

# Troubleshoot Uni-Directional Link Detection Errors on Nexus Switches



Document ID: 118908

Contributed by Dez Hill, Cisco TAC Engineer.  
Apr 22, 2015

## Contents

### Introduction

#### Prerequisites

Requirements

Components Used

#### Background Information

#### UDLD Error Conditions

Empty Echo

Tx-Rx Loop

Neighbor Mismatch

Sudden Cessation of UDLD Frames

#### Troubleshoot UDLD Error Conditions

Useful Commands

Useful TAC Information

## Introduction

This document describes how to troubleshoot Uni-Directional Link Detection (UDLD) error messages on a Cisco Nexus 7000 Series switch.

## Prerequisites

### Requirements

Cisco recommends that you have a basic knowledge of these topics:

- Cisco Nexus Operating System (NX-OS)
- Basic UDLD operations

### Components Used

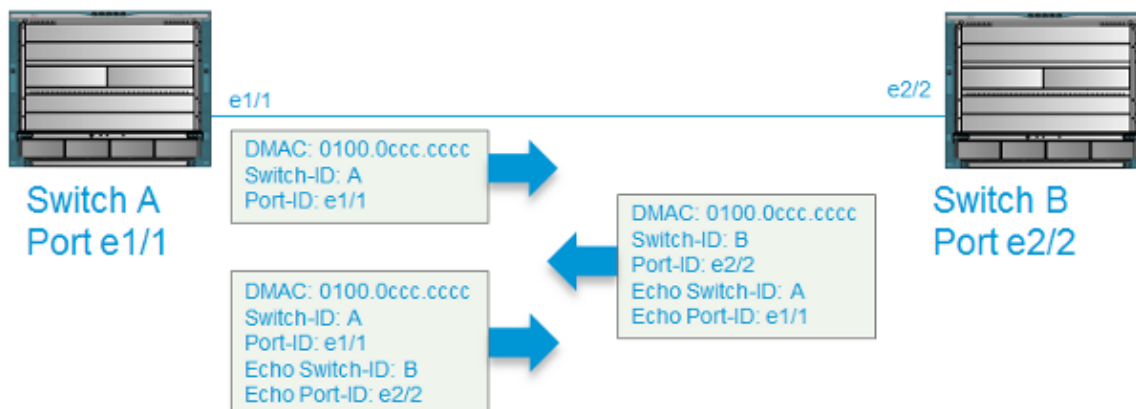
The information in this document is based on these software and hardware versions:

- Cisco Nexus 7000 Series switches
- Cisco NX-OS Version 6.2(10)

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.

## Background Information

The ports exchange UDLD packets during the UDLD detection process, to include the originator switch-ID and the originator port-ID. Once a UDLD packet is received, the switch echoes the peer switch-ID and port-ID back to the peer. Once the switches exchange echo packets, a bi-directional relationship is formed.



The UDLD error conditions exist when the switch does not receive the expected information from its UDLD peer.

This document describes these UDLD error conditions and how to troubleshoot them:

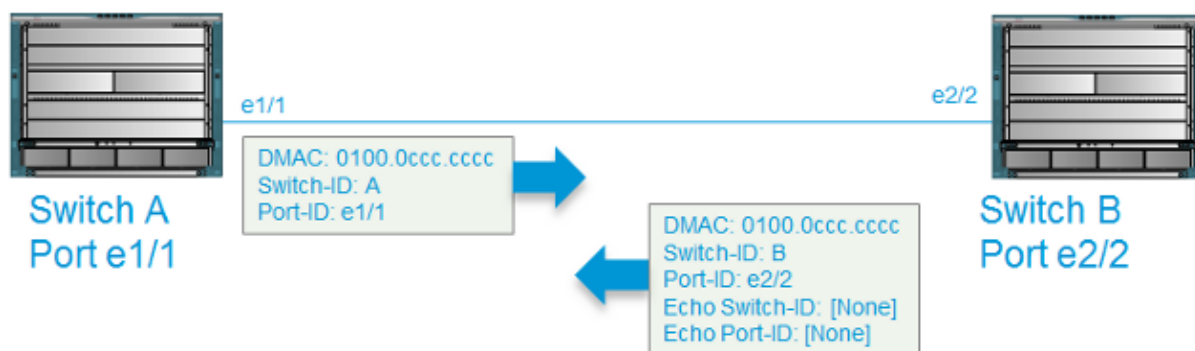
- Empty-echo
- Transmit-Receive (Tx-Rx) Loop
- Uni-direction
- Neighbor mismatch
- Sudden cessation of UDLD frames

## UDLD Error Conditions

This section describes the various types of UDLD error conditions and some probable causes.

### Empty Echo

This condition is present when *Switch-A* receives a UDLD frame from *Switch-B* without the expected echo of the Switch-A switch-ID and port-ID.



When an empty-echo is detected, the UDLD performs these actions:

<i>Mode</i>	<i>Action</i>
Normal Mode	err-disable port
Aggressive Mode	err-disable port

These syslog messages are then generated:

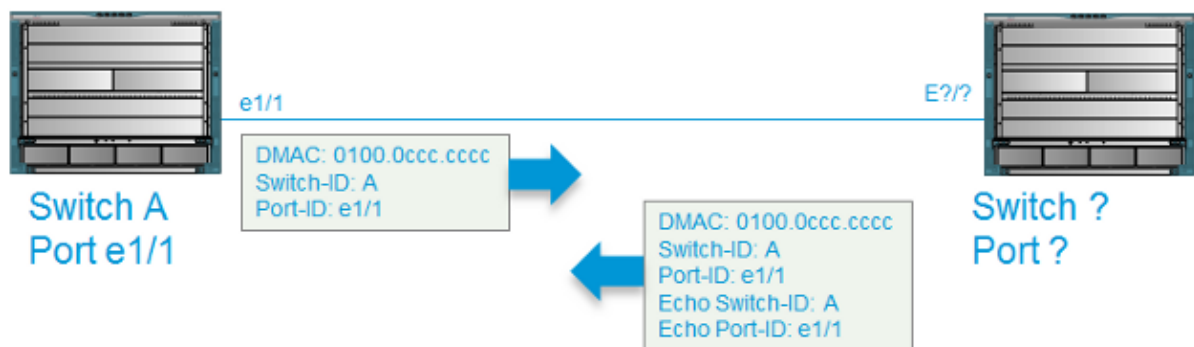
```
2015 Mar 19 11:57:56.155 N7kA ETHPORT-2-IF_DOWN_ERROR_DISABLED Interface Ethernet1/2
is down (Error disabled. Reason:UDLD empty echo)
2015 Mar 19 11:57:56.186 N7kA ETH_PORT_CHANNEL-5-PORT_INDIVIDUAL_DOWN individual port
Ethernet1/2 is down
2015 Mar 19 11:57:56.336 N7kA ETHPORT-2-IF_DOWN_ERROR_DISABLED Interface Ethernet1/2
is down (Error disabled. Reason:UDLD empty echo)
```

Here are some possible causes for this condition

- The UDLD bi-directional relationship has timed out on Switch-B because it does not receive the UDLD frames from Switch-A.
- Switch-B received the UDLD frames from Switch-A but did not process them.
- Switch-A did not send the UDLD frames to Switch-B.

## Tx-Rx Loop

This condition occurs when a UDLD frame is received on the same port from which it was transmitted.



When a Tx-Rx loop is detected, UDLD performs these actions:

<i>Mode</i>	<i>Action</i>
Normal Mode	err-disable port
Aggressive Mode	err-disable port

These syslog messages are then generated:

```
2015 Mar 20 14:52:30 N7kA %ETHPORT-2-IF_DOWN_ERROR_DISABLED: Interface Ethernet17/5
is down (Error disabled. Reason:UDLD Tx-Rx Loop)
2015 Mar 20 14:52:30 N7kA %ETHPORT-2-IF_DOWN_ERROR_DISABLED: Interface Ethernet17/5
```

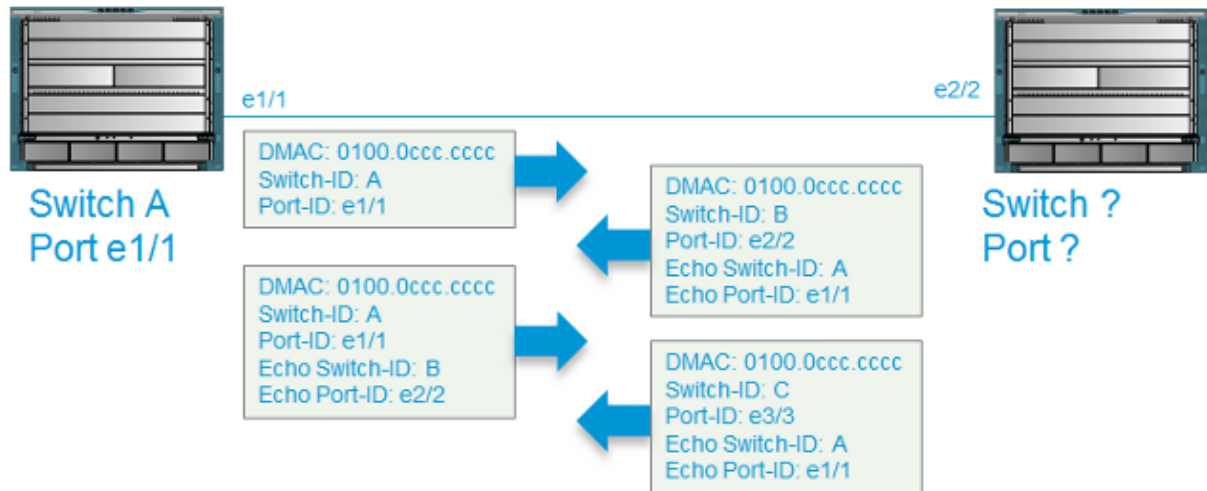
is down (Error disabled. Reason:UDLD Tx-Rx Loop)

Here are some possible causes for this condition:

- There might be incorrect wiring or a physical media issue.
- The intermediate devices reflect the frames back to the sending port.

## Neighbor Mismatch

This condition is present when Port-A on Switch-A receives a frame from a port other than that with which it already formed a UDLD bi-directional relationship.



When a neighbor mismatch is detected, UDLD performs these actions:

<i>Mode</i>	<i>Action</i>
Normal Mode	err-disable port
Aggressive Mode	err-disable port

These syslog messages are then generated:

```
2015 Mar 21 10:23:05.598 N7kA %ETHPORT-2-IF_DOWN_ERROR_DISABLED: Interface Ethernet3/21
is down (Error disabled. Reason:UDLD Neighbor mismatch)
2015 Mar 21 10:24:07.065 N7kA %ETHPORT-2-IF_DOWN_ERROR_DISABLED: Interface Ethernet3/21
is down (Error disabled. Reason:UDLD Neighbor mismatch)
```

Here are some possible causes for this condition:

- The UDLD port in question is a member of a port-channel on which a member port has changed states.
- There is an intermediate device between the two ports that formed the bi-directional relationship.

## Sudden Cessation of UDLD Frames

This condition is present when a port that has formed a bi-directional relationship does not receive a UDLD frame during the time-out interval (50 seconds by default).

When this condition is detected, the UDLD performs these actions:

<i>Mode</i>	<i>Action</i>
Normal Mode	UDLD marks port as <i>Undetermined</i> , and the port continues to function in accordance with its spanning-tree port state
Aggressive Mode	err-disable port

## Troubleshoot UDLD Error Conditions

This section describes general troubleshooting steps that you should complete if you encounter a UDLD *error-disabled* port.

Since UDLD errors indicate physical layer faults, it is appropriate to troubleshoot at the physical layer. When UDLD error messages are encountered, consider these questions:

- Does the error persist if the Small Form-Factor Pluggable Transceiver (SFP) is replaced?
- Does the error persist if the cable is replaced?
- Does the error persist if the connection is moved to a different physical port on the switch?

## Useful Commands

Use this command in order to restore all of the ports that have been placed into *error-disable* mode by the UDLD:

```
N7KA(config)# udld reset
```

Use this command in order to verify the bi-directional relationship:

```
N7KA-NORTH-AGG(config-if)# show udld eth 3/4
```

```
Interface Ethernet3/4
-----
Port enable administrative configuration setting: enabled
Port enable operational state: enabled
Current bidirectional state: bidirectional
Current operational state: advertisement - Single neighbor detected
Message interval: 7
Timeout interval: 5
```

```
Entry 1
```

```
-----
Expiration time: 39
Cache Device index: 1
Current neighbor state: bidirectional
Device ID: JAF1620ABAB
Port ID: Ethernet3/12
Neighbor echo 1 devices: JAF1617BACD
Neighbor echo 1 port: Ethernet3/4
```

```
Message interval: 15
Timeout interval: 5
CDP Device name: N7KB-SOUTH-AGG(JAF1620ABAB)
```

```
Last pkt send on: 400096, Aug 6 13:58:52 2014
Probe pkt send on: 400096, Aug 6 13:58:52 2014
Echo pkt send on: 395799, Aug 6 13:58:43 2014
Flush pkt send on: None.
```

```
Last pkt rcv on: 740333, Aug 6 13:58:52 2014
Probe pkt rcv on: 740333, Aug 6 13:58:52 2014
Echo pkt rcv on: 730454, Aug 6 13:58:43 2014
Flush pkt rcv on: None.
```

```
Deep pkt inspections done: None.
Mismatched if index found: None.
Deep pkt inspection drops: None.
```

Use this command in order to verify error counters on the physical interfaces, which determines whether the UDLD frames are dropped due to physical layer hardware faults:

```
RTP-Agg1# show interface ethernet 4/1 | i error/CRC/discard/drop
 0 runts 0 giants 0 CRC/FCS 0 no buffer
 0 input error 0 short frame 0 overrun 0 underrun 0 ignored
 0 watchdog 0 bad etype drop 0 bad proto drop 0 if down drop
 0 input with dribble 0 input discard
 0 output error 0 collision 0 deferred 0 late collision
 0 lost carrier 0 no carrier 0 babble 0 output discard
```

Use this command in order to check the CPU utilization, which determines whether high CPU utilization prevents the UDLD frames from being processed:

```
N7K-A# show system resources
Load average: 1 minute: 0.17 5 minutes: 0.25 15 minutes: 0.20
Processes : 1993 total, 1 running
CPU states : 0.18% user, 0.81% kernel, 98.99% idle
```

## Useful TAC Information

This section describes the outputs that you should collect before you restore the link (if the circumstances permit) in order to provide the Cisco Technical Assistance Center (TAC) the best chance to diagnose the root cause of the link being placed into *error-disabled* mode by the UDLD:

- **show tech-support lacp all** (if the failed interface is a member of a Link Aggregation Control Protocol (LACP) portchannel)
- **show tech-support module <x>** (where *x* is the module where the UDLD error is detected)
- **show tech-support ethpm**
- **show tech-support udd**
- **show udd internal event-history errors**
- **show udd internal event-history msgs | grep -a 3 -b 3 L2\_RX\_DATA**
- **show udd internal event-history ethernet <x/y>**
- **show log logfile | grep UDLD**
- **show log logfile | grep Ethernet<x/y>**
- **show processes cpu history**
- **show interface ethernet <x/y>**
- **show hardware internal errors module <x>**
- **show interface counters errors module <x>**

