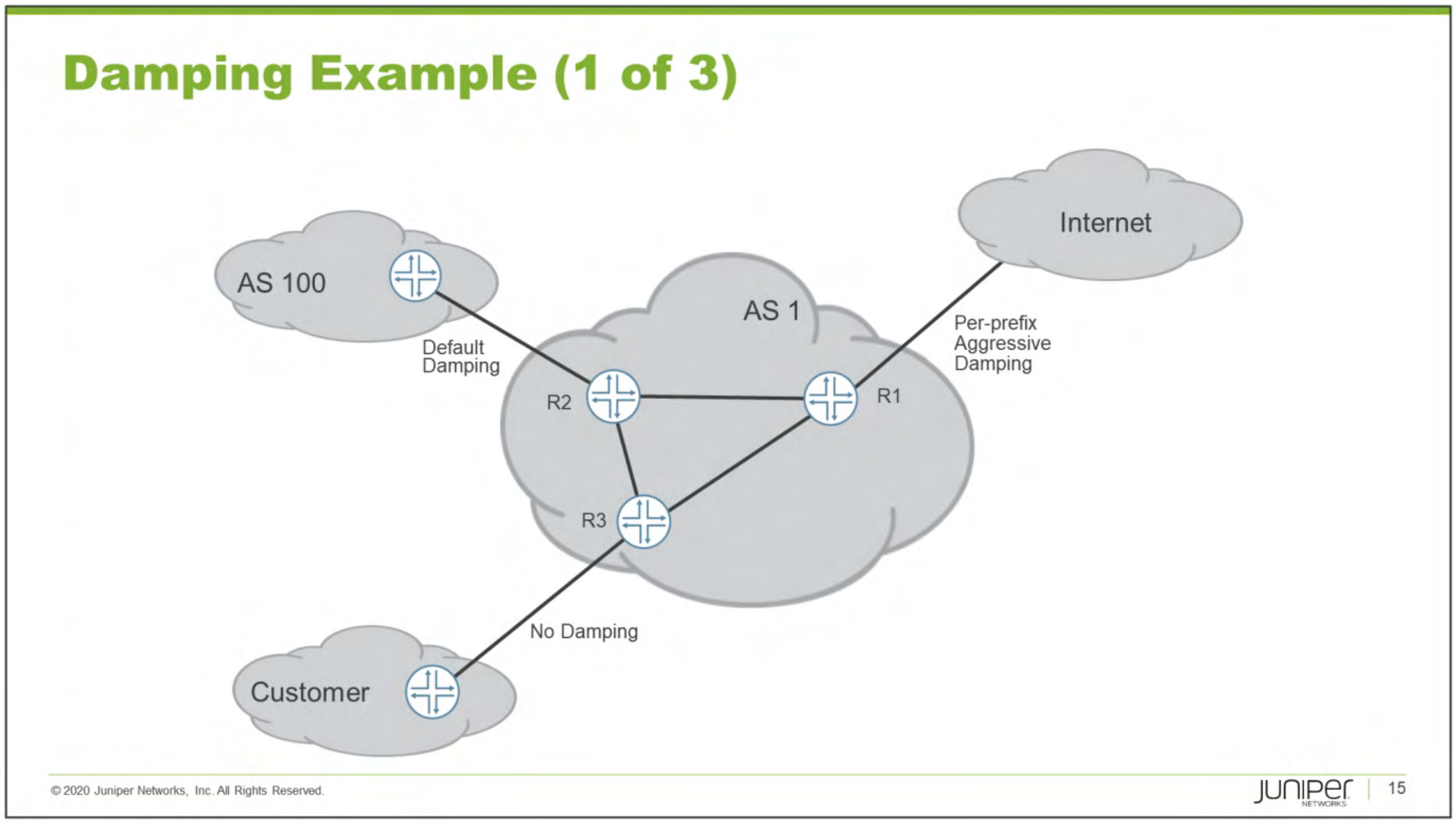
Once damping is enabled within the BGP portion of the Junos OS configuration, the default values introduced on previous slides are used for figure-of-merit calculations.

To alter these default values, you can create and define a damping profile within the `[edit policy-options]` configuration hierarchy.

Applied as Policy Action

Much like the AS-path and community attributes, you name and define a damping profile first. Then you can use it within a policy as an action. The slide shows the five damping parameters.

The presence of the **`disable`** keyword deserves a few words of explanation. You can use the keyword **`disable`** within a damping profile to *not* have the figure of merit be calculated for certain routes. This is often useful to exempt certain routes that should never be damped and made unusable. One good example of these types of routes are the root DNS servers in the Internet. If these servers become unreachable because of damping, the ISP and its customers experience DNS lookup failures. For example, DNS routes could have a damping profile of *`no-damping`* defined that contains a single statement: **`disable`**.



Example of Damping Use, Part 1

The slide shows a fairly sophisticated example of route damping in action.

On the slide, AS 1 wants to enable damping for all its EBGP peers, based on the following administrative decisions:

- AS 1 wants to operate the default damping values for routes from AS 100;
- AS 1 does *not* want to damp any routes from its customer; and
- AS 1 wants to damp all routes aggressively from the Internet except for certain prefixes.

The next few slides in this sequence examine how you can implement these damping policies.

The next slide outlines the routing policies you can use to accomplish these goals. We detail the application of these routing policies on a later slide.

Damping Example (2 of 3)

```
[edit policy-options]
policy-statement customer-inbound {
    then damping do-not-damp;
}
policy-statement internet-inbound {
    term let-some-through {
        from {
            route-filter 192.168.10.0/24 orlonger;
            route-filter 172.16.240.0/24 orlonger;
            route-filter 172.27.32.64/26 orlonger;
            route-filter 192.168.100.0/24 orlonger;
        }
        then {
            damping do-not-damp;
            accept;
        }
    }
    term damp-all-others {
        then damping aggressive-damp;
    }
}
damping do-not-damp {
    disable;
}
damping aggressive-damp {
    half-life 30;
    reuse 200;
    suppress 1500;
    max-suppress 120;
}
```

Damping profiles

© 2020 Juniper Networks, Inc. All Rights Reserved.

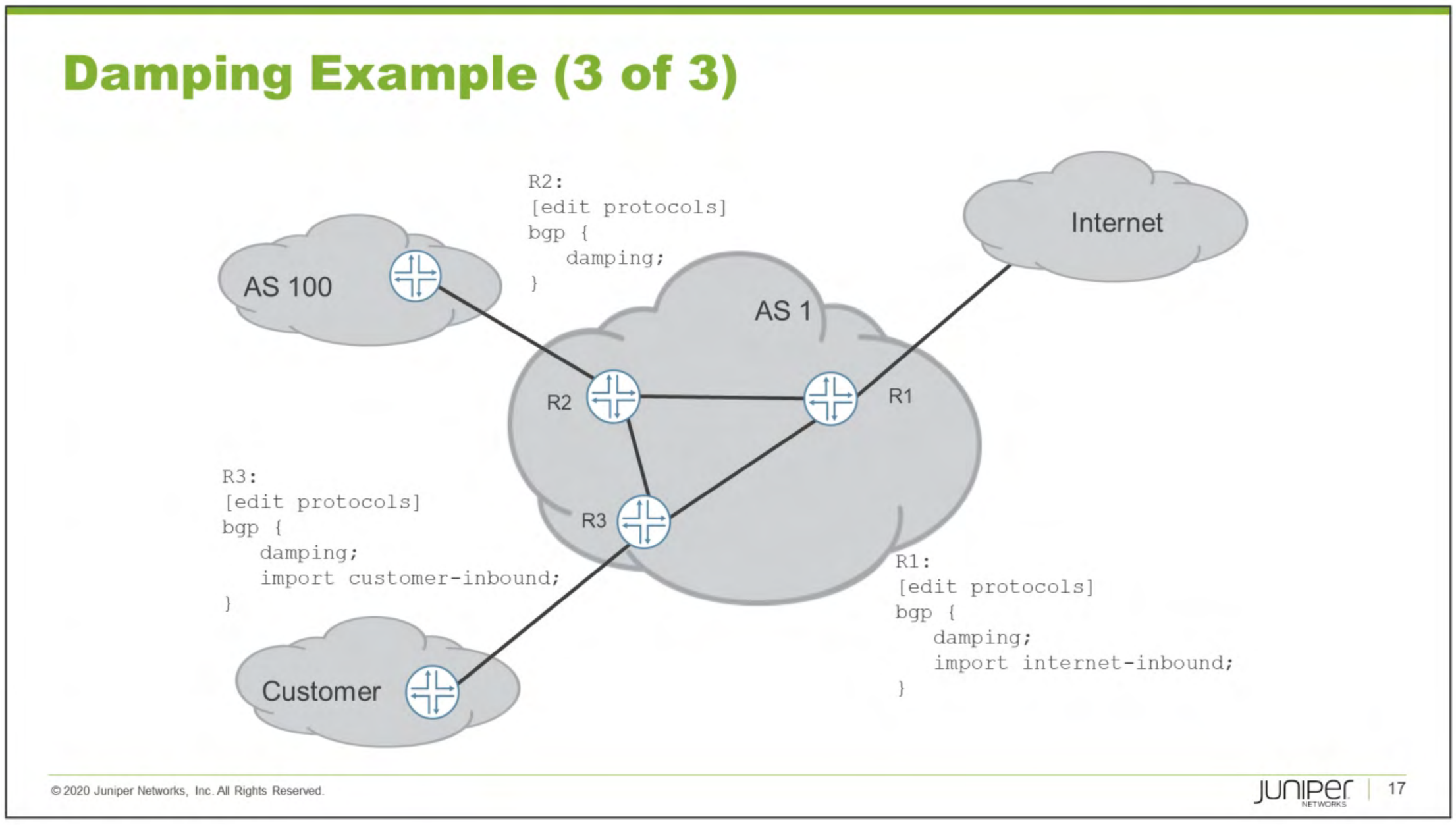
Example of Damping Use, Part 2

This slide shows all the damping profiles and policies to be used in AS 1 in the damping example. The profile *do-not-damp* has a variable of **disable** defined. The profile *aggressive-damp* has defined four variables as follows:

- **suppress** is 1500;
- **reuse** is 200;
- **half-life** is 30; and
- **max-suppress** is 120.

A policy named *customer-inbound* is defined with no `from` statement, so all possible routes match the policy. The policy has an action of damping *do-not-damp*. This action sets the profile of *do-not-damp* to all routes.

A policy named *internet-inbound* is defined with two terms. The *let-some-through* term has several `route-filter` statements with a `then` action defined of `damping` *do-not-damp* followed by an `accept` action. A second term of *damp-all-others* has no `from` statement defined, so all routes are subjected to the `damping` *aggressive-damp* action.



Example of Damping Use, Part 3

The routers in AS 1 are next updated to apply the policies we created on the previous slide.

Router R1 defines damping in BGP and an import policy of *internet-inbound*. This configuration enables damping on the router and applies profile parameters as per the policy. As mentioned previously, the currently configured policy contains a logic flaw that causes it not to meet the administrative requirements.

Router R2 simply defines damping within its BGP configuration. This configuration both enables damping and operates with the default parameters on routes from AS 100. No policy is needed on this router.

Router R3 defines damping within BGP and an import policy of *customer-inbound*. This configuration enables damping on the router and applies profile parameters as per the policy.

Viewing Withdrawn Routes

- **`show route damping history`** displays withdrawn routes having a figure of merit:

```
user@router> show route damping history extensive
inet.0: 15 destinations, 15 routes (14 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

200.200.200.0/24 (1 entry, 0 announced)
         BGP     Preference: /-101
                 Nexthop: 172.16.10.1 via fe-0/0/0.0, selected
                 State: <Hidden Ext>
                 Local AS:     2  Peer AS:     1
                 AS path: 1 I
                 Localpref: 100
                 Router ID: 192.168.1.1
                 Merit (last update/now): 2777/2454
                 Default damping parameters used
                 Last update: 00:02:45  First update: 00:04:35
                 Flaps: 3
                 History entry.  Expires in:       00:54:20
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Damping History

The slide shows the output from the **`show route damping history`** command.

Any routes displayed by this command were withdrawn from the router. However, the router retains a record of these routes should they be readvertised to the local router. Some notable details in the display include the following:

- The route is currently hidden. We see this in both the `State: <Hidden Ext>` field as well as the `Preference: /-101` field. Notice that no Junos OS protocol preference value is defined.
- There is a field (`Merit`) for the current figure-of-merit value. The two values that follow list the value after the last BGP update (or withdrawal), and the current value after experiencing some decay. For this route, the values are `Merit: 2777/2454`. Thus, the value at the last update/withdrawal was 2777 (note that this value need not necessarily exceed the default suppress threshold of 3000), and the current value is 2454.
- The default parameters are used (`Default damping parameters used`). If this route were evaluated by a policy with a damping action, the new damping profile name would appear in the output.

Viewing Decayed Routes

- **`show route damping decayed`** displays active routes having a figure of merit:

```
user@router> show route damping decayed extensive
inet.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200.200.200.0/24 (1 entry, 1 announced)
        *BGP     Preference: 170/-101
                 Nexthop: 172.16.10.1 via fe-0/0/0.0, selected
                 State: <Active Ext>
                 Local AS:     2  Peer AS:     1
                 AS path: 1 I
                 Localpref: 100
                 Router ID: 192.168.1.1
                 Merit (last update/now): 2000/1954
                 Default damping parameters used
                 Last update: 00:00:35   First update: 00:00:40
                 Flaps: 2
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Routes with Non-0 Figures of Merit

The slide shows the output from the **`show route damping decayed`** command.

Any routes displayed by this command were advertised to the router and are currently usable routes, but these routes have a figure of merit greater than 0. Some things to note in the display are:

- The route is currently active. We see this both by the asterisk (`*`) in the output as well as the `State: <Active Ext>` field.
- There is a field (`Merit`) for the current figure-of-merit value. The two values list the value after the last update (or withdrawal) and the current value after experiencing some decay. For this route, the values are `Merit: 2000/1954`.
- The default parameters are used (`Default damping parameters used`). If a policy with a damping action evaluated this route, the new damping profile name would appear in the output.

Viewing Suppressed Routes

- **`show route damping suppressed`** displays routes that were damped because of the figure of merit
- **`clear bgp damping`**
  - Suppressed routes can have the figure of merit reduced to 0
  - All routes (as well as individual routes) can be cleared

```
user@router> show route damping suppressed
inet.0: 15 destinations, 15 routes (14 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

200.200.200.0/24    [BGP] 00:01:21, localpref 100
                      AS path: 1 I
                    > to 172.16.10.1 via fe-0/0/0.0

user@router> clear bgp damping
inet.0: 15 destinations, 15 routes (15 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

© 2020 Juniper Networks, Inc. All Rights Reserved.

Damped Routes

The slide shows the output from the **`show route damping suppressed`** command.

Any routes displayed by this command were advertised to the router, but these routes have a figure of merit that is currently above the **`suppress`** threshold, and the route is unusable.

Manual Clearing

The route remains in this state until the figure of merit crosses below the **`reuse`** threshold. By using the **`clear bgp damping`** command, a route can have the figure of merit reduced to 0 administratively

On the slide, the route 200.200.200.0/24 is currently suppressed and hidden. After we issue the **`clear bgp damping`** command, the route is no longer hidden.

Summary

- In this content, we:
  - Explained the causes for route instability
  - Described the effect of damping on BGP routing
  - Explained the default behavior of damping on links
  - Controlled damping using the routing policy framework
  - Viewed damped routes using CLI commands

© 2020 Juniper Networks, Inc. All Rights Reserved.

We Discussed:

- The causes of route instability;
- The effect that route damping has on BGP routing;
- The default behavior of damping on links;
- Controlling damping using the routing policy framework; and
- Viewing damped routes using CLI commands.

Review Questions

1. Why is damping ignored on IBGP routes?
2. What is the half-life as this term applies to route damping?
3. What is the function of the **`max-suppress`** parameter?
4. What happens if the suppress threshold is set higher than the merit ceiling?
5. What kind of routes are shown with the **`show route damping decayed`** command?

© 2020 Juniper Networks, Inc. All Rights Reserved.

Review Questions

1.

2.

3.

4.

5.

Lab: BGP Route Damping

- Configure and monitor route damping

© 2020 Juniper Networks, Inc. All Rights Reserved.

Lab: BGP Route Damping

The slide provides the objective for this lab.

Answers to Review Questions

1.

IBGP sessions usually peer to loopbacks, so IGP routes must be able to come and go so that IBGP sessions always have a way to reach the loopback.

2.

The half-life is the rate at which the figure of merit is decreased to half its value once the value is larger than 0. The default value of the half-life is 15 minutes.

3.

The max-suppress parameter establishes the maximum time that a route can be suppressed. The default value is 60 minutes.

4.

If the suppress threshold is set higher than the merit ceiling no damping will occur.

5.

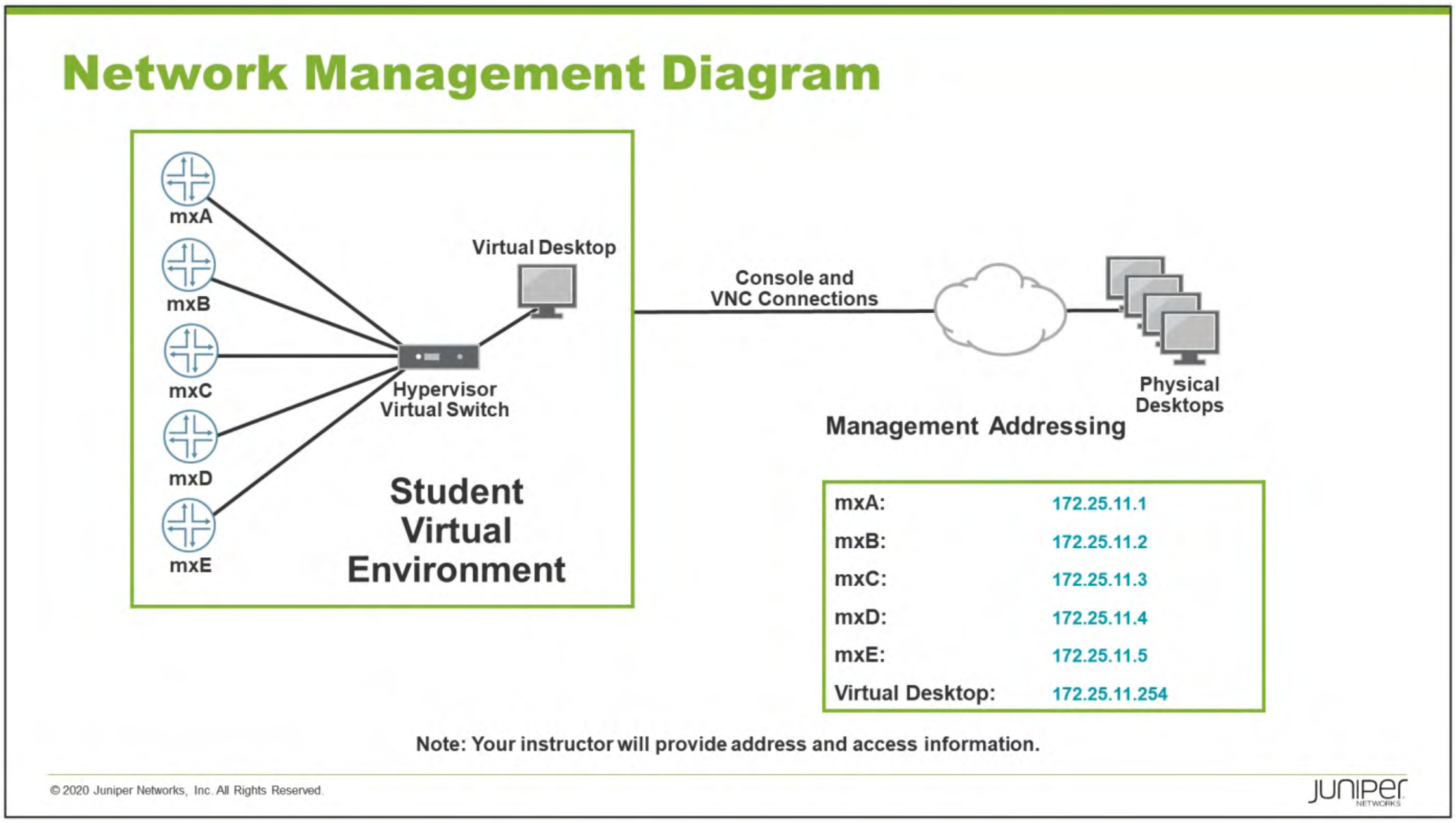
The command shows any routes that were advertised to the router and are currently usable routes, but have a figure of merit greater than 0.

Education Services

Advanced Junos Service Provider Routing

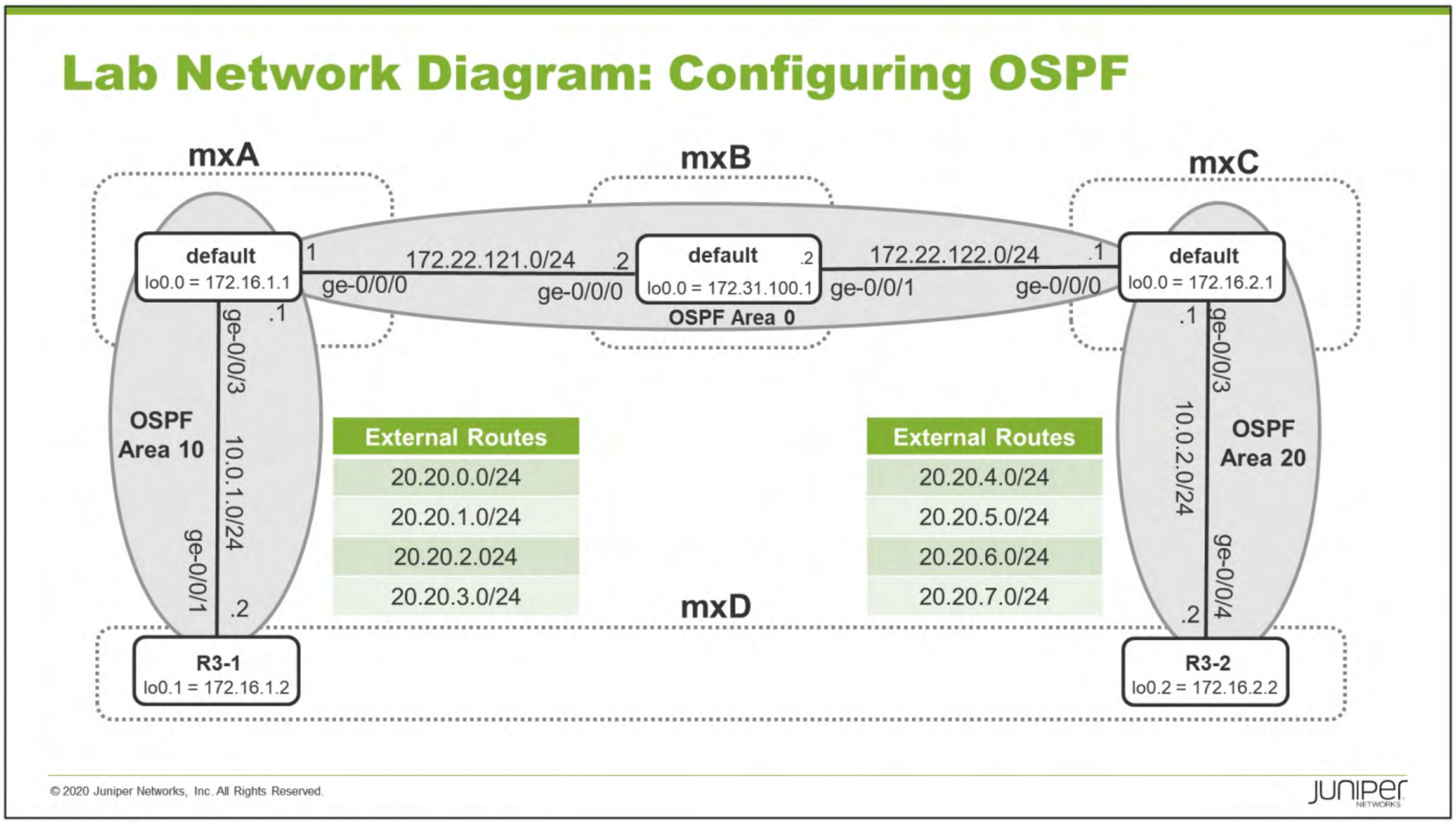
Appendix B: Lab Diagrams

Engineering Simplicity



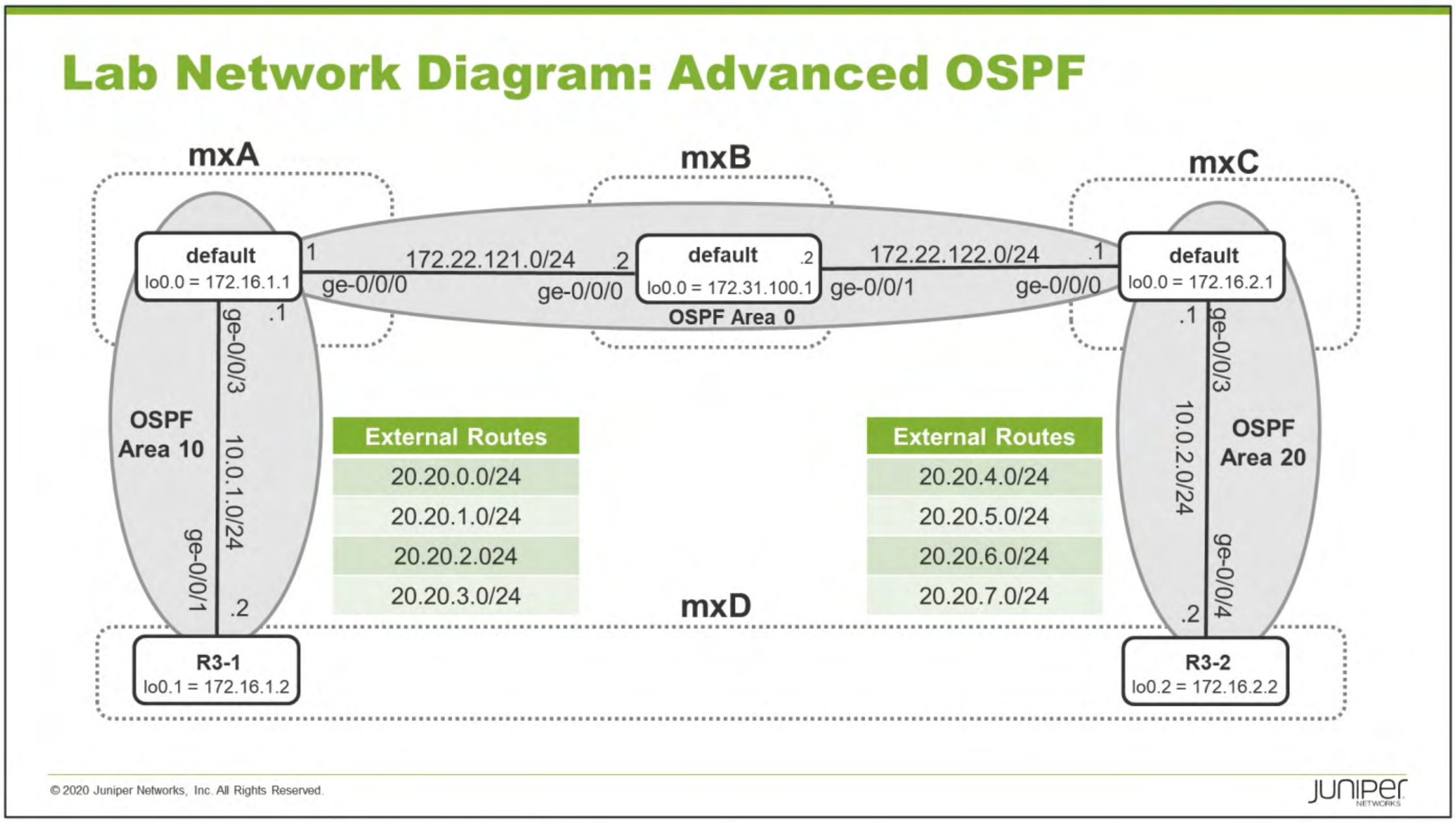
Lab Diagram: Network Management

The slide diagram is referenced in the lab.



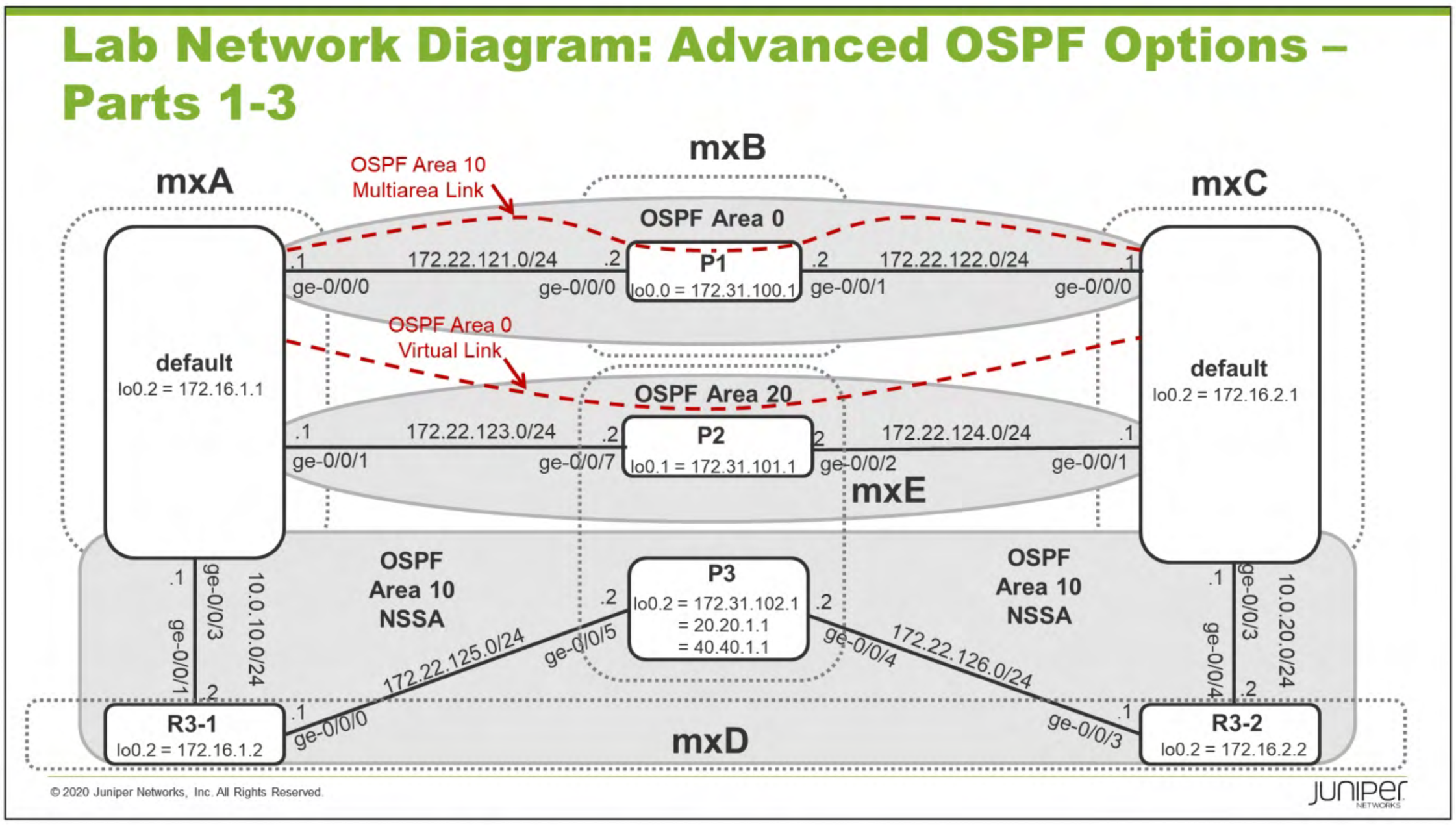
Lab Network Diagram: Configuring OSPF

The slide diagram is referenced in the lab.



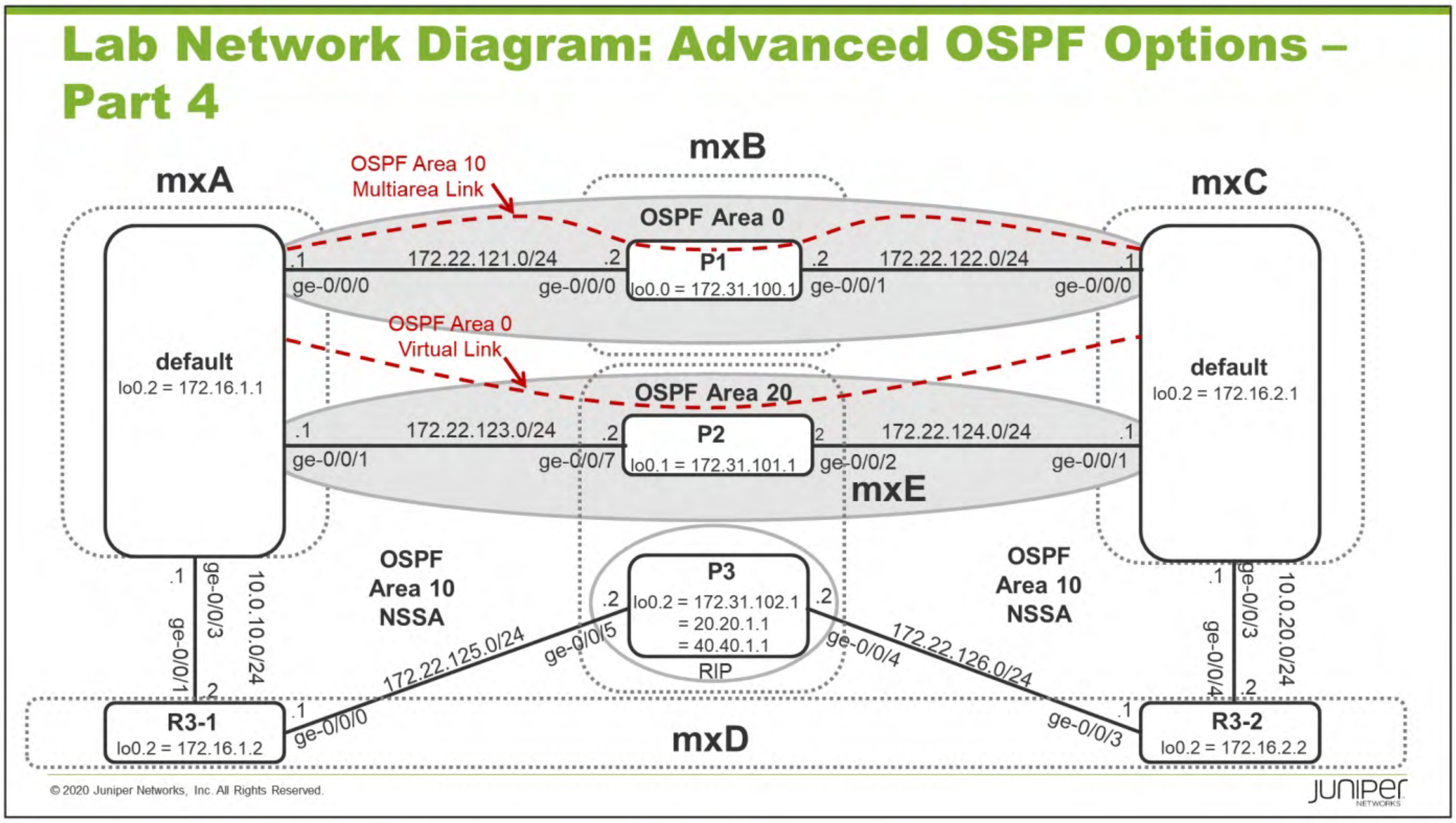
Lab Network Diagram: Advanced OSPF

The slide diagram is referenced in the lab.



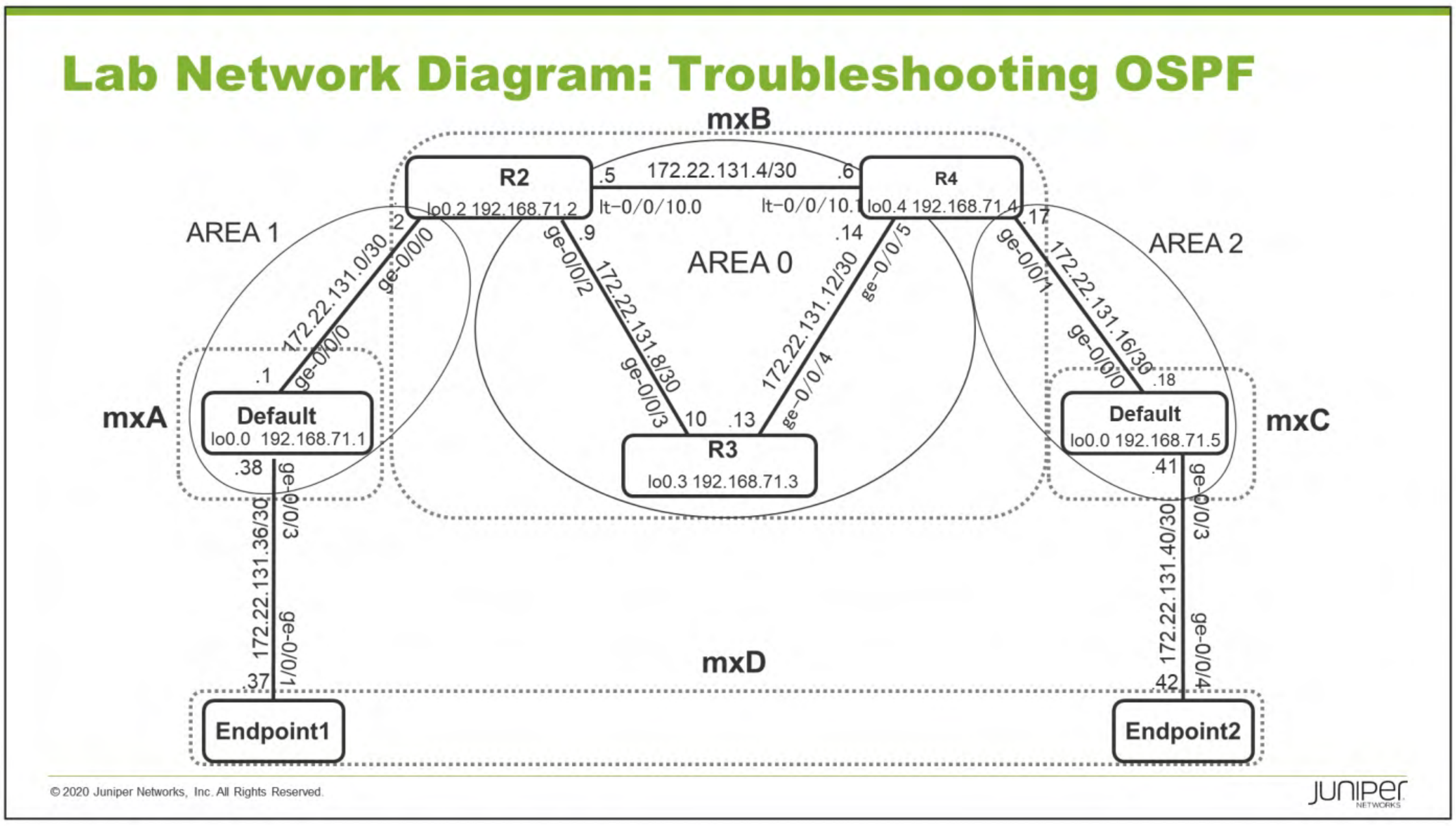
Lab Network Diagram: Advanced OSPF Options – Parts 1-3

The slide diagram is referenced in the lab.



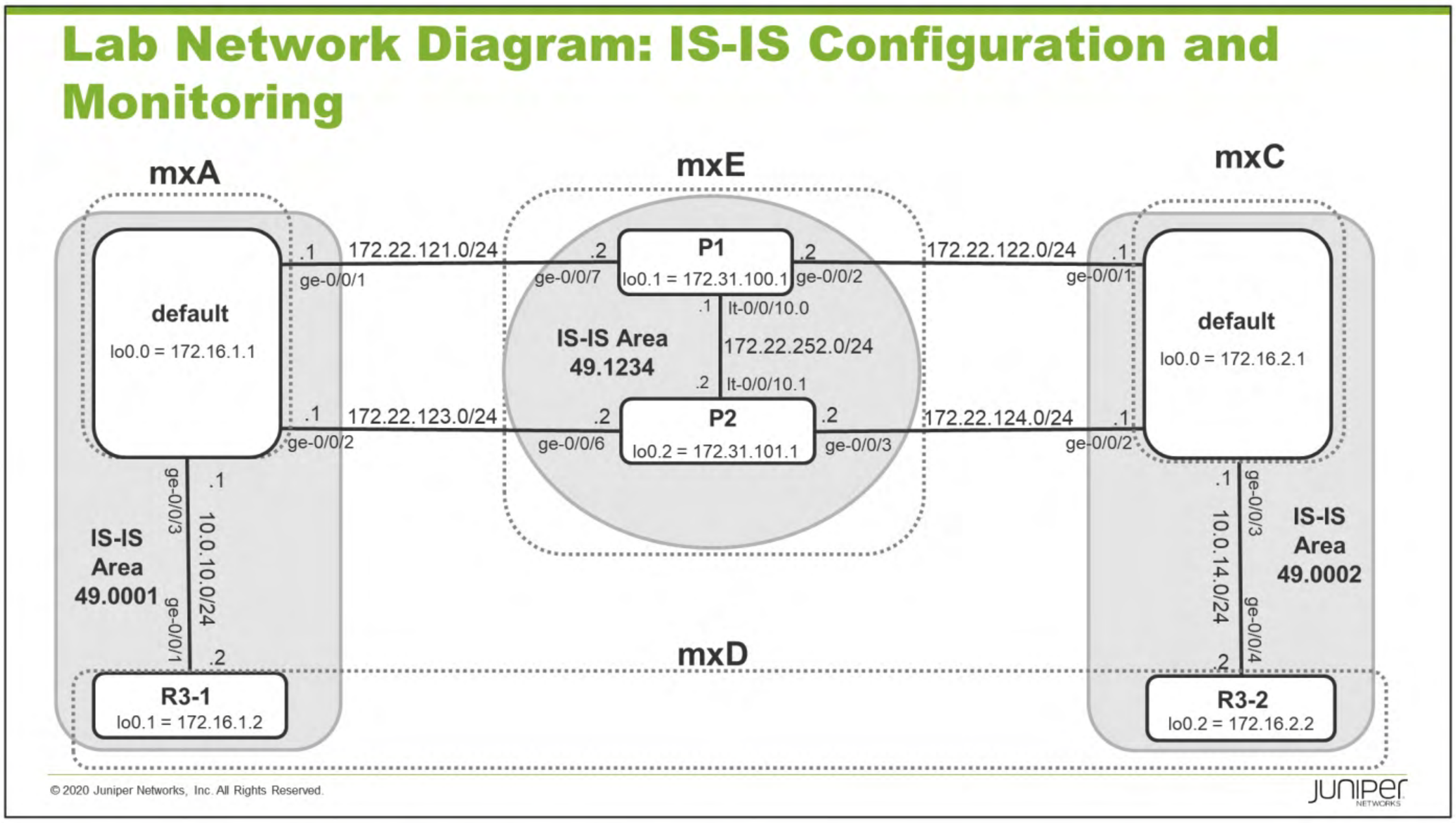
Lab Network Diagram: Advanced OSPF Options – Part 4

The slide diagram is referenced in the lab.



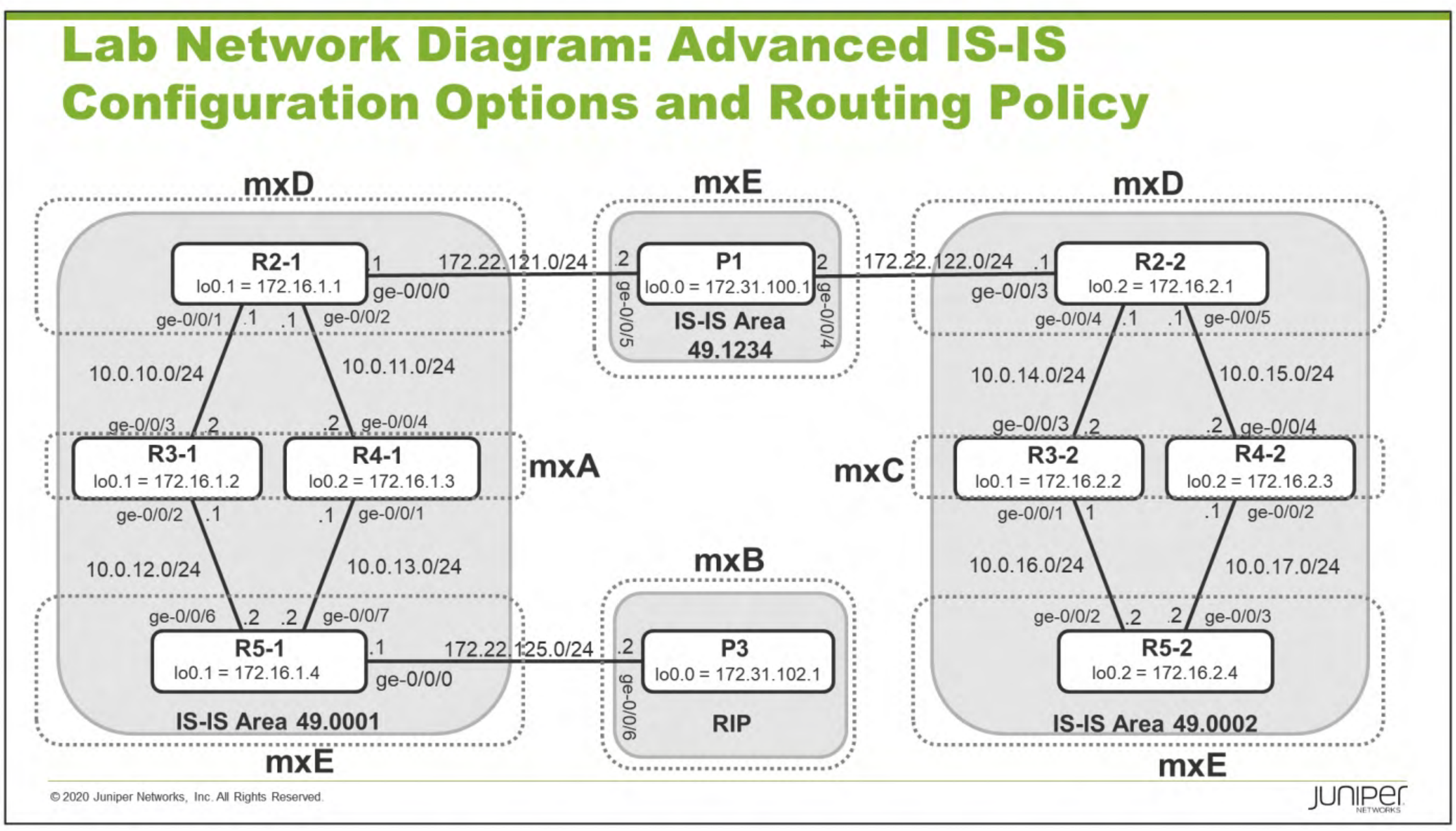
Lab Network Diagram: Troubleshooting OSPF

The slide diagram is referenced in the lab.



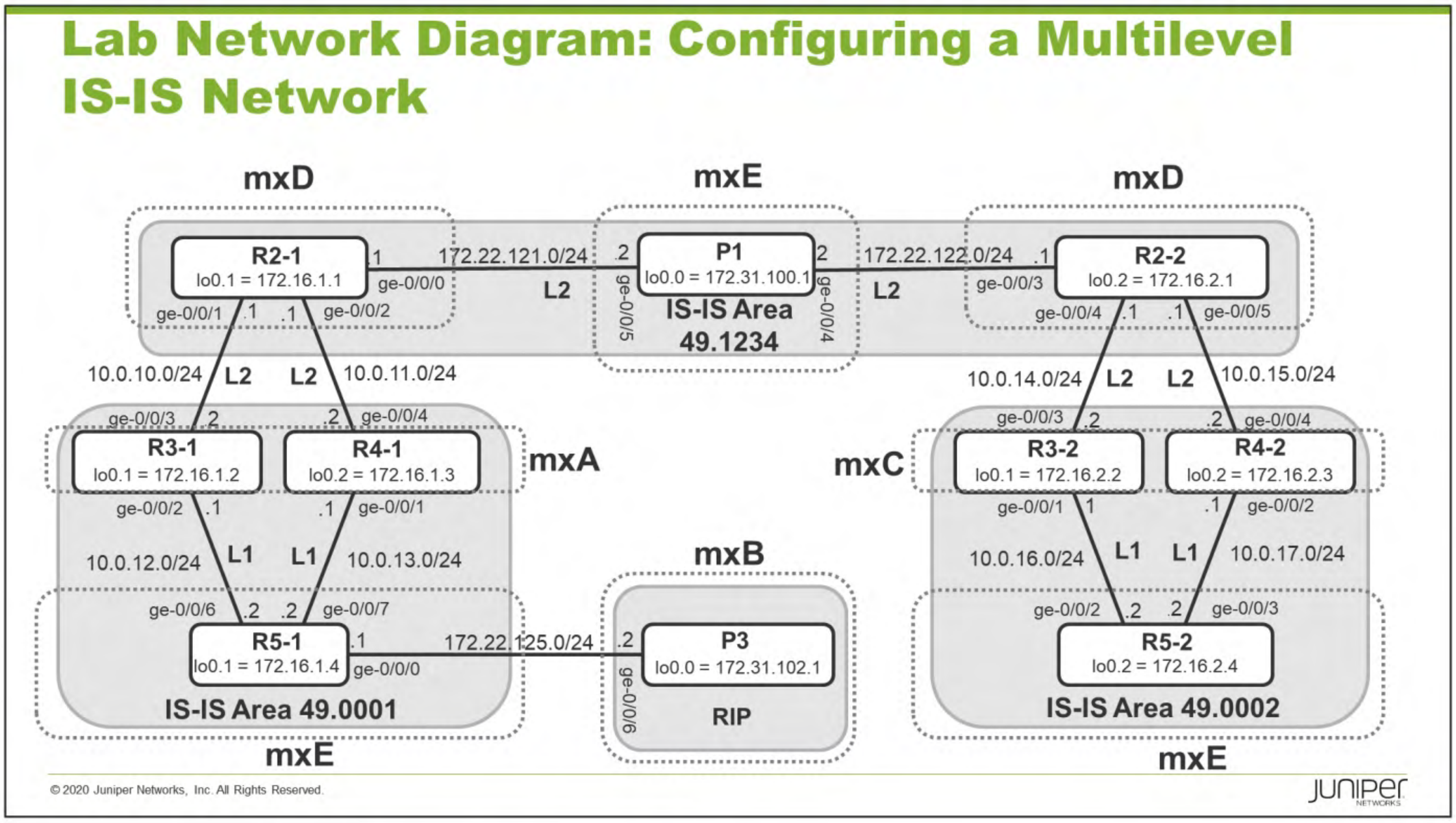
Lab Network Diagram: IS-IS Configuration and Monitoring

The slide diagram is referenced in the lab.



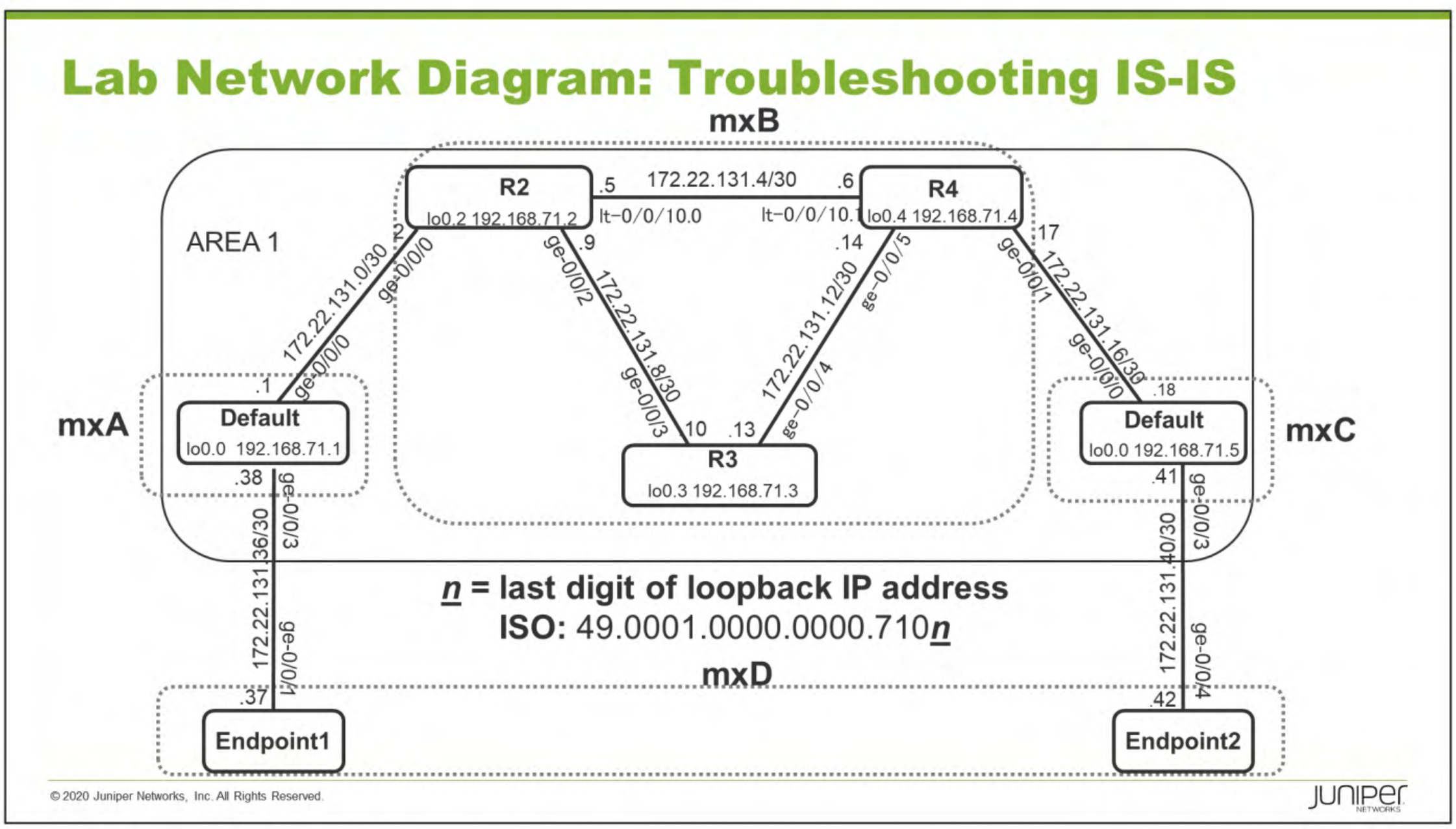
Lab Network Diagram: Advanced IS-IS Configuration Options and Routing Policy

The slide diagram is referenced in the lab.



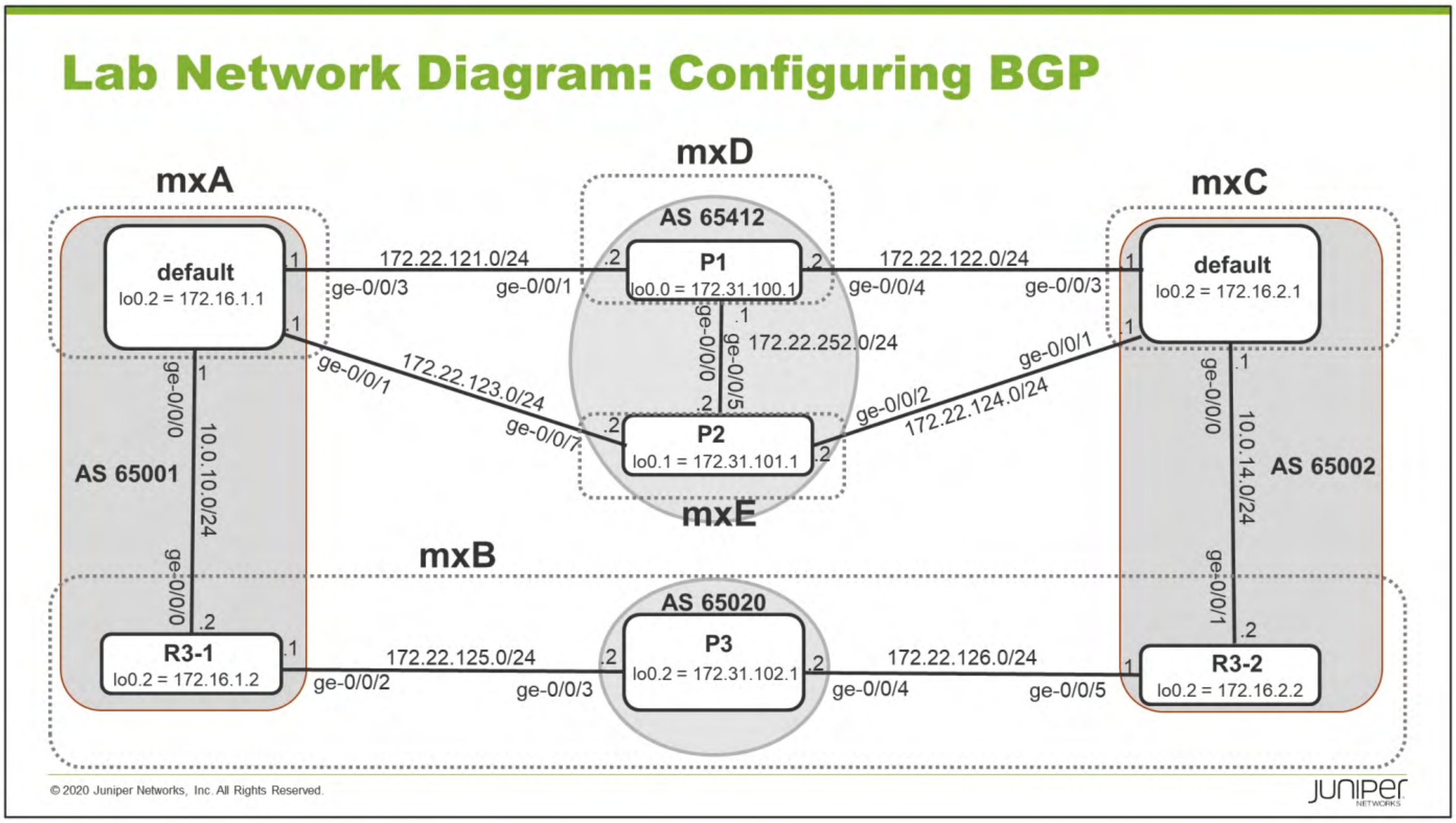
Lab Network Diagram: Configuring a Multilevel IS-IS Network

The slide diagram is referenced in the lab.



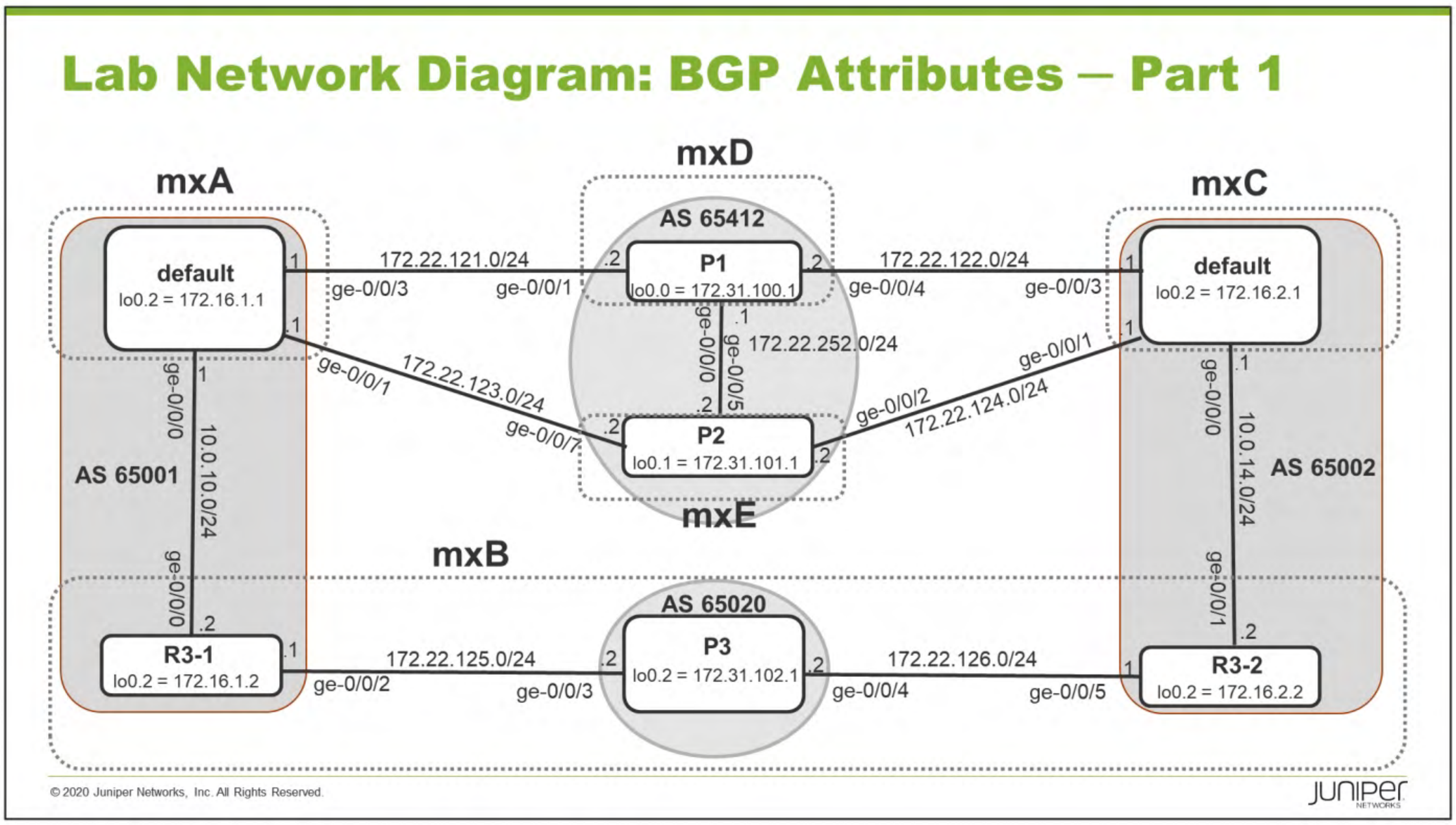
Lab Network Diagram: Troubleshooting IS-IS

The slide diagram is referenced in the lab.



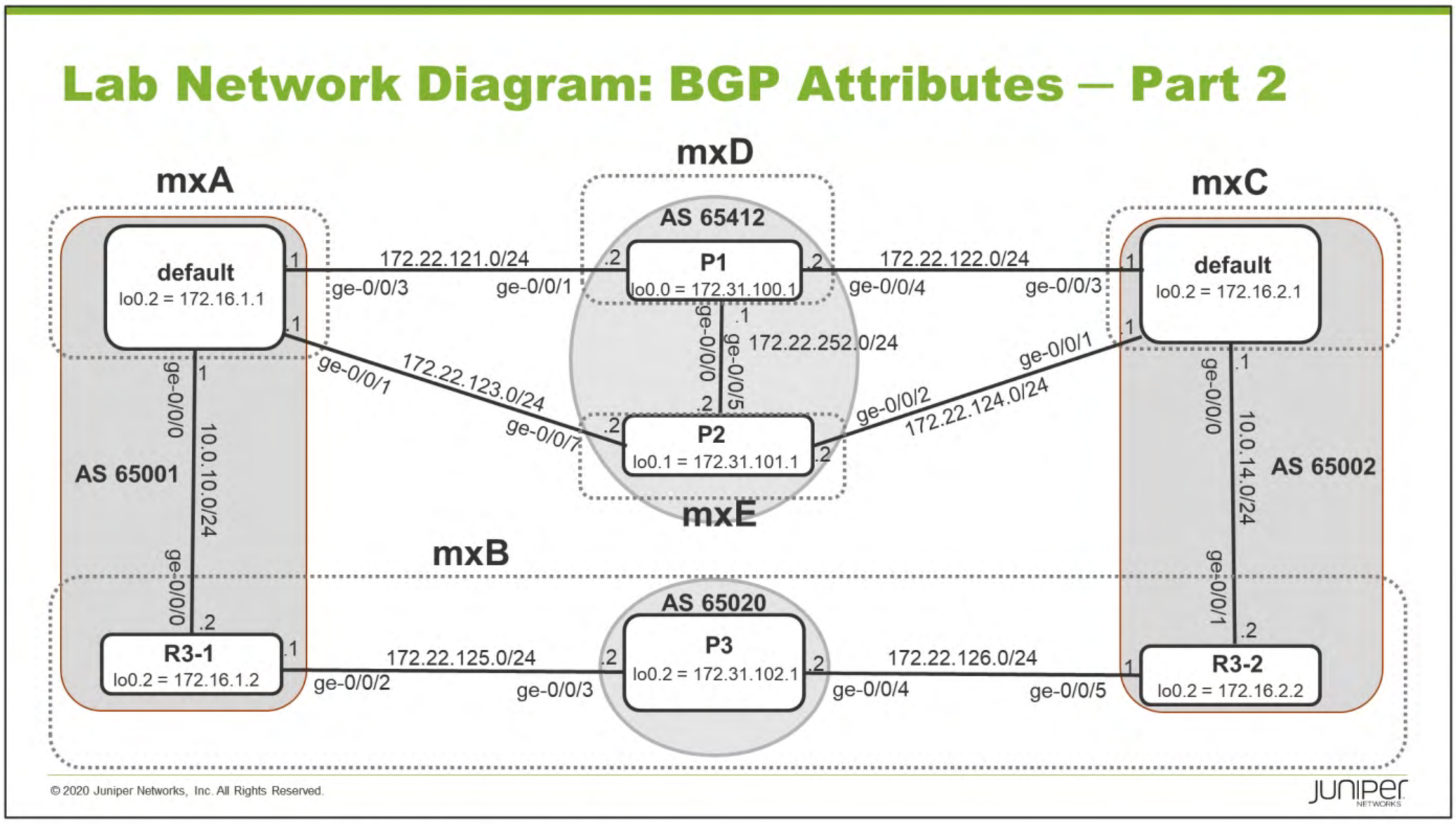
Lab Network Diagram: Configuring BGP

The slide diagram is referenced in the lab.



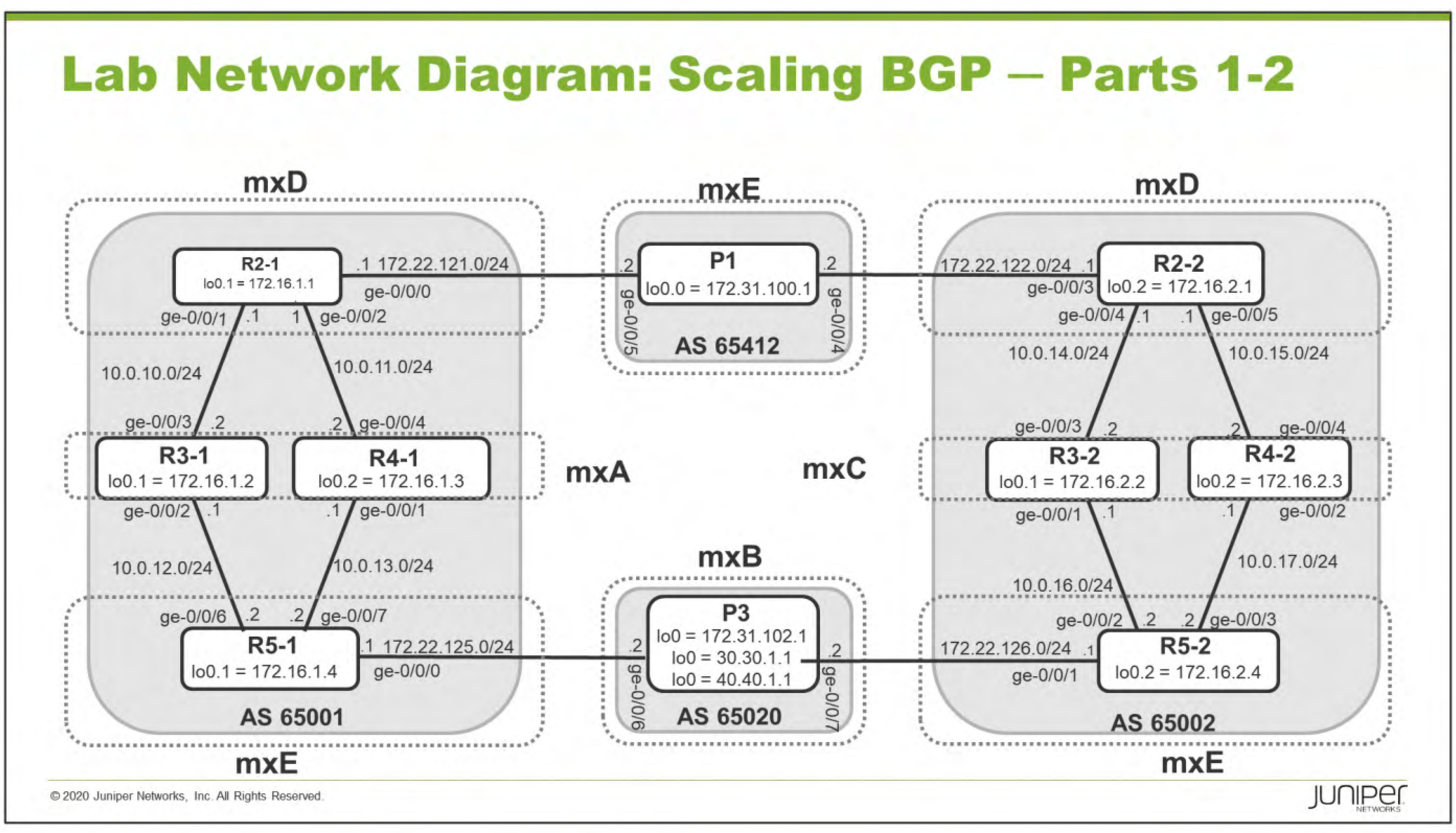
Lab Network Diagram: BGP Attributes — Part 1

The slide diagram is referenced in the lab.



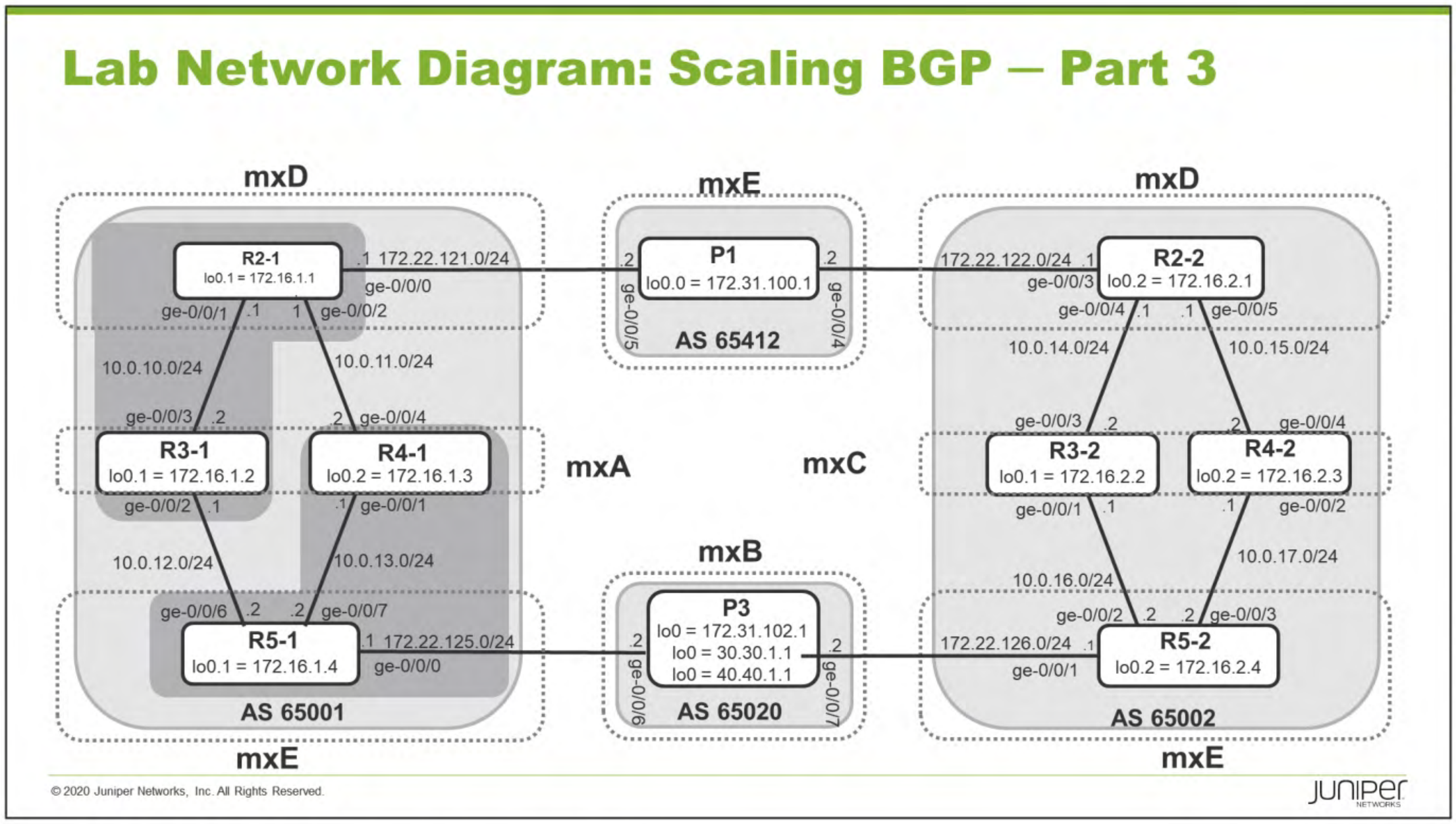
Lab Network Diagram: BGP Attributes — Part 2

The slide diagram is referenced in the lab.



Lab Network Diagram: Scaling BGP — Parts 1-2

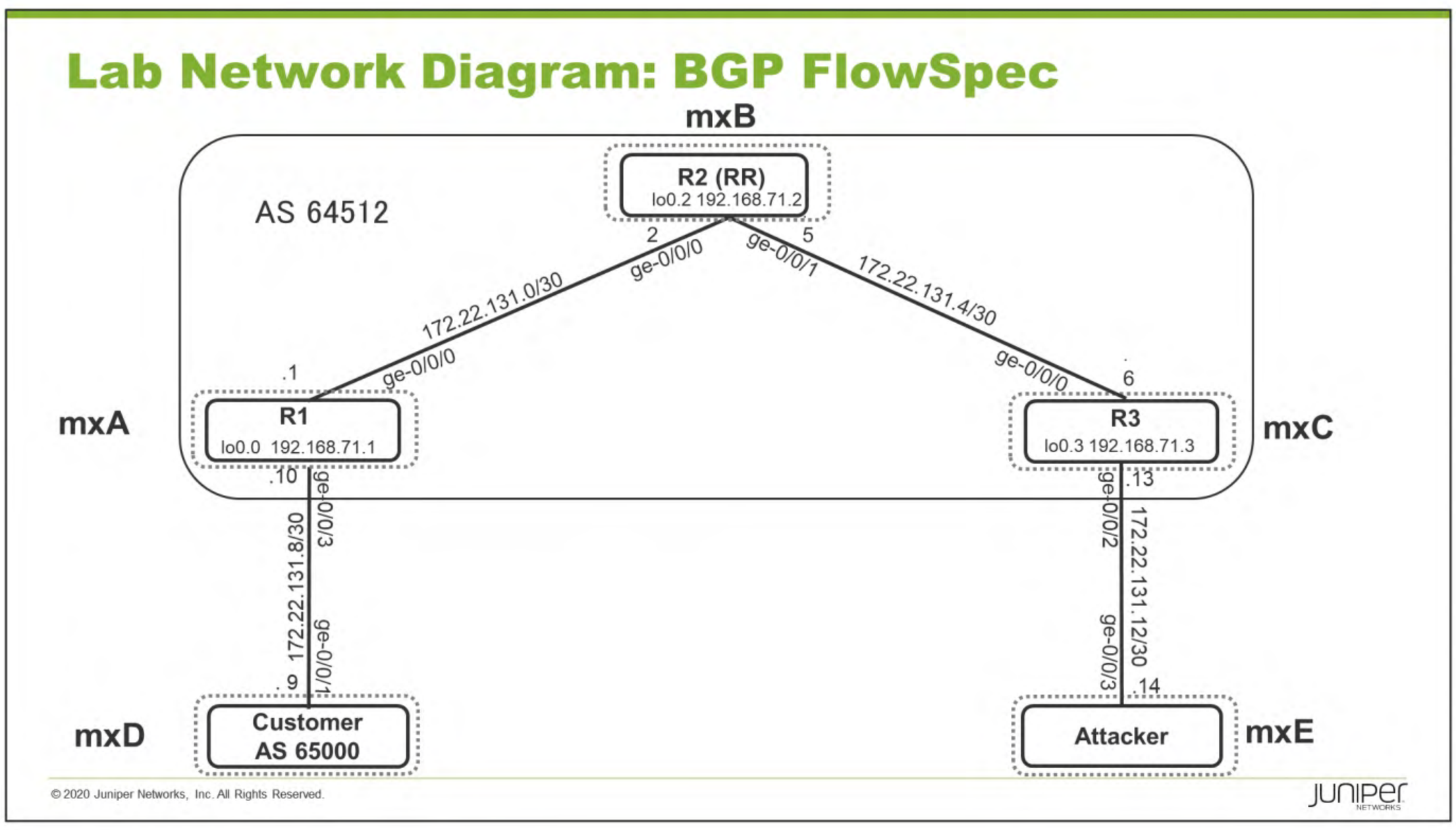
The slide diagram is referenced in the lab.



Lab Network Diagram: Scaling BGP — Part 3

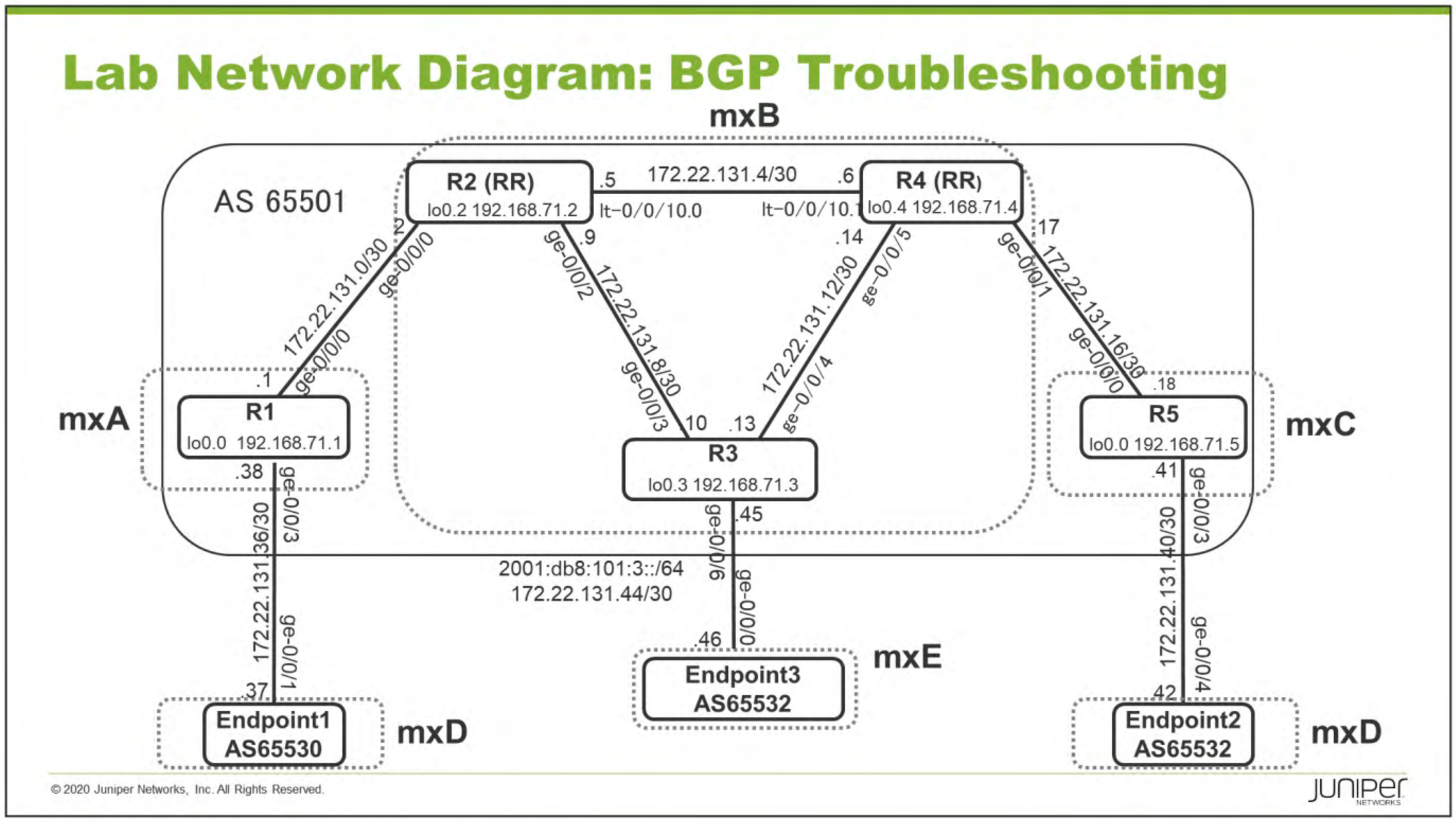
The slide diagram is referenced in the lab

.



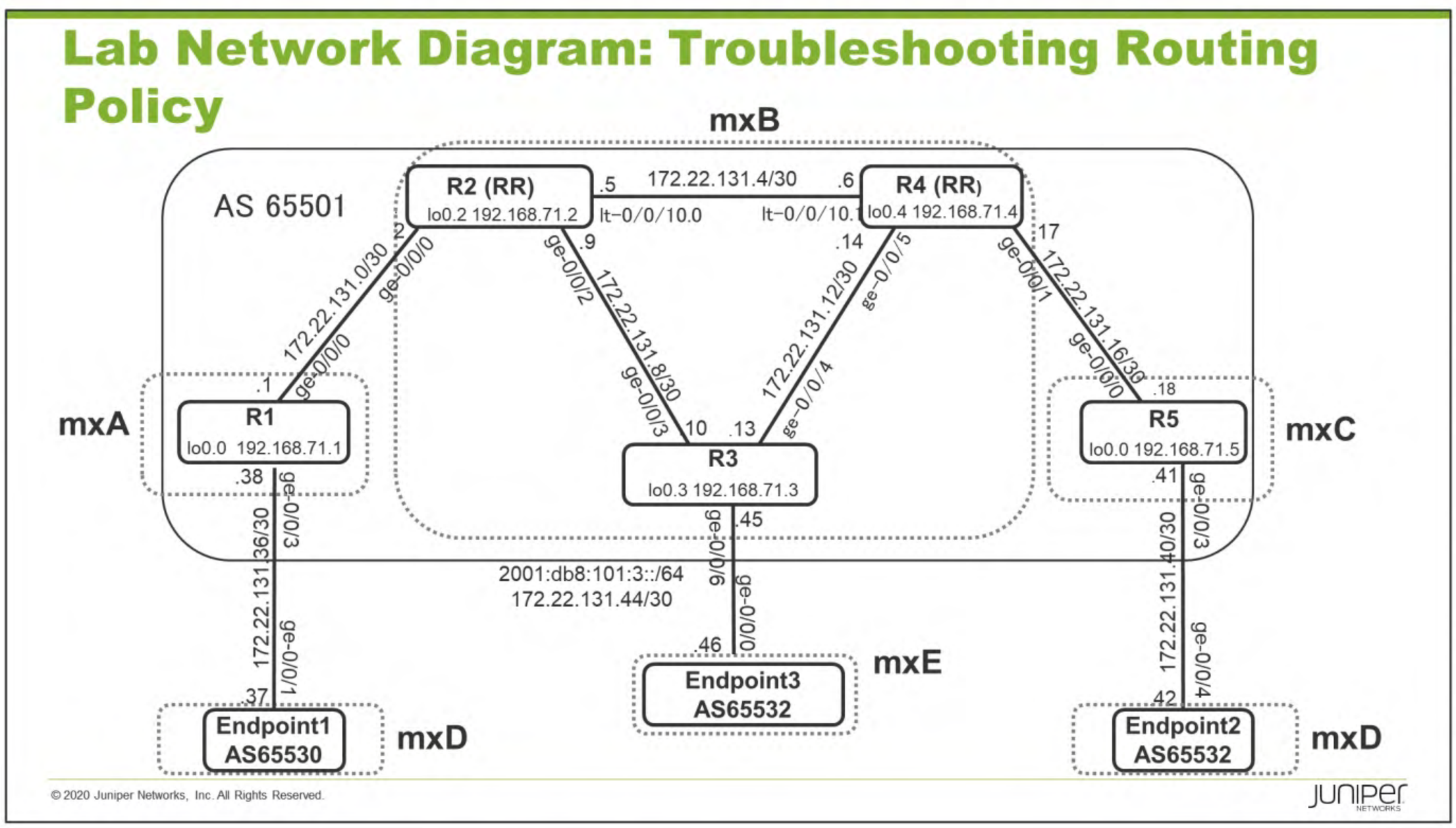
Lab Network Diagram: BGP FlowSpec

The slide diagram is referenced in the lab.



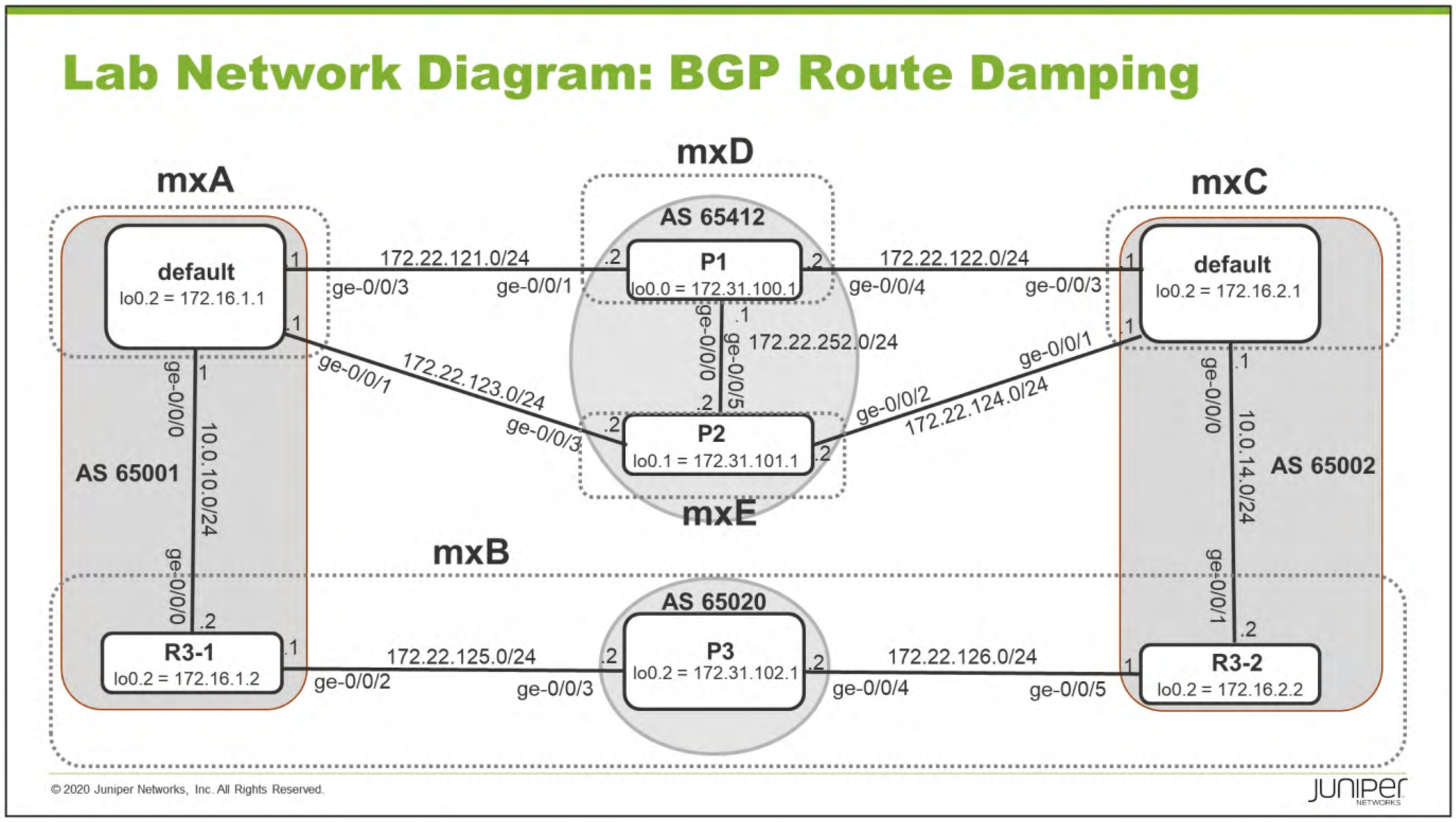
Lab Network Diagram: BGP Troubleshooting

The slide diagram is referenced in the lab.



Lab Network Diagram: Troubleshooting Routing Policy

The slide diagram is referenced in the lab.



Lab Network Diagram: BGP Route Damping

The slide diagram is referenced in the lab.

Acronym List

ABR . . . . . area border router
ARP . . . . . Address Resolution Protocol
AS . . . . . autonomous system
ASBR . . . . . autonomous system boundary router
BFD . . . . . Bidirectional Forwarding Detection
BGP . . . . . Border Gateway Protocol
BGP4 . . . . . Border Gateway Protocol version 4
CLI . . . . . command-line interface
CLNP . . . . . Connectionless Network Protocol
CLNS . . . . . Connectionless Network Service
CSNP . . . . . complete sequence number PDU
DIS . . . . . designated intermediate system
DoS . . . . . denial-of-service
DR . . . . . designated router
EGP . . . . . exterior gateway protocol
ES . . . . . end system
ES-IS . . . . . End System–to–Intermediate System
GUI . . . . . graphical user interface
IAB . . . . . Internet Advisory Board
IANA . . . . . Internet Assigned Numbers Authority
IBGP . . . . . internal BGP
ICMP . . . . . Internet Control Message Protocol
IGP . . . . . interior gateway protocol
IPv4 . . . . . IP version 4
IPv6 . . . . . IP version 6
IS . . . . . Intermediate System
ISO . . . . . International Organization for Standardization
ISP . . . . . Internet service provider
JNCP . . . . . Juniper Networks Certification Program
LSA . . . . . link-state advertisement
LSDB . . . . . link-state database
LSP . . . . . link-state PDU
MD5 . . . . . Message Digest 5
MED . . . . . multiple exit discriminator
MOSPF . . . . . Multicast Open Shortest Path First
NET . . . . . network entity title
NLPID . . . . . network layer protocol identifier
NLRI . . . . . network layer reachability information
NSSA . . . . . not-so-stubby area
OSI . . . . . Open Systems Interconnection
PDU . . . . . protocol data unit
PFE . . . . . Packet Forwarding Engine
PSNP . . . . . partial sequence number PDU
RIB . . . . . Routing Information Base
RID . . . . . router ID
rpd . . . . . routing protocol daemon
SNPA . . . . . subnetwork point of attachment
SPF . . . . . shortest path first
TED . . . . . traffic engineering database
TLV . . . . . type/length/value
ToS . . . . . type of service
TTL . . . . . time-to-live

Education Services

Corporate and Sales Headquarters
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or 408.745.2000
Fax: 408.745.2100
www.juniper.net

APAC and EMEA Headquarters
Juniper Networks International B.V.
Boeing Avenue 240
1110 PZ SCHIPHOL-RIJK
Amsterdam, Netherlands
Phone: 31.0.207.125.700
Fax: 31.0.207.125.701

Copyright 2020
Juniper Networks, Inc. All rights reserved.

Juniper Networks, the Juniper Networks logo, Junos, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. All other trademarks, services marks, registered marks, or registered services marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Engineering
Simplicity

EDU-JUN-AJSPR, Revision V19A