

Table of Contents

<u>Troubleshooting Problems When Adding a Trunk</u>	1
<u>Document ID: 10775</u>	1
<u>Introduction</u>	1
<u>addtrk Failure</u>	2
<u>Quick Checks of addtrk on a Physical Trunk</u>	2
<u>Isolating the Failure</u>	2
<u>Network Trace</u>	2
<u>Display Statistics</u>	3
<u>Quick Checks of addtrk on a Virtual Trunk</u>	4
<u>Isolating the Failure</u>	5
<u>Comm Fail</u>	5
<u>Permanent Comm Fail</u>	5
<u>Intermittent Comm Fail</u>	5
<u>Unreachability</u>	6
<u>Introduction</u>	6
<u>Isolating the Failure</u>	6
<u>Conclusion</u>	7
<u>Related Information</u>	7

Troubleshooting Problems When Adding a Trunk

Document ID: 10775

Introduction

addtrk Failure

Quick Checks of **addtrk** on a Physical Trunk

Isolating the Failure

Network Trace

Display Statistics

Quick Checks of **addtrk** on a Virtual Trunk

Isolating the Failure

Comm Fail

Permanent Comm Fail

Intermittent Comm Fail

Unreachability

Introduction

Isolating the Failure

Conclusion

Related Information

Note: For information about Setting Up Trunks, refer to that section in the user manual.

Introduction

Trunks are intra-node communication links in a network. A trunk can connect any combination of IPX, IGX, or BPX nodes. Trunks are activated after node configuration.

A trunk can connect any combination of IPX, IGX, or BPX nodes. Trunk characteristics are:

Physical line type:	T1 (including fractional), E1 (including fractional), Subrate, E3, T3, or OC3 (STM1)
Communication technology:	Asynchronous Transfer Mode (ATM) or StrataCom FastPackets

Error messages such as "addtrk fails with no response from other node," "Comm Fail," or "unreachable nodes," can be attributed to a variety of causes. The information in this document can assist you in narrowing down the cause of the error message.

Note: It is assumed that the correct hardware and firmware versions are loaded. Refer to the Cisco WAN Switching Solutions user documentation for more information about the particular product you are using.

addtrk Failure

If **addtrk** fails with the message "no response from the other node", communication between two adjacent nodes has failed. A distinction is made in this document between physical and virtual trunks, as virtual trunks could go through a cloud involving even more components.

Quick Checks of addtrk on a Physical Trunk

Your first check should be to compare the Payload Scramble setting on both sides of the trunk. Enter the **cnftrk** command to show it. Both settings must be identical. This also applies to virtual trunks that do not go through a cloud but are directly over a physical trunk.

Next enter a **nwstats** command on both nodes to look for errors.

3. First clear the stats with:

```
nwstats c
```

2. Make sure statistics are turned on.
1. See whether any of the error counts increase.

On BPX you can also use **srstats** to find errors. Enter

```
srstats n
```

to print networking-related stats.

Also look at the **Peak VRAM Queue Depth**. It should be no greater than one or two digits. If it is larger than two, clear the stats with:

```
nwstats c
```

and see whether the queue depth quickly increases. A four-digit Peak VRAM Queue Depth could mean cells are being dropped. If the number of BPX frames (Bframes) is significantly larger than the number of FastPackets, AAL5 traffic is flooding the segmentation and reassembly process (SAR). Try entering:

```
dsplog
```

The log might show some unexpected hardware failures.

Isolating the Failure

In this case, packets are generated by the BCC/NPC and sent via the local trunk card and remote trunk card to the remote BCC/NPC. Response packets are sent back the other way. The following sections describe some of the possible points of failure.

Network Trace

First, do a network trace on both sides of the trunk using the command **nwtrace**.

In order to trace only the relevant cells, define a filter. The values are slightly different for BPX and IPX.

For the BPX:

- **Node Num: 224** – Blind address
- **Part Num: *ltrk*** – This is the logical trunk ID, which can be found by using the **dnib** command

The other fields are left as "ANY". The first screen of the **dnib** command looks something like this:

```
Trunk: 0 1 2 3 4 .....  
LogCd: - - 2 - 2 .....  
Port:  - - 1 - 0 .....  
Vtrk:  - - 1 - - .....  

```

The number after "Trunk" is the number of logical trunk IDs; the next three lines contain the (logical) *slot.port.vtrk* numbers as displayed by the **dsptrks** command, with the notable exception that the port number is decremented by one.

We see virtual trunk 2.2.1 has the logical trunk ID of 2, and physical trunk 2.1 has the logical trunk ID of 4.

For the IPX/IGX:

If you are on an IPX/IGX, the node number depends on the trunk. To figure out the blind node number used on your trunk, leave **Node Num** and **Part Num** at **ANY** and set the function code to **28**. This will trace only the first message, but this is probably enough. In case you want to trace more messages, the trace of this first message tells you the Node Num to be used, and you can change the filter using **nwtrace** accordingly.

After entering **nwtrace**, type **cnw** to clear the network trace buffer on both sides. Then, type **addtrk** again.

Now display the trace with:

```
dnw
```

On the node where **addtrk** was issued, you will see one or more messages transmitted. The result could be "Ack", or probably, "Tmout." Determine whether the other node received any message.

- If no, one of the first three legs did not relay the packets; continue to issue **addtrk** commands from the same node.
- If yes, one of the three return legs did not work; issue **addtrk** from now on from the other node.

Display Statistics

For BPX:

If you are on a BPX, you can now use the command **dsptrkstats** to see whether any cells were transmitted on the trunk. Note that statistics must be enabled with the **on2** command. You can clear the stats first with **clrtrkstats** before issuing **addtrk** on that node.

- If **dsptrkstats** shows more than 0 (zero) cells transmitted, the leg from the BCC to the trunk card worked.

- If it shows zero, there is a problem between the BCC and the trunk card, or else the firmware did not transmit cells it received from the BCC.

Check whether the transparent channel is programmed. This can be done with the **dspchrte** command:

```
dspchrte slot.port chn d
```

slot.port is the same as for **addtrk**. *chn* is **0** for BNI cards or (**port – 1**) for BXM cards.

If the response indicates that the channel is deleted, you have found the cause. If the channel is not deleted, the data did not make it across the cross-bar switch, or else the firmware ignored it.

For BXM cards, the command:

```
dspchstats slot.port.chn 1
```

might show something unusual (meaning non-zero numbers), which would indicate some indisposition of the firmware. This is a Resource problem. Check whether the cross-bar switch is programmed correctly, or whether something upset the card.

If cells were transmitted by the trunk card, see if those cells have looped back and whether the number of received cells is equal to the number of transmitted cells. If not, look at the other trunk card (on the other node) to see whether it received anything. Using a **dsptrkstats** command for a BPX should show the number of cells received, and the number should increase with every issued **addtrk** command.

If it shows that no cells were received, something may have happened on the trunk (this is not very likely). Otherwise, you might see a non-zero entry for "Cell header mismatch error count" and the VPI/VCI for that header. (This currently works only on BNI cards.)

For BNI cards:

Check whether channel 10 is programmed:

```
dspchrte slot.port 10 d
```

For BXM cards:

```
dspchrte slot.port chn d
```

where *chn* is **42** for port 1, **312** for port 2, **582** for port 3, for example.

If the channels are programmed, there may be a firmware problem.

Quick Checks of addtrk on a Virtual Trunk

Basically, a virtual trunk is handled the same as a physical trunk. In case of a virtual trunk over a cloud (the normal case), ASI cards and connections are involved.

Payload scrambling has to be consistent between the trunk and the ASI card to which it is connected. The two ends of the trunk don't have to be the same. By default, E3 and OC3 ASI cards enable payload scrambling, and T3 disables it.

The virtual path identifier (VPI) values configured at the trunk ends must be consistent with the connection defined for this virtual trunk.

Again, **nwstats** and **srstats** might give some quick hints. See the section Quick Checks of `addtrk` on a Physical Trunk for details.

Isolating the Failure

Do the same initial analysis as for physical trunk.

If the packets are lost between the two trunk end points, you can now look at the statistics for the ASI card whose trunks are connected too.

The following command will show where cells are going:

```
dspchstats slot.port.network 1
```

where *port* refers to the line port, and *Network* refers to cells going over the muxbus/crossbar switch of the BPX/IPX/IGX.

If the cloud is not only a daxcon, but also involves a trunk, stats on the trunk can also be looked at (**dsptrkstats** for a BPX). Look for loss of connectivity between the ASI and this internal trunk. If this is the case, you have a Connection Management problem.

Comm Fail

Permanent Comm Fail

The case of a permanent Comm Fail is very similar to the problem of not being able to add a trunk, and the same analysis procedure should be used.

In this case, do not trace a message with function code 28; use function codes 63 and 64. These messages are also sent to the blind nib, so the addresses to trace are the same as for the `addtrk` problem.

Intermittent Comm Fail

There can be several reasons for an intermittent Comm Fail. The **nwtrace** commands can help by showing whether messages get ACKed; or whether you receive Comm Fail Test messages (63, 64) from the remote, but no ACK.

Again, isolate the failure by following the path of the packets and looking at every intermediate link. Use the statistics command to see whether cells (packets) are being dropped.

The problem is most likely a Connection Management problem if data is being pumped across the trunks and overload is generated.

It is more likely a problem related to Resource when no overload is present. Qbins may not have been set up correctly.

Unreachability

Introduction

This is the most involved problem because, as with physical trunks, it usually involves more than two nodes. Many of the techniques discussed above, however, can also be applied here.

As an example, consider a situation where node "B" becomes unreachable from node "A". A message is sent from node "A" to node "B", but no ACK is received by node "A". At this time, node "B" might still consider node "A" reachable. There is no periodic test, like the Comm Fail test, to test reachability. It is only after node "B" becomes unreachable from node "A" that there are Comm Break test messages.

Isolating the Failure

Network Trace

Again, it is a good first step to run **nwtrace** on both nodes to narrow down the point of failure. This time **Node Num** has to be set to the node number of the remote node. The function code of the CommBreak test message is 60.

Assume the message from node "A" does not reach node "B". In case the ACK was lost along the way, you can force a Comm Break for the other direction by VT-ing from node "B" to node "A", then try to find where the messages get lost from "B" to "A".

First, determine which route the packets take from "A" to "B". Then, to hop, you can use the same tools that you used to determine the cause for a Comm Fail. The **drtop** command shows the next hop for a packet destined for node "B". Then, log in the via node. Enter the **drtop** command to show the next one, and so forth.

If one of the trunks on the path is in Comm Fail state, this problem reverts to analyzing a Comm Fail problem.

Displaying Statistics

On the BPX:

When using **dsprkstats** (on BPX) to see whether cells are transmitted or received on a trunk, take into account that other cells are also being sent on this trunk. You can use the command **on1** to switch off the Comm Fail test on either side of the trunk to isolate more, but there might be more traffic (messages to other nodes or user data) which must either be accounted for or switched off.

On the IPX/IGX:

If an IPX/IGX is a via node, you can use the commands **setmrt** and **pktrace** to find out whether packets for node B are forwarded on this via node. The command:

```
setmrt 9 node_B_s_node_number
```

allows the via node to trace the forwarded packets for node B. The **pktrace** command has to be set up with VCI Msb as **9** and VCI Lsb as the *node number* of node B.

To clear the trace buffer, enter:

```
cpk
```

To display the trace buffer (in unstuffed format), enter:

```
dpk u
```

If a BXM card is on the route, to actually see whether cells of the message from A to B go in the right direction, enter:

```
dspchstats
```

To display transmitted (to port) or received (from port) cells, enter:

```
dspchstats slot.port.chn 1
```

where *chn* is calculated as $47 + (\text{port} - 1) * 270 + \text{node_num_of_B}$

If virtual trunks are involved, follow the procedure outlined in section on virtual trunks during Comm Fail.

Conclusion

After the failing link has been identified, see whether the **dsptrkstats** command (on BPX) or the **dspchstats** command (on BPX BXM cards) show any errors or usual statistics that could give a hint to the reason of failure.

The **nwstats** and **srstats** commands are only useful at the end node (node A and node B).

If you have tried all the troubleshooting information in this document and you are unable to locate the source of or resolve the problem, call Cisco TAC and open a case.

Related Information

- [Cisco WAN Switching Solutions – Cisco Documentation](#)
- [Guide to New Names and Colors for WAN Switching Products](#)
- [Software Center – WAN Switching Software](#)
- [Technical Support – Cisco Systems](#)

All contents are Copyright © 1992—2004 Cisco Systems Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: Sep 08, 2004

Document ID: 10775
