

Troubleshooting Invalid Cells on ATM

Document ID: 10454

Introduction

Prerequisites

Requirements

Conventions

Invalid Cells on the LS1010 and 8510

Invalid Cells on the Catalyst 8540

Invalid Cells on Ethernet Interfaces

NetPro Discussion Forums – Featured Conversations

Related Information

Introduction

Cisco ATM switch routers, which include the LS1010 and the Catalyst 8500 series, use a switching fabric with a shared-memory architecture. In some cases, the switch drops cells and increments the invalid cells counter, as shown in this output of one of the platform-dependent commands:

- **show switch fabric** – Catalyst 8540
- **show controller atm 2/0/0** – LS1010
- **show controller controller0** – Catalyst 8510

These switches increment the invalid cells counter when they discard an ATM cell that has a valid header error checksum (HEC) value but arrives on a non-existent virtual circuit (VC). Possible reasons include these:

- Corrupted cell header
- Incomplete or no configuration of that VC in the switch fabric
- Misconfiguration: for example, if you configure a virtual path identifier/virtual channel identifier (VPI/VCI) pair only on a router and not on the attached ATM switch, cells transmitted on this VC from the router are considered invalid by the switch.

This document explains the invalid cells counter on Cisco ATM campus switches and provides tips on how to troubleshoot incrementing invalid cells.

Prerequisites

Requirements

There are no specific requirements for this document.

Conventions

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

Invalid Cells on the LS1010 and 8510

The output of **show controllers atm 2/0/0 (or 13/0/0) or atm 0** (dependent on software version and chassis) on the LS1010 or Catalyst 8510 prints a table of the most recently received invalid cells. The **show controllers atm** command is cleared on read, which means that the invalid cells counter is cleared when the show command is executed. If you do not receive invalid cells continuously on an interface, the invalid cells counter will show as zero when you read a subsequent time.

Note: *!-- Comments are in italics and refer to the line above the comment.*

```
cisco#show controllers atm 2/0/0
MMC Switch Fabric (idb=0x607F7DE0)
  Key: discarded cells - # cells discarded due to lack of resources
                        or policing (16-bit)
      invalid cells - # good cells that came in on a non-existent conn.
      memory buffer - # cell buffers currently in use
      RXcells      - # rx cells (16-bit)
      TXcells      - # tx cells (16-bit)
      RHEC         - # cells with HEC errors
      TPE         - # cells with memory parity errors

discarded cells = 0
invalid cells   = 132
! -- The displayed invalid cells value is cleared on read.
memory buffer   = 0
port  type    status  RXcells TXcells RHEC   TPE  PACE_I  PACE_M  PACE_X  PACE_Y
0/0/0 155MBPS xytrpm 0xD00D 0x2420 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0/0/1 155MBPS xytrpm 0x969D 0x2DDE 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0/0/2 155MBPS xytrpm 0x43CF 0x6D9B 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0/0/3 155MBPS xytrpm 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0/1/0 155MBPS xytrpm 0xF7AC 0xE115 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0/1/1 155MBPS xytrpm 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0/1/2 155MBPS xytrpm 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0/1/3 155MBPS xytrpm 0x7969 0x3575 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
1/0/0 622MBPS xytrPm 0xB54F 0x8B73 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
2/0/0 CPU                0x9496A8 0x5EAA4D
3/0/0 155MBPS xytrpm 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
3/0/1 155MBPS xytrpm 0xFB23 0xB8FB 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
3/0/2 155MBPS xytrpm 0xC5F9 0x2319 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
3/0/3 155MBPS xytrpm 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
3/1/0 155MBPS xytrpm 0x9B0A 0x52F0 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
3/1/1 155MBPS xytrpm 0x6B08 0x2342 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
3/1/2 155MBPS xytrpm 0x7467 0x0737 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
3/1/3 155MBPS xytrpm 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
! -- Invalid cell log starts below.
Invalid Cell Log
      time stamp          port  pt clp gfc vpi  vci
41 0xBDC15C5C.0x851EB690 3/0/2 0x1 0x0 0x0 0x0 0x5
42 0xBDC15C5D.0x851EB568 3/1/1 0x1 0x0 0x0 0x0 0x5
43 0xBDC15C64.0x851EAD50 3/1/1 0x1 0x0 0x0 0x0 0x11
44 0xBDC15C65.0x851EAC28 3/1/1 0x1 0x0 0x0 0x0 0x11
45 0xBDC15C66.0x851EAB00 3/1/1 0x1 0x0 0x0 0x0 0x11
46 0xBDC15C68.0x851EA8B0 3/1/1 0x1 0x0 0x0 0x0 0x11
47 0xBDC15C69.0x851EA788 0/0/2 0x1 0x0 0x0 0x0 0x11
48 0xBDC15C6B.0x851EA538 0/0/2 0x1 0x0 0x0 0x0 0x11
49 0xBDC15C6D.0x851EA2E8 3/0/1 0x1 0x0 0x0 0x0 0x11
50 0xBDC15C6E.0x851EA1C0 3/0/1 0x1 0x0 0x0 0x0 0x11
51 0xBDC15C6F.0x851EA098 3/0/1 0x1 0x0 0x0 0x0 0x11
```

You can use these objects of the CISCO–RHINO–MIB to poll your ATM switch router for the number of invalid cells:

```
ciscoAtmSwitchInvalidCells OBJECT-TYPE
    SYNTAX Counter32
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total invalid cells of the switch."
    ::= { ciscoLS1010CpuSwitchGroup 5 }
ciscoAtmSwitchInvalidCellHeaderTable OBJECT-TYPE
    SYNTAX SEQUENCE OF CiscoAtmSwitchInvalidCellHeaderEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A list of invalid cell header entries."
    ::= { ciscoLS1010CpuSwitchGroup 6 }

ciscoAtmSwitchInvalidCellHeaderEntry OBJECT-TYPE
    SYNTAX CiscoAtmSwitchInvalidCellHeaderEntry
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "a entry of invalid cell header."
    INDEX { ciscoAtmSwitchInvalidCellHeaderIndex }
    ::= { ciscoAtmSwitchInvalidCellHeaderTable 1 }

CiscoAtmSwitchInvalidCellHeaderEntry ::= SEQUENCE {
    ciscoAtmSwitchInvalidCellHeaderIndex INTEGER,
    ciscoAtmSwitchInvalidCellHeader OCTET STRING
}

ciscoAtmSwitchInvalidCellHeaderIndex OBJECT-TYPE
    SYNTAX INTEGER(1..64)
    ACCESS not-accessible
    STATUS mandatory
    DESCRIPTION
        "A sequence number that identifies a invalid cell header
        entry in the table."
    ::= { ciscoAtmSwitchInvalidCellHeaderEntry 1 }

ciscoAtmSwitchInvalidCellHeader OBJECT-TYPE
    SYNTAX OCTET STRING(SIZE(5))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The most recently received invalid cells header.
        octet 0 is port number (0-32),
        octet 1 bit 7-5 is PTI,
        octet 1 bit 4 is CLP,
        octet 1 bit 3-0 is GFC,
        octet 2 is VPI,
        octet 3 is high byte of VCI
        octet 4 is low byte of VCI."
    ::= { ciscoAtmSwitchInvalidCellHeaderEntry 2 }
```

Invalid Cells on the Catalyst 8540

The **show switch fabric** command on the Catalyst 8540 does not print a log of the most recent invalid cells, but you can use these commands to determine on what VPI and VCI the invalid cell arrived.

1. Execute the **show switch fabric** command and determine the MSC# with incrementing invalid cells. Each of the two required switch processors in the Catalyst 8540 contains four MSC ASICs. Each ASIC forms the switch fabric for several ports.

```
8540#sh switch fabric
swc_presence_mask: 0x7
Switch mode: R_20G
Number of Switch Cards present in the Chassis: 3
SWC SLOT          SWC_TYPE          SWC_STATUS
=====
          5          EVEN          ACTIVE
          6          STANDBY         STANDBY
          7          ODD           ACTIVE
```

MMC Switch Fabric (idb=0x6142B774)

```
Key: Rej. Cells - # cells rejected due to lack of resources
                or policing (16-bit)
    Inv. Cells  - # good cells that came in on a non-existent conn.
    Mem Buffs   - # cell buffers currently in use
    RX Cells    - # rx cells (16-bit)
    TX Cells    - # tx cells (16-bit)
    Rx HEC      - # cells Received with HEC errors
    Tx PERR     - # cells with memory parity errors
```

MSC#	Rej. Cells	Inv. Cells	Mem. Buffs	Rx Cells	Tx Cells	Rx HE
MSC 0:	0	0	0	0	0	0
MSC 1:	0	0	0	0	0	0
MSC 2:	0	0	0	0	0	0
MSC 3:	0	0	0	0	0	0
MSC 4:	0	0	0	0	0	0
MSC 5:	0	0	0	0	0	0
MSC 6:	0	0	0	0	0	0
MSC 7:	0	0	0	0	0	0

[output omitted]

2. Execute the **show mmc ports** command and determine which physical ports use the particular MSC.

```
8540#show mmc ports
int a0/0/0: msc#: 0 port#: 12
int a0/0/1: msc#: 0 port#: 8
int a0/0/2: msc#: 0 port#: 4
int a0/0/3: msc#: 0 port#: 0
int a0/0/4: msc#: 0 port#: 14
int a0/0/5: msc#: 0 port#: 10
int a0/0/6: msc#: 0 port#: 6
int a0/0/7: msc#: 0 port#: 2
int a0/0/8: msc#: 1 port#: 12
int a0/0/9: msc#: 1 port#: 8
int a0/0/10: msc#: 1 port#: 4
int a0/0/11: msc#: 1 port#: 0
int a0/0/12: msc#: 1 port#: 14
int a0/0/13: msc#: 1 port#: 10
int a0/0/14: msc#: 1 port#: 6
int a0/0/15: msc#: 1 port#: 2
[output omitted]
```

3. Execute the **show mmc msc_reg all <m>** command to view details of the invalid cells. The value for "m" is the MSC number. This sample output was taken from a switch with invalid cells on MSC# 1:

```

Switch#sh mmc msc_reg all 1
gcr0[1] = 0x0000A112
...
icc[0] = 0x00000026
! -- Will be non-zero if you have invalid cells.
...
ich[1] = 0x00000D00 0x00640064
      vci:64   pti:0   clp:0   vpi:64   ssp:0   sp:D
...
! -- "vci" identifies the VCI of the last invalid cell.
! -- "vpi" identifies the VPI of the last invalid cell.
! -- "sp" identifies the port (p) on which the invalid cell was received.

```

4. Execute the **show atm vc int atm** command to check if the connection exists in the system.

Note: The invalid cells log can refer to port numbers and VPI/VCI values that do not correspond to actual port numbers and VCs. The reason is that the port interface (PIF) chip on some modules considers idle cells on some ports as invalid cells. The interface chip actually changes the default VPI of 0 for idle cells to a different value. For example, VPI/VCI 0/16 on port #1 will be changed to VPI/VCI 4/16 on the PIF chip. The interface chip drops idle cells from these ports and increments the invalid cells counter. On ports 0 and 6, the interface chip does not change the default idle cell value of VPI/VCI 0/0 since the physical VPI is the same as VPI on the interface chip.

Invalid Cells on Ethernet Interfaces

Ethernet interfaces also can experience cell drops due to invalid cells in an ATM switch router. Ethernet interfaces derive much of their local intelligence from a PIF ASIC, which segments Ethernet frames for transmission across the switch router's internal ATM switch fabric. A PIF checks if a received frame or packet is received for a protocol for which the interface is configured. It then searches the content-addressable memory (CAM) table and determines the egress VPI/VCI value for the destination port. Finally, the PIF segments the frame into cells, applies a five-byte header with the appropriate egress VPI/VCI information, and sends the cells out the switch fabric. If the PIF needs to drop a frame, it tags all cells in that frame with a VPI/VCI =0/0, and the switch fabric then drops these cells.

An Ethernet PIF drops cells and increments the invalid cells counter because of one of these reasons:

- Layer-2 filter of MAC addresses reachable out the same interface as the received Ethernet frame. The Catalyst 8500 filters such "local" MAC addresses when it sends received frames into the switch fabric on the 0/0 VPI/VCI. These drops are equivalent to the in-discards counter on the Catalyst 5000.
- Packet discard on a multiplex-based (MUX-based) Gigabit Ethernet line card. On such cards, when a received packet has a non-forwardable or unrecognizable protocol and must be discarded, the switch puts the packet on the 0/0 VPI/VCI.
- Microcode put a packet on a VPI/VCI that is not actually established in the switch core.

NetPro Discussion Forums – Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums – Featured Conversations for Router and IOS Architecture
Network Infrastructure: LAN Routing and Switching
Network Infrastructure: WAN Routing and Switching

Related Information

- **Technical Support & Documentation – Cisco Systems**
-

All contents are Copyright © 1992–2006 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: Dec 20, 2006

Document ID: 10454
