Recovering From errDisable Port State on the CatOS Platforms

Document ID: 12017

Contents

Introduction

Customers often contact Cisco Technical Support when they notice one or more of their switch ports have become error–disabled; that is, the ports have a status of errDisable. They want to know why this happened and how the ports can be restored to normal. This document describes what the errDisable state is, how to recover from it, and provides two examples of recovering from errDisable. Throughout this document, the terms errDisable and error–disable are used interchangeably. (errDisable is the status of a port as shown by the **show port** command, error–disable or error–disabled are the English language equivalents of errDisable.)

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

The information in this document is based on these software and hardware versions. You need these in order to create the examples in this document:

- Two Catalyst 4000/5000/6000 family switches (or their equivalent) in a lab environment with cleared configurations. Our primary machine was a Catalyst 5500 running CatOS 5.4(2). This was connected to a Catalyst 6509 running 5.3(5a)CSX, but could be any CatOS machine and capable of EtherChannel and portfast.

- Two RJ–45 Ethernet crossover cables.
- CatOS 5.4(x) on at least one switch.
- Two FastEthernet ports in each switch capable of EtherChannel and portfast.
- A terminal connection to one or both of the switches.

The information in this document was produced from an isolated lab environment. Ensure that you first understand the potential impact of any command on your network before using it. The **clear config all** command was entered on each switch to ensure a default configuration. Should you want to replicate and experiment with these errors, please only try to duplicate them in an isolated environment that will not impact your live network. These examples are for instruction only. Output from some commands has been truncated where it does not enhance the discussion.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

errDisable

Platforms Using errDisable

The errDisable feature is supported on Catalyst switches running CatOS (Catalyst 2948G, 4500/4000, 5500/5000 & 6500/6000) as well as Catalyst switches running Cisco IOS (Catalyst 2900XL/3500XL, 2950, 2970, 3550, 4500 & 65000). The way the errDisable is implemented varies between platforms. This document will specifically focus on error–disable for the switches running CatOS software.

Function of errDisable

The errDisable feature was first implemented in CatOS release 3.2(2). If the configuration showed a port to be enabled, but software on the switch detected an error situation on the port, the software would shut down that port. In other words, the port was automatically disabled by the switch operating system software because of an error condition encountered on the port.

When a port is error–disabled, it is effectively shut down and no traffic is being sent or received on that port. The port LED is set to the color orange and when you enter the **show port** command, the port status shows **errdisable**. Here is an example of what an error–disabled port would look like from the command line interface of the switch.

```
Cat5500> (enable) show port 11/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
11/1                     errdisable  1 normal   auto   auto  10/100BaseTX
```

The error–disable function serves two purposes. First, it lets the administrator know when and where there is a port problem. Second, it eliminates the possibility that this port could cause other ports on the module (or the entire module) to fail due to buffers being monopolized by the bad port, port error messages monopolizing inter–process communications on the card, even ultimately causing serious network issues. The error–disable feature helps prevent these situations.

Causes of errDisable

At first, this feature was implemented to handle special collision situations where the switch detected *excessive* or *late collisions* on a port. *Excessive collisions* occur when a frame is dropped because of encountering 16 collisions in a row. *Late collisions* occur after every device on the wire should have recognized that the wire was in use. These types of errors could be caused by a *cable that is out of specification (too long, wrong type, defective), a bad network interface card (NIC) card* (with physical problems, or driver problems), or a *port duplex misconfiguration*. This last cause is common because of failures to negotiate the speed and duplex properly between two directly connected devices (for example, a NIC card connected to a switch). Only half–duplex connections should ever have collisions in a LAN; due to the Carrier–Sense Multi–Access (CSMA) nature of Ethernet, collisions are normal for half–duplex, as long as they do not exceed a small percentage of traffic.

As the capabilities of the CatOS grew, there were more ways that a port could become error–disabled. For example on the catalyst 6500 running catOS, the Errdisable feature is supported for these connectivity issues:

- ARP inspection
- Broadcast suppression
- BPDU port–guard
- Channel misconfiguration
- Crossbar failure
- Duplex mismatch
- Layer 2 protocol tunnel misconfiguration
- Layer 2 protocol tunnel threshold exceeded
- UDLD

The error–disable function allows the switch to shut down a port when it encounters any of these situations. Remember, a port being error–disabled is not by itself a cause for alarm, as long as one determines and resolves its root cause. An error–disabled port is a symptom of a deeper problem that must be resolved.

Recovery from errDisable

In order to recover from errDisable you should do two things:

1. Identify and fix whatever caused the ports to become error–disabled (cable, NICs, EtherChannel, and so on).

   If you do not identify and fix the underlying issue that caused the ports to be error–disabled, then the ports will just become error–disabled again when the problem reoccurs. Some errors can occur quite often (an example is the error detected by BPDU portguard, which can occur every two seconds). If you tried to reenable the ports without fixing the source of the problem they would just become error–disabled again.
2. Reenable the port.

   Just fixing the source of the problem will not cause the ports to become enabled again. Once you fix the source of the problem, the ports are still disabled (and the port LEDs are still orange); the ports must be reenabled before they will become active. At first the only way to reenable the port was to manually enter the **set port enable** command for the ports in question. Over time there have been optional extensions added to the error–disable feature to make it more flexible and automatic.

Note: An error–disabled port is not the only reason a port LED could go orange; it is only one of the reasons. That is why it is always good to check the port status with the **show port** command.

Extensions to errDisable

Set Option Errport – CatOS 4.2(2)

Some customers wanted to have the ability to determine whether a port should be shut down due to special collision errors discovered by the CatOS. There were some situations, such as if the link was a backbone connection, for example, where shutting down the ports would actually be worse than the errors that were encountered on the ports; it would be more desirable to leave the ports functioning as much as possible until the problem could be addressed, rather than shutting them down. So in release 4.2(2), a new command was added to the CatOS called **set option errport** that allows the administrator to determine what action the switch took upon discovering a port having these special collision errors. The original and default state is **set option errport** *disable* , where the switch will put a port in error–disabled state when encountering the error–disable type of special collision errors. In contrast, if the command **set option errport** *enable* is used, then the switch will leave the ports enabled, even though it encounters collision errors that would normally disable those ports.

This command affects the switch globally; it cannot be issued for an individual port. It is not listed in the command reference, but is listed in the release notes for 4.2(2) (Release Notes for Catalyst 5000 Family Software Release 4.x). Please note that this command appears to be counter–intuitive; one must enable the errport option to disable the err–disable feature (enabled by default). Put more clearly, simply use the **set option errport** *enable* command to prevent a port from becoming error–disabled.

The **set option errport** command is only recommended if you realize that you incur some risk of other ports on the module being affected if you allow these error conditions to continue. It is only a stop–gap measure, not a "fix" to the problem; it merely prevents the ports that are encountering these errors from being shut down until you can address the real problem. Use with caution.

Set errdisable–timeout – CatOS 5.4(1)

With CatOS release 5.4(1), a new command called **set errdisable–timeout** is introduced. This command is a more sophisticated version of the **set option errport** command discussed earlier. This command will automatically reenable an error–disabled port after a configurable amount of time (from 30 seconds to 24 hours, specified in seconds), eliminating the need to manually reenable the error–disabled port.

This command will affect the ports that are enabled by the current configuration on the switch but have been put into the error–disable state by the CatOS software. Use the command **show errdisable–timeout** to see the current status of the errdisable–timeout feature. It is possible to specify five separate areas where this feature can be enabled: bpdu–guard, channel–misconfig, duplex–mismatch, (which includes the special collision errors mentioned above), *udld*, *other*. This way it can still give you permanent error–disable protection in areas where you want it, but allow you to selectively pick areas where you would rather have the ports keep functioning until you can fix the problem.

In software versions 5.2.1 and 5.2.2 for the Catalyst 6000 series, there is a software defect that causes network outages when a port changes state to error–disabled. When a ports goes errDisable, the switch will cause all learned MAC address to be inadvertently learned on the error–disabled port. This will cause the network outages on the associated VLAN. This software defect has Cisco bug ID CSCdm48887 and the issue is resolved in software versions 5.2.3 and later.

The short–term workaround for preventing this issue is as follows:

1. Issue the command **set option errport enable** to disable the error–disabled feature.
2. Re–enable all error–disabled ports using the **set port enable** *mod_num/port_num* command.

   Example: **set port enable 3/1**

3. Clear the MAC address table using the **clear cam dynamic** command to restore the dynamically learned MAC addresses.

How To Recover a Port From errDisable State

At this point in the document, we provide two examples of how you might encounter an error–disabled port and how to fix them; a brief discussion of three other reasons that a port could become error–disabled; and a summary of the commands discussed relating to error–disabled ports. The specific examples shown below for these issues are easy to duplicate in a lab environment.

Use these steps in order to recover a port from errDisable state:

1. Version of Software Used in this Document

   The **show version** command displays the software version the switch is running for this document. This is here just to show what version of CatOS we were using for this test and what modules were involved.

```
Cat5500> (enable) show version
WS-C5500 Software, Version McpSW: 5.4(2) NmpSW: 5.4(2)
Copyright (c) 1995-2000 by Cisco Systems
NMP S/W compiled on Apr  7 2000, 16:59:29
MCP S/W compiled on Apr 07 2000, 16:49:24

System Bootstrap Version: 5.1(1)

Hardware Version: 1.3  Model: WS-C5500  Serial #: 069041642

Mod Port Model       Serial #  Versions
--- ---- ----------  --------- ---------------------------------------
1   0    WS-X5540    013459824 Hw : 1.1
                               Fw : 5.1(1)
                               Fw1: 5.1(1)
                               Sw : 5.4(2)
                               Sw : 5.4(2)
11  24   WS-X5225R   012121634 Hw : 3.1
                               Fw : 4.3(1)
                               Sw : 5.4(2)

        DRAM                   FLASH                   NVRAM
Module Total   Used    Free    Total   Used    Free    Total Used  Free
------ ------- ------- ------- ------- ------- ------- ----- ----- -----
1       32768K 18567K  14201K   8192K   4171K   4021K   512K  179K  333K

Uptime is 0 day, 0 hour, 4 minutes

Cat5500> (enable) show module
Mod Slot Ports Module-Type              Model               Status
--- ---- ----- ------------------------ ------------------- --------
1   1    0     Supervisor IIG           WS-X5540            ok
15  1          Route Switch Feature Card
11  11   24    10/100BaseTX Ethernet    WS-X5225R           ok

Mod Module-Name        Serial-Num
--- ------------------ --------------------
1                      00013459824
11                     00012121634

Mod MAC-Address(es)                        Hw     Fw         Sw
--- -------------------------------------- ------ ---------- ------------------
1   00-90-ab-28-d0-00 to 00-90-ab-28-d3-ff 1.1    5.1(1)     5.4(2)
5   00-10-7b-7c-09-d4 to 00-10-7b-7c-09-df 3.0    3.1(1)     5.4(2)
```

```
6    00-e0-1e-6c-80-da to 00-e0-1e-6c-80-dc 1.0    4.1(1)    5.4(2)
8    00-10-7b-44-16-40 to 00-10-7b-44-16-57 1.3    3.1(1)    5.4(2)
10   00-10-7b-0c-32-d0 to 00-10-7b-0c-32-db 2.0    3.1(1)    5.4(2)
11   00-50-a2-f4-e4-50 to 00-50-a2-f4-e4-67 3.1    4.3(1)    5.4(2)
```

2. How to Determine if Ports are in the errDisable State

   You can enter the **show port** command in order to determine if your port has been error–disabled.
   This is an example of an active port; further below is the same port in the error–disabled state.

   ```
   Cat5500> (enable) show port 11/1
   Port  Name          Status     Vlan       Level  Duplex Speed Type
   ----- -------       ---------- ---------- ------ ------ ----- ------------
   11/1                connected  1                 normal a-half a-100 10/100BaseTX


   Cat5500> (enable) show port 11/1
   Port  Name          Status     Vlan       Level  Duplex Speed Type
   ----- -------       ---------- ---------- ------ ------ ----- ------------
   11/1                errdisable 1                 normal  auto  auto 10/100BaseTX
   ```

   Note: When a port is error–disabled, the LED associated with the port on the front panel is solid
   orange.

3. How to Determine the Reason for the Error–Disabled State (console messages, syslog, show
   errdisable–timeout)

   When the switch puts a port in the error–disabled state, it sends a message to the console and
   describes why the port was disabled. These are two sample messages that show why a port is
   disabled: one from the portfast BPDU–guard feature, and another from an EtherChannel
   configuration problem.

   ```
   2000 May 09 19:09:18 %SPANTREE-2-RX_PORTFAST:Received BPDU on PortFast
   enable port. Disabling 11/1

   2000 May 09 19:09:18 %PAGP-5-PORTFROMSTP:Port 11/1 left bridge port


   2000 May 09 19:22:11 %SPANTREE-2-CHNMISCFG: STP loop – channel 11/1-2
   is disabled in vlan 1

   2000 May 09 19:22:11 %PAGP-5-PORTFROMSTP:Port 11/1 left bridge port 11/1-2
   ```

   Note: The messages do not explicitly state errDisable or error–disabled; however, they do indicate
   that the switch is disabling the port. After the console messages are generated, they are not saved,
   unless you utilize a syslog server in your network. If you configure the switch to send these messages
   to a syslog server, then you will have a more permanent record of when and why the port was
   disabled. For information on how to configure your switch to send messages to a syslog server, see
   the document Configuring System Message Logging in the CatOS 5.4 Configuration Guide.

   If you are running CatOS 5.4(1) or later, there is a feature called errdisable–timeout which, if enabled,
   tells you why a port was disabled. This is an example

   ```
   Cat5500> (enable) show errdisable-timeout
   ErrDisable Reason   Timeout Status Port ErrDisable Reason
   ------------------- -------------- ---- ----------------
   bpdu-guard          enable         11/1 bpdu-guard
   channel-misconfig   disable
   duplex-mismatch     disable
   udld                disable
   other               disable
   ```

```
Interval: 30 seconds
```

4. How to Correct the Problem. After you discover why the ports were disabled, you should first fix the root problem, then reenable the port.

    a. Fix the Root Problem

        This depends on what the triggering problem actually is. There are numerous things that could trigger the shutdown. These are some of the most noticeable and common causes.

        ◊ EtherChannel Misconfiguration

        For EtherChannel to work, the ports involved must have consistent configurations; the same VLAN, same trunk mode, same speed, same duplex, and so forth. Most of the configuration differences within a switch are caught and reported when you create the channel. In some situations, usually when you use the ON mode (as opposed to auto or desirable), everything can be consistent on one switch so that switch starts channeling. But, the connected neighboring switch cannot be set the same and can cause the first switch to become error–disabled. If both of the switches support Port Aggregation Protocol (PAgP), you can configure the channel modes on each switch to be desirable instead of on in order to avoid this problem.

        ◊ Duplex Mismatch

        Duplex mismatches are common because of failures to auto–negotiate speed and duplex properly. Unlike with half–duplex, which must wait until no other devices are transmitting on the same LAN segment, a full–duplex device will transmit whenever it has something to send, regardless of other devices. If this transmission occurs while the half–duplex device is transmitting, the half–duplex device will consider this either a collision (during the slot time), or a late collision (after the slot time). Since the full–duplex side never expects collisions, it will never realize that it must retransmit that dropped packet. A low percentage rate of collisions are normal with half–duplex, but not with full–duplex. If the switch port receives a lot of late collisions, this usually indicates a duplex mismatch problem; make sure ports on both sides of the cable are set to the same speed and duplex. The **show port** command will tell you the speed and duplex for Catalyst switch ports. Later versions of Cisco Discovery Protocol (CDP) can warn you about a duplex mismatch before the port is actually put in error–disable state. In addition, there may be settings on a NIC card that cause the problem (things like auto polarity features – if in doubt, turn them off). If you have multiple NIC cards from a vendor and they all appear to have the same problem, check the manufacturer's web site for release notes and make sure you have the latest drivers from the NIC manufacturer. Other causes for late collisions include a bad NIC (with physical problems, not just configuration problems), a bad cable, or a cable segment that is too long.

```
2000 May 09 19:19:09 %CDP-4-DUPLEXMISMATCH:Full/half duplex
mismatch detected on port 11/3
```

        ◊ BPDU Port–Guard

        Some newer versions of switch software monitor if portfast is enabled on a ports. A port using portfast should be connected to an end–station, not to devices that generate STP packets called BPDUs. If the switch notices a BPDU coming in a port that has portfast enabled, it will put the port in errDisable mode.

        ◊ UDLD

        UDLD is a protocol on some new versions of software that discovers if communication over a link is one–way only, and therefore partially broken. A

damaged fiber cable or other cabling/port issue could cause this one−way only communication. Spanning tree loops can occur with this problem. UDLD allows the port to detect a unidirectional link, and can be configured to put a port in errDisable state when it detects this condition.

◊ Other

Any process within the switch that recognizes a problem with the port can place it in the **error−disable state**. Look at the console messages or the message that were sent to a syslog server that state why the port is being shut down. Also, if the errdisable−timeout feature is enabled (minimum CatOS 5.4(1)), the show errdisable−timeout will tell you the general reason that the port was disabled.

b. Reenable the Port

After you fix the root problem, the ports will still be disabled; you must reenable the ports. This can be done manually using the **set port enable** command.

```
Cat5500> (enable) set port enable 11/1-2
Ports 11/1-2 enabled.
```

If you have CatOS 4.2(2) or later, one can use the **set option errport** command as described above to prevent ports from becoming error−disabled. Since you are not actually fixing the source of the problem this can be risky. If you have CatOS 5.4(1) or later, you can use the **errdisable−timeout** command to automatically reenable the ports as described in the next section.

5. How to Reenable the Port Automatically Using errdisable−timeout − CatOS 5.4(1)

The **errdisable−timeout** command allows you to selectively pick which type of errors will automatically reenable the ports after a specified amount of time. The output shows the default state which is errdisable−timeout disabled (not active) for all five possible conditions. If any condition was enabled, the ports with this condition would be reenabled after 30 seconds.

```
Cat5500> (enable) show errdisable-timeout
ErrDisable Reason   Timeout Status
------------------- --------------
bpdu-guard          disable
channel-misconfig   disable
duplex-mismatch     disable
udld                disable
other               disable

Interval: 30 seconds
```

To turn **errdisable−timeout** on, use the following command to choose the errdisable conditions.

```
Cat5500> (enable) set errdisable-timeout enable ?
  bpdu-guard                 BPDU Port-guard
  channel-misconfig          Channel misconfiguration
  duplex-mismatch            Duplex Mismatch
  udld                       UDLD
  other                      Reasons other than the above
  all                        Apply errDisable timeout to all reasons

Cat5500> (enable) set errdisable-timeout enable bpdu-guard
Successfully enabled errdisable-timeout for bpdu-guard.

Cat5500> (enable) set errdisable-timeout interval 30
Successfully set errdisable timeout to 30 seconds.
```

A nice feature of this command is that if you enable errdisable–timeout, it will list generally why the ports have been put into error–disable state. For more detailed descriptions, you must refer to the messages displayed at the time of occurrence. Remember that the first step in fixing the error–disable condition is to fix the original error that brought about the shutdown. Notice below that the reason port 11/1 was shut down was because of the bpdu–guard feature.

```
Cat5500> (enable) show errdisable-timeout
ErrDisable Reason   Timeout Status   Port ErrDisable Reason
------------------- --------------   ---- -----------------
bpdu-guard          enable           11/1 bpdu-guard
channel-misconfig   disable
duplex-mismatch     disable
udld                disable
other               disable

Interval: 30 seconds
```

Here is an example of what displays when the switch reenables a port because of the errdisable–timeout function.

```
Cat5500> (enable)
2000 May 09 19:17:27 %MGMT-5-ERRDISPORTENABLED:Port 11/1 err-disabled
by bpdu-guard enabled by errdisable timeout
```

6. What if You Reenable the Port Without Fixing the Problem?

If you reenable the port without fixing the problem, the ports will just become error–disabled again. This will continue over and over again until you solve the real problem. Notice the three messages below. In the first one, the switch describes disabling port 11/1 because it received a BPDU on a port that is enabled for portfast (this is an error causing situation if bpdu–guard is on). After 25 seconds, the port is automatically reenabled by the errdisable–timeout feature. Then, four seconds later, the port is error–disabled again because the real problem was never fixed.

```
2000 May 09 19:17:33 %SPANTREE-2-RX_PORTFAST:Received BPDU on PortFast
enable port. Disabling 11/1

2000 May 09 19:17:58 %MGMT-5-ERRDISPORTENABLED:Port 11/1 err-disabled by
bpdu-guard enabled by errdisable timeout

2000 May 09 19:18:02 %SPANTREE-2-RX_PORTFAST:Received BPDU on
PortFast enable port. Disabling 11/1
```

The benefit of having to manually reenable the ports is that it reminds you and prompts you to deal with the real problem.

7. Can I Eliminate Ports From Becoming Error–Disabled Due to Collisions

Here is an example of how to keep the switch from error–disabling a port due to excessive or late collisions. The **set option errport** command became available in CatOS release 4.2(2). Again, please remember that this should be used only as a "stop–gap" type of measure. It keeps the ports from being error–disabled due to collisions but can leave you vulnerable to collisions that would normally cause the switch to shut the port down. When you execute this command, it will stop the switch from disabling the port due to collisions.

```
Cat5500> (enable) set option errport enable
Error port option is enabled

Cat5500> (enable) show option errport
Option errport   : enabled
```

Here is an example of how to return to the default state, which is allowing the switch to error–disable a port.

```
Cat5500> (enable) set option errport disable
Error port option is disabled

Cat5500> (enable) show option errport
Option errport   : disabled
```

The command **show option errport** will show the current mode the error–disable feature is in. Also, the **set option errport enable** command does not fix the cause of the errors; it only keeps the port from being shut down because of the errors. There still exists the possibility that errDisable ports could affect other ports on the module if the errors persist or become drastic. So, you should use this command only if you understand that these errors could potentially cause larger problems within the switch module and you are willing to take those risks.

Fixing an Error–Disabled Port – Examples

In this section, we present two examples of fixing an error–disabled port.

Error–Disabled Because of BPDU Port–Guard

A new feature starting in CatOS 5.4(1) allows the switch to monitor ports that have portfast enabled. A port using portfast must only be connected to an end station (such as a workstation or server), *not* to devices that generate spanning tree BPDUs, like switches, or bridges and routers doing bridging. If the switch receives a spanning tree BPDU on a port that has portfast enabled, it will put the port in errDisable mode in order to guard against potential loops. Portfast assumes that a port on a switch has no possibility of generating a physical loop, and thus skips the initial spanning tree checks for that port, avoiding end stations from timing out on boot up. Portfast must be implemented carefully by the network administrator; on ports where portfast has been enabled, BPDU guard helps ensure that the LAN stays loop–free.

Here is how you turn this feature on. This example was picked because it is easy to create an error–disable situation.

```
Cat5500> (enable) set spantree portfast bpdu-guard enable
Spantree portfast bpdu-guard enabled on this switch.
```

Our Catalyst 5500 switch is connected to another switch (a 6509) that we made to be the root of the spanning tree. The 6509 will be sending us BPDUs every 2 seconds (using default spanning tree settings). When we enable portfast on the 5500 switch port, the bpdu–guard feature will watch for BPDUs coming in on this port. When a BPDU comes into the port, meaning that a non–end device has been detected off of that port, the bpdu–guard feature will shut the port down to avoid possible Spanning tree loop.

```
Cat5500> (enable) set spantree portfast 11/1 enable

Warning: Spantree port fast start should only be enabled on ports connected
to a single host.  Connecting hubs, concentrators, switches, bridges, etc. to
a fast start port can cause temporary spanning tree loops.  Use with caution.
Spantree port 11/1 fast start enabled.

Cat5500> (enable)

2000 May 09 19:09:18 %SPANTREE-2-RX_PORTFAST:Received BPDU on
PortFast enable port. Disabling 11/1

2000 May 09 19:09:18 %PAGP-5-PORTFROMSTP:Port 11/1 left bridge port 11/1
```

In the message above the switch indicated that it received a BPDU on a portfast enabled port, so it is shutting down port 11/1. When we look at the status of the port, it reads errDisable.

```
Cat5500> (enable) show port 11/1

Port  Name              Status     Vlan       Level  Duplex Speed Type
----- ----------------- ---------- ---------- ------ ------ ----- ------------
11/1                    errdisable 1          normal  auto  auto 10/100BaseTX
```

To fix these situations, we must address the underlying problem, and then reenable the port. Since this is a port with an improper connection (portfast enabled and connected to another switch), we will turn off the portfast feature. Again, portfast is only supposed to be used on ports connected to end stations.

```
Cat5500> (enable) set spantree portfast 11/1 disable
Spantree port 11/1 fast start disabled.
```

Even though we fixed the root of the problem, notice that the port is still in error–disable state. If you looked at the port LED, it would still be orange. We must reenable the port before it will become active again.

```
Cat5500> (enable) show port 11/1

Port  Name              Status     Vlan       Level  Duplex Speed Type
----- ----------------- ---------- ---------- ------ ------ ----- ------------
11/1                    errdisable 1          normal  auto  auto 10/100BaseTX
```

In the following example we reenable the port manually using the **set port enable** command. Now the port will return to normal status.

```
Cat5500> (enable) set port enable 11/1
Port 11/1 enabled.

Cat5500> (enable) show port 11/3
Port  Name              Status     Vlan       Level  Duplex Speed Type
----- ----------------- ---------- ---------- ------ ------ ----- ------------
11/1                    connected  1          normal a-half a-100 10/100BaseTX
```

Error–Disabled Because of EtherChannel Misconfiguration

Here is another common error–disable situation that can occur on ports capable of EtherChannel. If one switch is configured for EtherChannel and the other is not, it can cause the spanning tree process to shut down the channeled ports on the side configured for EtherChannel. In this scenario we connected two crossover cables from the 5500 switch to another switch. We turned on EtherChannel on the 5500 switch using the command **set port channel 11/1–2** on. The ON mode of EtherChannel does not send PAgP packets to negotiate with the other side before channeling; it just assumes the other side is channeling. In addition, we did not turn EtherChannel on for the other switch; we left these ports as individual unchanneled ports. If left in this state for a minute or so, STP on the 5500 will think there is a loop. This will cause the channeling ports to be put in error–disable state. Notice below that a loop was detected and the ports were disabled. The **show port channel** command shows that the ports are no longer channeling; and, when we look at one of the ports involved, we see its status is errdisable.

```
Cat5500> (enable)
2000 May 09 19:20:02 %PAGP-5-PORTTOSTP:Port 11/1 joined bridge port 11/1-2
2000 May 09 19:20:27 %PAGP-5-PORTTOSTP:Port 11/2 joined bridge port 11/1-2
2000 May 09 19:22:11 %SPANTREE-2-CHNMISCFG: STP loop -
channel 11/1-2 is disabled in vlan 1

2000 May 09 19:22:11 %PAGP-5-PORTFROMSTP:Port 11/1 left bridge port 11/1-2
2000 May 09 19:22:11 %PAGP-5-PORTFROMSTP:Port 11/2 left bridge port 11/1-2
```

```
Cat5500> (enable) show port channel
No ports channeling
```

The EtherChannel was torn down because the ports were placed in error−disable on this switch.

```
Cat5500> (enable) show port 11/1
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
11/1                     errdisable 1          normal  auto   auto 10/100BaseTX
```

In order to determine what the problem was, we need to look at the error message. The message said that the EtherChannel encountered a spanning tree loop. As we know from the paragraph above, this can occur when one device (our switch in this case) has EtherChannel turned on manually by using the ON mode (as opposed to desirable) and the other connected device (the other switch in this case) does not have EtherChannel turned on at all. One way to fix the situation is to set the channel mode to desirable on both sides of the connection, and then reenable the ports. This will cause each side to form a channel only if they both agree to channel. If they do not agree to channel, they will continue to function as normal ports.

Note: For a list of things that can cause EtherChannel misconfiguration errors, look in the Configuration Guide on EtherChannel for the CatOS version you are using. The newer releases have specific sections of the Configuration Guide titled Configuring Fast EtherChannel and Gigabit EtherChannel that list the dependencies for a channel to form correctly, including the channel modes to configure.

```
Cat5500> (enable) set port channel 11/1-2 desirable non-silent
Port(s) 11/1-2 are assigned to admin group 21.
Port(s) 11/1-2 channel mode set to desirable.

Cat5500> (enable) show port 11
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
11/1                     errdisable 1          normal  auto   auto 10/100BaseTX
11/2                     errdisable 1          normal  auto   auto 10/100BaseTX
```

Notice that even though we turned off the EtherChannel feature and set the EtherChannel mode to desirable, the ports are still disabled. We have corrected the cause of the problem, but now we must reenable the ports before we can use them.

```
Cat5500> (enable) set port enable 11/1-2
Ports 11/1-2 enabled.

Cat5500> (enable) show port 11
Port  Name               Status     Vlan       Level  Duplex Speed Type
----- ------------------ ---------- ---------- ------ ------ ----- ------------
11/1                     connected  1          normal a-full a-100 10/100BaseTX
11/2                     connected  1          normal a-full a-100 10/100BaseTX

Cat5500> (enable) show port channel 11/1
Port            Status     Channel              Admin Ch
                           Mode                 Group Id
-----           ---------- -------------------- ----- -----
11/1            connected  desirable non-silent  21   833
11/2            connected  desirable non-silent  21   833

Port  Device-ID                       Port-ID                  Platform
----- ------------------------------- ------------------------ ----------------
11/1  TBA04090489(Cat6000)      5/13                           WS-C6506
11/2  TBA04090489(Cat6000)      5/14
----- ------------------------------- ------------------------ ----------------
```

Verify the Configuration

- **show version** to display the version of the software being used on the switch
- **show module** to display which modules are used on the switch
- **show port** to view the current status of the switch port
- **show option errport** to view the status of the set option errport command
- **show errdisable−timeout** to display the current settings of the errdisable−timeout feature and the reason why any ports are currently error−disabled

Troubleshoot the Configuration

- **show port** to view the current status of the switch port
- **show port channel** to view the current status of the EtherChannel
- **show option errport** to view the status of the **set option errport** command
- **set option errport disable** to allow the switch to disable any ports that have errors which the operating system deems worthy of being disabled. This is the default state and would only be different if someone had previously issued the **set option errport enable** command
- **show errdisable−timeout** to display the current settings of the errdisable−timeout feature and the reason why any ports are currently error−disabled
- **set errdisable−timeout** can be used to help determine why a port was error−disabled (used in conjunction with the **show errdisable−timeout** command)

Command Summary

Syntax:	show version
As used in this document:	show version

Syntax:	show module [*mod_num*]
As used in this document:	show module

Syntax:	show port [*mod_num*[*/port_num*]]
As used in this document:	show port 11/1 show port 11

Syntax:	show port channel *[mod_num[/port_num]]* [statistics	info [spantree	trunk	protocol	gmrp	gvrp	qos]]
As used in this document:	show port channel

Syntax:	set port channel *port_list* mode {on	off	desirable	auto} [silent	non−silent]
As used in this document:	set port channel 11/1−2 desirable non−silent

Syntax:	set port enable *mod_num/port_num*
As used in this document:	set port enable 11/1−2

Syntax:	show errdisable–timeout
As used in this document:	show errdisable–timeout

Syntax:	set errdisable–timeout [enable|disable] *[bpdu–guard | channel–misconfig | duplex–mismatch | udld | other]*
As used in this document:	set errdisable–timeout enable bpdu–guard

Syntax:	set errdisable–timeout interval seconds
As used in this document:	set errdisable–timeout interval 30

Syntax:	set spantree portfast *mod_num/port_num* {enable | disable}
As used in this document:	set spantree portfast 11/1 enable set spantree portfast 11/1 disable

Syntax:	set spantree portfast bpdu–guard {enable | disable}
As used in this document:	set spantree portfast bpdu–guard enable

Related Information

- **Errdisable Port State Recovery on the Cisco IOS Platforms**
- **Configuring Fast EtherChannel and Gigabit EtherChannel (CatOS 5.4)**
- **Release Notes for Catalyst 5000 Family Software Release 4.x**
- **EtherChannel Technology Support**
- **Using Portfast and Other Commands to Fix Workstation Startup Connectivity Delays**
- **Technical Support & Documentation – Cisco Systems**