11/03/95

AIXlink/X.25 APAR IX52536 Enhancement Summary

Special Notices

Information in this document is correct to the best of our knowledge at the time of this writing. Please send feedback by fax to "AIXServ Information" at (512) 823-4009.

Please use this information with care. IBM will not be responsible for damages of any kind resulting from its use. The use of this information is the sole responsibility of the customer and depends on the customer's ability to evaluate and integrate this information into the customer's operational environment.

About This Document

This document describes the new and improved functionality that is included in APAR IX52536 of the AIXlink/X.25 LPP and contains the following information:

- x25status command
- Dialup Functionality
- PAD functionality

New x25status Command

A new command, x25status, has been added to allow customers to view the current status of their ports. It will display the current state of of the packet layer on all defined ports. The following is a list of possible packet layer states:

```
Disconnected
Restart in Progress
Registration in Progress
Line Clean up in Prgress
Packet Layer Connected
```

The packet layer states actually reflect the state of the link between the DTE and the DCE.

Every time a virual circuit is established, the Active SVCs or Active PVCs count is incremented depending on the type of virtual circuits the port uses.

Here is some sample output:

```
SAMPLE 1:
---------
Link status for abbas
                                          Active           Active
Port              Packet State            SVCs              PVCs
sx25a0                      PACKET LAYER CONNECT
sx25a1                      DISCONNECT
sx25a2                      DISCONNECT
where "abbas" is your machine name.

In this example, port sx25a0 has a link, but a virtual
circuit has not been established.  Ports sx25a1 and
sx25a2 are not active.

SAMPLE 2:
---------

Link status for abbas
                                                 Active  Active
Port            Packet State                     SVCs    PVCs
sx25a0          PACKET LAYER CONNECTED           1       0
sx25a1          PACKET LAYER CONNECTED           1       0

In this example, both ports have a link with the DCE and an established
virtual circuit.
```

New Dialup Functionality

Summary of Changes

The dialup functionality in the AIXlink/X.25 LPP changed to allow:

1. Both incoming calls and outgoing calls to be made on the same port.
2. The connection between the DTE and the DCE to be established when needed.
3. The connection between the DTE and DCE to be disconnected after a period of inactivity.

Enhancement Specifics

Enhancement #1:

Previous implementations of the AIXlink/X.25 dailup functionality restricted a port to either sending outgoing calls (OUTCALL) or receiving incoming calls (INCALL). This restriction was very limiting and required customers to reconfigure the port to change the call direction. Hence the AIXlink/X.25 LPP was enhanced so that all ports can send and receive calls. Since the call direction information is no longer needed, the V25bis DTE Call Direction option was removed from the SMIT Change / Show X.25 General Parameters screen.

Enhancements #2 and #3:

Previous implementations of the AIXlink/X.25 dialup functionality established the link between the local DTE and the local DCE when the port was configured AVAILABLE. This link remained established until one of the ports was unconfigured. The new implementation of the dialup functionality establishes the link when a call is issued by the DTE or a call is received by the DCE. The link remains established until one of the ports is unconfigured or the packet layer idle line timer on one of the ports expires.

The packet layer idle line timer can be configured from the Change / Show X.25 Packet Parameters screen in SMIT. Once the link is torn down, it can be re-established if a call is received from the remote DTE or if another call is sent from the local DTE. If a call request fails, a timeout equal to the packet layer idle line timer must expire before another call request may be sent from that port. However, incoming calls can still be processed.

```
An example,
     DTE                          DTE B
idle line timer = 60 seconds         idle line timer = 30 seconds
```

Sequence of Events

1. DTE A sends a call to DTE B. This causes the link between DTE A and DCE A to be established. Then, the call is put onto the network.
2. When DCE B receives the call, it notifies DTE B which causes the link between DCE B and DTE B to be established.
3. The call is accepted and data is transferred between DTE A and DTE B.
4. Then, either DTE A or DTE B decides to end the call. This causes the virtual circuit to be disconnected.

   However, the links between DTE A and DCE A and DTE B and DCE B are still established.
5. After 15 seconds, DTE B calls DTE A.

   Since the connection between DTE B and DCE B is still connected, the call immediately goes out.

   Again, since the link between DTE A and DCE A is still established, the call is immediately delivered to DTE A.
6. DTE A accepts the call and data is transferred.

   After a while DTE B decides to end the call. Again, the virtual circuit is cleared, but the links between the DTEs and their respective DCEs are still connected.
7. Now, DTE B does not send or receive any calls for 30 seconds. So, DTE B's idle line timer expires. This causes the link between DTE B and DCE B to be disconnected.

   In the meantime, DTE A does not receive or send any calls for 60 seconds.

So, DTE A's idle line timer expires and the link between DTE A and DCE A is disconnected.

8. Then, DTE A sends a call to DTE B. The link between DTE A and DCE A gets established.

   However, the call request fails for some reason. The application that sent the call on DTE A receives a CLEAR indication from the packet layer. Since the call request failed, applications running in DTE A will not be able to send any more call requests for 60 seconds (idle line timeout). If they attempt to send a call request before the timer expires, the applications on DTE A will receive an immediate CLEAR indication from the packet layer.

   However, DTE A can receive calls during this timeout period. The timeout only effects the outgoing calls, not the incoming calls.

   - So, if DTE B sends a call request to DTE A 5 seconds after DTE A's call request failed and the call request actually got to DTE A then, DTE A could accept the call.

     However, DTE A would still not be able to send a call request for another 55 seconds.

New Pad Functionality

This release of X.25 has enhanced PAD functions that are described below. Instructions for use are also included.

The new PAD functions allow the user to customize the local PAD parameters, customize the remote tty parameters, automatic login of incoming calls, and restriction of outgoing calls. For outgoing connections the user may customize the X.3 parameters and for incoming connections they can customize the X.3 & tty parameters.

These changes have effected the x29d daemon. The user may now start the daemon with flags for use with this feature. The x29d daemon MUST be running for this feature to work.

It also must be running for restriction of outgoing calls and validation of incomings calls. However, the daemon can be started to perform only 1 function, that is the restriction of outgoing calls.

Overview of New Features

There are three basic PAD features that are controlled by the four new PAD configuration files located in the /etc/sx25pad directory:

1. Automatic login of incoming calls based on calling address.
2. Automatic selection of tty and PAD parameters for incoming calls based on calling address and login user.
3. Restriction of outgoing calls to specific addresses based on calling user.

The four PAD configuration files are named, as follows:

tty Defines tty attributes, default tty attributes are "cooked" (ie. local echo, editing, etc.).

profile Defines PAD parameters, default is transparent with some changes.

address Selects features based on address, default uses "cooked" tty and default profile.

user Selects features based on username, default uses "cooked" tty, default profile, and all outgoing calls for all users.

For more information on the address and user files please see the top of these files for a more detailed description of how to use them and what the various attributes are.

Configuration file format

Each PAD configuration file contains a set of stanzas of this form:

```
key1:
        attribute1 = value1
        attribute2 = value2
        attribute3 = value3
        ...

key2:
        ...
```

Each new key begins after a blank line. The key field and some attribute values may contain pattern-matching characters that are defined as follows:

* Matches zero or more characters

? Matches any single character

[] Matches any single character in the enclosed set of characters; ranges of characters can be defined using the '-' separator, e.g. "[0-9a-f]"

| Separates values in a list of values

For example:

12* Matches any value beginning with "12"

12? Matches any value beginning with "12" followed by any single character

[1-3]* Matches any value beginning with "1", "2", or "3"

12*|23* Matches any value beginning with either "12" or "23"

The configuration file is searched sequentially, and the first key value that matches the search key is used. To avoid redundant attribute definitions, it is possible to use the special "default" attribute to specify another stanza which supplies attribute values not explicitly defined in the matching stanza. If the

attribute definition is still not found after searching any specified default stanzas, the stanza with the special "default" key is checked. For example:

```
default:
        attribute1 = value1
        attribute2 = value2
        attribute3 = value3

key1:
        attribute1 = value1

key2:
        default    = key1
        attribute2 = value2

key3:
        default    = key2
        attribute3 = value3
```

In this case, the value for key2's attribute1 will come from key1, since it is not defined in key2's stanza, and key1 is key2's default stanza. Since an arbitrary number of default links can be followed, key3's attribute1 will also come from key1, as the default links are followed from key3 to key2, and key2 to key1. Finally, the value for key2's attribute3 will come from the default stanza, since neither key2 nor key2's default stanza, key1, define a value for it.

Call packets may be received without calling the address specified. In these cases, the "null" key value may be used in the address file to match a non-existent calling address.

New Features in Detail:

Automatic login feature

The automatic login feature is implemented by defining the "login_user" attribute in the address file, with a value of an existing username. When a call is received from a matching address, the call is automatically logged in as that user, instead of being presented with the normal system authentication method, (ie. getty/AIX login prompt).

For example:

```
31*:
        login_user = user1
```

will automatically login any calls from an address beginning with "31" as the user "user1".

Automatic tty and PAD parameter selection

The tty and profile files are used to define sets of parameters by name. These are then pointed to by the "in_tty" and "in_profile" attributes, respectively, in the user and/or address files. This allows the selection of any combination of host tty and remote PAD parameters based on calling address, login user, or both.

For example, if the following stanza exists in the tty file:

```
tty1:
        echo  = off
        vmin  = 1
        vtime = 1
```

and the following stanza exists in the user file:

```
user1:
        in_tty = tty1
```

then any call that logs in as "user1" will have its tty parameters set according to the values defined in the "tty1" stanza in the tty file.

Since these two attributes, "in_tty" and "in_profile", may appear in either the user or address files, it is necessary to decide which should take priority if both are defined. By default, the user attribute has priority; i.e., if both the matching address and user entries define in_tty or in_profile, then the value is taken from the user file. However, this default prioritization can be changed by explicitly setting the priority of the attribute as follows:

```
attribute = value;priority
```

where "priority" is a decimal number, with lower numbers being more favored, and 0 being the most favored priority. For example, assume the user file has the following stanza:

```
user1:
    in_tty = tty1
```

and the address file has the following stanza:

```
addr1:
    in_tty = tty2
```

If a call is received from addr1 and logs in as user1, then the tty would be set according to the "tty1" definition in the tty file. If, however, the address stanza looks like:

```
addr1:
    in_tty = tty2;0
```

then all calls received from addr1 will get the "tty2" settings, regardless of which user they log in as.

Restriction of Outgoing Calls

The "out_allow" and "out_deny" attributes can be defined in the user file to control which addresses specific users may place PAD calls to. The attribute values may consist of patterns, so that a complex set of addresses can be specified. In order for a call to succeed, the called address must match the "out_allow" attribute for the calling user, and it must not match the "out_deny" attribute.

For example, if the following stanza exists in the user file:

```
user1:
        out_allow = 31*
        out_deny  = 31[3-5]*
```

Then user1 is allowed to call any address starting with "31", except for those starting with "313", "314", or "315".

Note that the user file is shipped with the default stanza having the "out_allow" attribute set to "*", which permits all users without explicit entries in the user file to place calls to any address.

X29D Usage Changes & Flags

The default x29d mode is to begin executing both as an outgoing PAD call validation daemon, and an X.29 listener daemon. (i.e. A single process with dual functionality.)

If an instance of x29d is executing as an outgoing PAD call daemon--because it was invoked with the -n flag--and x29d is invoked in the default mode--with no flags, a new instance begins executing only as an incoming X.29 call server daemon.

The x29d may now be started in the following manner.

```
x29d [-a] [-p user] | [x-n]
```

-p<user> Set the initial TTY and X.3 attributes of an X.29 session as described in step 3 on page 4. <user> is an AIX login ID. This flag can not be used in conjunction with the -n option and requires that x29d is already executing as the X.29 Listener daemon.

-a Output the calling X.25 address of the system which initiated an X.29 session to stdout. x29d executes and obtains the X.25 address as in step 3 on page 4 but, an AIX login ID is not needed since this option is not related to X.3 or TTY attributes.

Note: This flag **can not** be used in conjunction with the -n option and requires that x29d is already executing as the X.29 Listener daemon.

-n Start execution of x29d as a outgoing call validation daemon. The -a and -p options **can not** be used with this option.

Reader's Comments

Please fax this form to (512) 823-4009, attention "AIXServ Information". You may also e-mail comments to: elizabet@austin.ibm.com. These comments should include the same customer information requested below.

Use this form to tell us what you think about this document. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

If you need technical assistance, contact your local branch office, point of sale, or 1-800-CALL-AIX (for information about support offerings). These services may be billable. Faxes on a variety of subjects may be ordered free of charge from 1-800-IBM-4FAX. Outside the U.S. call 415-855-4329 using a fax machine phone.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Note: If you have a problem report or item number, supplying that number may help us determine why a procedure did or did not work in your specific situation.

Problem Report or Item #: Branch Office or Customer #:

Be sure to print your name and fax number below if you would like a reply:

Name: Fax Number:

END OF DOCUMENT (x.25.apar.414.tcp, 4FAX# 4726)