ARAP Debugging Enhancements

Feature Summary

The AppleTalk Remote Access Protocol (ARAP) debugging has been enhanced to allow debugging on one or more specific lines rather than on all the lines connected to an access server.

Benefits

Allowing users to specify a single line via an additional parameter for debugging produces the following benefits:

- Focused results—Users get only the information they need.
- Reduced server load—Heavily loaded servers are subject to developing ARAP problems which need to be fixed by debugging. However, debugging itself increases the server work load. By focusing on specific lines, the impact of debugging activity on the server is minimized.
- Targeting flexibility—By being able to debug on just the lines in a group of lines, users can solve problems in rotary groups in which there is no way to specify which line or group of lines a remote user will be assigned.

Platforms

This feature is supported on all Cisco routers and access servers that support the AppleTalk software feature set.

Supported MIBs and RFCs

None.

Configuration Tasks

Perform the following tasks to enable ARAP debugging:

- Specify the debugging mode (for example, internal, memory).
- Specify what is to be debugged (for example, a line, group of lines, console).
- Enter the **show debug** command to examine the debug output.

Configure ARAP debugging as shown in Table 1.

Configure

Table 1 Configuring ARAP debugging

Step	Command	Purpose
1	router> **debug arap** {**internal** | **memory** | **mnp4** | **v42bis**}	Enter debug mode. Specify the type of the debug. To debug internal ARA packets, specify **internal**. To debug the memory allocated to ARA, specify **memory**. To debug the serial protocol, specify **mnp4**. To debug compression, specify **v42bis**.
2	router# **debug arap internal** *[linenum* [**aux**|**console**|**tty**|**vty**]]	Replace the *linenum* variable with a single line number. Specify the target for the debug. Specify **aux** to debug an auxiliary line, **console** to debug a primary terminal line, **tty** to debug a physical terminal asynchronous line, or **vty** to debug a virtual terminal line

Verify

Enter the **show debug** command to see if the debug level and target are set correctly:

```
router# show debug
router# show debug
AppleTalk Remote Access:
ARAP MNP4 debugging is on for line 7
```

Configuration Examples

The following example sets ARAP debugging in memory mode on line 7. The **show debug** command confirms the configuration.

```
router# debug arap mn 7
 ARAP MNP4 debugging is on for line 7
router# debug arap mn 8
 ARAP MNP4 debugging is on for line 8
router# debug arap mn 9
 ARAP MNP4 debugging is on for line 9
router# show debug
 AppleTalk Remote Access:
   ARAP MNP4 debugging is on for line 7
   ARAP MNP4 debugging is on for line 8
   ARAP MNP4 debugging is on for line 9
```

Note that you can debug several lines (for example, lines in a rotary), but you must turn on debugging one line at a time.

The following example sets ARAP debugging in internal mode on line 6, memory mode on line 10, and V.42bis compression mode on line 6. The **show debug** command confirms the configuration.

```
router# debug arap in 6
 ARAP internal packet debugging is on for line 6
 router# debug arap me 10
 ARAP memory debugging is on for line 10
 router# debug arap v 6
 ARAP V.42bis debugging is on for line 6
 router# show debug
 AppleTalk Remote Access:
   ARAP V.42bis debugging is on for line 6
   ARAP internal packet debugging is on for line 6
   ARAP memory debugging is on for line 10
```

The following example sets ARAP debugging for each mode in succession and for all lines. The **show debug** command confirms the configuration.

```
router# debug arap mnp4
 ARAP MNP4 debugging is on
 router# debug arap internal
 ARAP internal packet debugging is on
 router# debug arap v42bis
 ARAP V.42bis debugging is on
 router# debug arap memory
 ARAP memory debugging is on
 router# show debug
 AppleTalk Remote Access:
   ARAP MNP4 debugging is on
   ARAP V.42bis debugging is on
   ARAP internal packet debugging is on
   ARAP memory debugging is on
 router#
```

The following example sets all debugging (including ARAP debugging) for all modes and for all lines. The **show debug** command confirms the configuration.

```
router# debug all
 This may severely impact network performance. Continue? [confirm]y
 All possible debugging has been turned on
 router# show debug
 "debug all" is in effect.
```

Caution Do not use the **debug all** command in networks with medium to high traffic loads as you may overload and crash the router. The warning that turning on all debugging utilities can "severely impact network performance" is considered something of an understatement.

The following example turns off ARAP debugging. The **show debug** command confirms the configuration.

```
router#
 router# undebug all
 All possible debugging has been turned off
 router# show debug
 router#
```

The following is a sample of debug output for two lines, 2 and 4:

```
ARAP MEM TTY 4: arap_getbuffer 94745C
 ARAP MEM TTY 4: arap_datagram_done 7BD324
 MNP4 TTY 4:mnp4_input()
 MNP4 TTY 2:mnp4_input()
 ARAP MEM TTY 2: arap_getbuffer 7BD158
 MNP4 TTY 2:Rcv LA Nr[31] Nk[08]
 ARAP MEM TTY 2: arap_datagram_done 7BD6BC
 MNP4 TTY 4:mnp4_input()
 ARAP SMARTBUF TTY 2: ring end 936C62, start 934ED4, need 58 bytes
 ARAP SMARTBUF TTY 2: new seq 161

 ARAP TTY 4: Received TICKLE
 ARAP TTY 4: --------- ACKing 125 ---------
 ARAP SMARTBUF TTY 2: ring end 936C28, start 934ED4, need 58 bytes
 ARAP SMARTBUF TTY 2: new seq 160
 ARAP SMARTBUF TTY 2: ring end 9342B4, start 9322EC, need 64 bytes
 ARAP SMARTBUF TTY 2: new seq 144
 ARAP SMARTBUF TTY 2: search...
 ARAP SMARTBUF TTY 2: search...
0 ddp; trailing; 1 ddp; trailing; 2 ddp; trailing; 3 ddp; trailing; 4 ddp; trailing; 5
ddp; 6 offset; 7 ddp; trailing; 8 ddp; 9 offset; 10 ddp; trailing; 11 ddp; trailing; 12
ddp; trailing; 13 ddp; trailing; 14 ddp; 15 ddp; trailing; 16 ddpARAP SMAR
@TBUF TTY 2: ring end 936C62, start 934ED4, need 58 bytes
 ARAP SMARTBUF TTY 2: new seq 161

 ARAP TTY 4: Received TICKLE
 ARAP TTY 4: --------- ACKing 125 ---------
 ARAP TTY 2: Received TICKLE
 ARAP TTY 2: --------- ACKing 114 ---------

 V42bis TTY 4: OUT uncomp (12): 0 10 16 33 0 9 1 195 255 255 255 255
 V42bis TTY 4: OUT comp (6): 10 38 229 203 3 0
 V42bis TTY 4: IN comp (6): 205 145 196 79 2 0
 V42bis TTY 4: IN uncomp (12): 0 10 16 143 0 9 0 0 255 255 255 255
 V42bis TTY 4: OUT uncomp (6): 0 4 16 143 0 0
 V42bis TTY 4: OUT comp (6): 182 244 235 0 2 0
 V42bis TTY 4: IN comp (6): 217 111 250 0 2 0
 V42bis TTY 4: IN uncomp (6): 0 4 16 33 0 0
 V42bis TTY 2: IN comp (5): 247 225 15 102 0
 V42bis TTY 2: IN uncomp (12): 0 10 16 132 0 9 255 219 255 255 255 255
 V42bis TTY 2: OUT uncomp (6): 0 4 16 132 0 0
 V42bis TTY 2: OUT comp (6): 126 63 196 65 2 0
   ...........
   ...........
```

The boldfaced portion of this example shows that for line 2, `LA` is the MNP4 acknowledge frame, `31` is the sequence number of the last frame, and `08` is the window size.

Command Reference

debug arap

Use the **debug arap** EXEC command to display AppleTalk Remote Access Protocol (ARAP) events. The **no** form of this command disables debugging output.

[**no**] **debug arap** {**internal** | **memory** | **mnp4** | **v42bis**} [*linenum* [**aux** | **console** | **tty** | **vty**]]

Syntax Description

internal	Debug internal ARA packets.
memory	Debug memory allocation for ARA.
mnp4	Debug low-level asynchronous serial protocol.
v42bis	Debug V.42bis compression.
linenum	(Optional) Line number. The number ranges from 0 to 999, depending on what type of line is selected.
aux	(Optional) Auxiliary line.
console	(Optional) Primary terminal line.
tty	(Optional) Physical terminal asynchronous line.
vty	(Optional) Virtual terminal line.

Usage Guidelines

This command first appeared in Cisco IOS Release 10.2. Use the **debug arap** command with the **debug callback** command on access servers to debug dial-in and callback events.

Use the **debug modem** command to help catch problems related to ARAP auto-detection (that is, **autoselect arap**). These problems are very common and are most often caused by modems, which are the most common cause of failure in ARAP connection and configuration sessions.

Sample Display

The following shows sample **debug arap internal** output:

```
Router# debug arap internal

ARAP: ---------- SRVRVERSION ----------
ARAP: ---------- ACKing 0 ----------
ARAP: ---------- AUTH_CHALLENGE ----------
arapsec_local_account setting up callback
ARAP: ---------- ACKing 1 ----------
ARAP: ---------- AUTH_RESPONSE ----------
arap_startup initiating callback ARAP 2.0
ARAP: ---------- CALLBACK ----------
TTY7 Callback process initiated, user: dialback dialstring 40
TTY7 Callback forced wait = 4 seconds
TTY7 ARAP Callback Successful - await exec/autoselect pickup
```

```
TTY7: Callback in effect
ARAP: ---------- STARTINFOFROMSERVER ----------
ARAP: ---------- ACKing 0 ----------
ARAP: ---------- ZONELISTINFO ----------
ARAP: ---------- ZONELISTINFO ----------
ARAP: ---------- ZONELISTINFO ----------
ARAP: ---------- ZONELISTINFO ----------
ARAP: ---------- ZONELISTINFO ----------
```

Related Command

debug callback
debug modem

What to Do Next

For more information, refer to the “Configuring AppleTalk Remote Access” chapter in the *Dial Solutions Configuration Guide*.