pathSolutions

WHITE PAPER

Error Counters Overview

Network Truth Lies in the Details

(877) 748-1777
Sales@PathSolutions.com
www.PathSolutions.com

Contents

(877) 748-1777
Sales@PathSolutions.com
www.PathSolutions.com

What Are Error Counters?

Switches and routers contain hundreds if not thousands, of variables that contain information relating to network operation, efficiency, and configuration. Every packet that gets processed, delayed, or dropped is tracked with sufficient granularity to determine if a problem was encountered by the device and interface. This information is kept inside the device as a counter. As a result, storing the information does not require a lot of resources, which enables the device to accomplish its primary task of processing packets at optimal speed.

However, there is one major drawback: there is no history of when these counters were incremented. For example, a counter might show that it processed 12,000 packets since the device was rebooted. If the device was rebooted six months ago, there is no way of determining if it was a steady stream of 12,000 packets over six months, or if the network was quiet until one month ago when it processed all 12,000 packets in a five second period and then went back to being quiet.

This is like having a tally counter track how many people entered a door. At the start of the day, the counter starts at 0 (zero). If nobody enters the door, it will remain zero for the entire day. If a manager makes rounds and checks the counter every five minutes, they can determine when this door was used, and by how many people.

On a network device, when the device is first booted up, its error counters all start at zero. As the device processes packets, it increments its counters to track how it dispositioned those packets. If a network troubleshooting solution checks the error counters every five minutes, it can determine when problems occurred, even though it is not looking specifically at individual packets.

(877) 748-1777
Sales@PathSolutions.com
www.PathSolutions.com

Where Does Error Counter information Come From?

Error counter information can come from two primary locations: The Ethernet chipset, and the device OS software.

The Ethernet chipsets used by device manufacturers have a number of counters that expose layer-1 and layer-2 problems. Some of the physical layer processing problems that the Ethernet chipsets support are: Alignment Errors, FCS Errors, Collisions, Internal MAC Transmit Errors, and Carrier Sense Errors. These errors can expose problems with cabling, the PHY chipset (https://en.wikipedia.org/wiki/PHY_(chip)), or the MII (Media Independent Interface: https://en.wikipedia.org/wiki/Media-independent_interface). These error counters typically involve some sort of hardware or configuration failure.

Other error counters come from the device OS software that tracks data going to and from the Ethernet chipset like: Inbound Unknown Protocols, Deferred Transmissions, Frame Too Longs, Outbound Queue Length, Inbound Discards, and Outbound Discards.

These error counters typically involve configuration failures, or performance limitations of the interface.

Why Are Error Counters Important?

Nearly two decades ago, network equipment manufacturers realized that operational costs could be reduced if additional information was available for troubleshooting through their devices. If their customers could see how the devices were performing, they could repair sub-optimal operating environments and fix a significant number of problems.

Manufacturers see this information as critically important to running smooth network operations. Otherwise they would not have gone to the expensive R&D effort to add this information to their devices.

What Are the Most Common Error Counters?

Some error counters are easier to understand than others. Here is a list of common error counters, with their official definition (as designated by the IEEE or IETF), and a basic description that provides some context as well as examples.

AlignmentErrors

Official definition: A count of frames received on a particular interface that are not an integral number of octets in length and do not pass the FCS check. The count represented by an instance of this object is incremented when the alignmentError status is returned by the MAC service to

the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC.

Basic definition: All frames on the segment should contain a number of bits that are divisible by eight (to create bytes). If a frame arrives on an interface that includes some spare bits left over, the interface does not know what to do with the spare bits. Example: If a received frame has 1605 bits, the receiving interface will count 200 bytes and will have 5 bits left over. The Ethernet interface doesn't know what to do with the remaining bits. It will discard the bits and increment the Alignment Error count. Because of these remaining bits, it is more likely that the CRC check will fail causing FCS Errors counter to also increment.

FCSErrors

Official definition: A count of frames received on a particular interface that are an integral number of octets in length but do not pass the FCS (Frame Check Sequence) check. The count represented by an instance of this object is incremented when the frameCheckError status is returned by the MAC service to the LLC (or other MAC user). Received frames for which multiple error conditions obtain are, according to the conventions of IEEE 802.3 Layer Management, counted exclusively according to the error status presented to the LLC.

Basic definition: An FCS error is a legal sized frame with a bad frame check sequence (CRC error). An FCS error can be caused by a duplex mismatch, faulty NIC or driver, cabling, hub, or induced noise.

IfInUnknownProtos

Official definition: The number of packets received via the interface which were discarded because of an unknown or unsupported protocol.

Basic definition: If the physical and data-link layer do their job successfully and deliver a frame to the correct MAC address, it is assumed that the requested protocol will be available on the machine. If the protocol is not available, the frame is discarded. If your machine receives an AppleTalk packet, but your machine isn't running AppleTalk, it will discard the packet and increment this counter.

SingleCollisionFrames

Official definition: A count of successfully transmitted frames on a particular interface for which transmission is inhibited by exactly one collision. A frame that is counted by an instance of this

object is also counted by the corresponding instance of either the ifOutUcastPkts or ifOutNUcastPkts object and is not counted by the corresponding instance of the dot3StatsMultipleCollisionFrames object.

Basic definition: If a network interface attempts to transmit a frame, and detects a collision, it will attempt to re-transmit the frame after the collision. If the retransmission was successful, then the event is logged as a single collision frame.

MultipleCollisionFrames

Official definition: A count of successfully transmitted frames on a particular interface for which transmission is inhibited by more than one collision. A frame that is counted by an instance of this object is also counted by the corresponding instance of either the ifOutUcastPkts or ifOutNUcastPkts object and is not counted by the corresponding instance of the dot3StatsSingleCollisionFrames object.

Basic definition: If a network interface attempts to transmit a frame, and detects a collision, it will attempt to re-transmit the frame after the collision. If the retransmission also causes a collision, then Multiple Collision Frames is incremented.

ExcessiveCollisions

Official definition: A count of frames for which transmission on a particular interface fails due to excessive collisions.

Basic definition: If there are too many collisions (beyond Multiple Collision Frames), the transmission will fail.

LateCollisions

Official definition: The number of times that a collision is detected on a particular interface later than 512 bit-times (64 bytes) into the transmission of a packet. Five hundred and twelve bit-times corresponds to 51.2 microseconds on a 10 Mbit/s system. A (late) collision included in a count represented by an instance of this object is also considered as a (generic) collision for purposes of other collision-related statistics.

Basic definition: Collisions should be detected within the first 64 bytes of a transmission. If an interface transmits a frame and detects a collision before sending out the first 64 bytes, it declares it to be a "normal collision" and increments Single Collision Frames (or Multiple Collision Frames if more collisions follow). If an interface transmits a frame and detects a

collision after sending out the first 64 bytes, it declares it to be a Late Collision. If a machine detects a Late Collision, it will treat the collision like any other collision (send a jam signal, and wait a random amount of time before attempting to retransmit). The other sending machine may or may NOT have detected the collision because it was so late in the transmission. The other sending machine may detect the collision AFTER it is done sending its frame, and will believe that its frame was sent out successfully.

FrameTooLongs

Official definition: If a frame is detected on an interface that is too long (as defined by ifMTU), this counter will increment.

Basic definition: Frame Too Longs occur when an interface receives a frame that is longer (in bytes) than the maximum transmission unit (MTU) of the interface.

MacReceiveErrors

Official definition: A count of frames for which transmission on a particular interface fails due to an internal MAC sublayer transmit error. A frame is only counted by an instance of this object if it is not counted by the corresponding instance of either the dot3StatsLateCollisions object, the dot3StatsExcessiveCollisions object, or the dot3StatsCarrierSenseErrors object. The precise meaning of the count represented by an instance of this object is implementation- specific. In particular, an instance of this object may represent a count of transmission errors on a particular interface that are not otherwise counted.

Basic definition: This is the number of frames that could not be transmitted due to an unknown problem. This unknown problem is not related to collisions or carrier sense errors. The device manufacturer's documentation may provide additional information on locating the source of these errors.

InternalMacTransmitErrors

Official definition: A count of frames for which transmission on a particular interface fails due to an internal MAC sublayer transmit error. A frame is only counted by an instance of this object if it is not counted by the corresponding instance of the dot3StatsLateCollisions object, the dot3StatsExcessiveCollisions object, or the dot3StatsCarrierSenseErrors object. The precise meaning of the count represented by an instance of this object is implementation-specific. In particular, an instance of this object may represent a count of transmission errors on a particular interface that are not otherwise counted.

Basic definition: If a transmission error occurs, but is not a late collision, excessive collision, or carrier sense error, it is counted as an error here. NIC vendors may identify these kinds of errors specifically. Check with the device's manufacturer to determine their interpretation of InternalMacTransmitErrors.

CarrierSenseErrors

Official definition: The number of times that the carrier sense condition was lost or never asserted when attempting to transmit a frame on a particular interface. The count represented by an instance of this object is incremented at most once per transmission attempt, even if the carrier sense condition fluctuates during a transmission attempt.

Basic definition: Carrier Sense Errors occur when an interface attempts to transmit a frame, but no carrier is detected, and the frame cannot be transmitted.

DeferredTransmissions

Official definition: A count of frames for which the first transmission attempt on a particular interface is delayed because the medium is busy. The count represented by an instance of this object does not include frames involved in collisions.

Basic definition: If an interface needs to transmit a frame, but the network is busy, it increments DeferredTransmissions. Transmissions that are deferred are buffered up and sent at a later time when the network is available again.

IfOutErrors

Official definition: The number of outbound packets that could not be transmitted because of errors.

Basic definition: These packets could not be transmitted due to one or more various data-link layer errors. The root cause of these errors are undefined. In order to more accurately research these types of errors, you should deploy a packet analyzer in front of this interface to track the specific errors that occur, as the device is not capable of tracking any additional information relating to these errors. If this interface provides Ethernet specific errors, these errors may be detailed in that section.

IfOutDiscards

Official definition: The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

Basic definition: If too many packets are queued to be transmitted, and the network interface is not fast enough to transmit all of the packets, it may be discarded.

IfInErrors

Official definition: The number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol.

Basic definition: These packets contained one or more various data-link layer errors, and were thus discarded before being passed to the network layer. The root cause of these errors are undefined. In order to more accurately research these types of errors, you should deploy a packet analyzer in front of this interface to track the specific errors that occur, as the device is not capable of tracking any additional information relating to these errors. If this interface provides Ethernet specific errors, these errors may be detailed in that section.

IfInDiscards

Official definition: The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.

Basic definition: If too many packets are received, and the protocol stack does not have enough resources to properly handle the packet, it may be discarded.

SQETestErrors

Official definition: A count of times that the SQE TEST ERROR message is generated by the PLS sublayer for a particular interface. The SQE TEST ERROR message is defined in section 7.2.2.2.4 of ANSI/IEEE 802.3-1985 and its generation is described in section 7.2.4.6 of the same document.

Basic definition: SQE stands for "Signal Quality Error", and may also be referred to the Ethernet "heartbeat". With early Ethernet cards that required transceivers, the transceiver would send a "Signal Quality Error" back to the Ethernet card after each frame was transmitted to insure that

the collision detection circuitry was working. With modern network cards, this SQE test can cause network cards to believe that an actual collision occurred, and a collision is sent out on the network when a SQE test is detected. This can seriously degrade network performance, as each frame successfully transmitted on the network is followed by a collision caused by the SQE test.

ifOutQLen

The length of the output packet queue (in packets). This number should return to zero in a short amount of time. If it ends up being any non-zero value for any length of time, you should consider upgrading the interface to a faster technology, or full duplex (if not already enabled).

What Do Error Counters Tell You?

While each error counter's definition can be used to diagnose problems, the *combination* of error counters can provide far more visibility when troubleshooting network faults.

Example 1:

If an interface has a number of Deferred Transmissions, there wasn't enough bandwidth to transmit the packets that were supposed to be transmitted, and as a result, they had to be placed in a temporary buffer. If you also have Outbound Discards, you ran out of buffers and the packets had to be discarded. This tells you that the interface does not have enough bandwidth to handle the traffic – even if your 5 minute utilization rate is low.

Example 2:

If you have FCS Errors, and have no Alignment Errors or Collisions, then you most likely have a cabling problem. You may also see Carrier Sense Errors at the same time.

But if you have FCS Errors with Alignment Errors, but no Collisions, you are on the full-duplex side of a duplex-mismatch.

If you have FCS Errors with Alignment Errors and Collisions, you are on the half-duplex side of a duplex-mismatch.

Example 3:

If you have Outbound Queue Length, Deferred Transmissions, and Outbound Discards, there is a serious bandwidth limitation occurring, as packets are currently still lined up in queue to be emptied.

As these examples illustrate, there are many more possibilities that can occur via combinations of errors. To completely understand the network's health and performance, you must also consider the combination of errors.

(877) 748-1777
Sales@PathSolutions.com
www.PathSolutions.com

Total Network Visibility

All of this data existing on network equipment requires educated interpretation. This creates a problem for network operators: How do you make this information available, and have it automatically interpreted to produce meaningful actionable result?

PathSolutions TotalView automatically interrogates switches, routers, firewalls, gateways, and other network devices for error counter, performance, and configuration information. It does this for every device manufacturer: if the manufacturer makes this information available via standard SNMP, it is collected.

The method of collection is important as there is a significant amount of information. The information needs to be collected in an efficient manner so that network links are not saturated with collection activities, and CPU usage on the network devices is kept at a minimum.

TotalView collects multiple data elements with a *single* SNMP fetch—sending and receiving far fewer packets than other collection engines. It also uses a single thread for collection. This ensures that a network device doesn't deal with a flood of management packets causing CPU spikes during collection. Instead, it's a slow and steady pace of collection on devices.

Once the data is collected, it is put through TotalView's Network Prescription™ engine that analyzes the error counters with a heuristics algorithm and produces a plain-English resolution to the problem. This speeds understanding of the root-cause of the problem.

Additional Resource:

Webinar: Finding Packet Loss in the Network

(877) 748-1777
Sales@PathSolutions.com
www.PathSolutions.com