Title: Troubleshooting Ethernet/Fast Ethernet and Fragmentation Issues on a NetScreen Device

Document Number:
Version: January 2002
OS Ver. this Paper Applies to: 2.6.x and above
HW Platforms this Paper Applies to: All
Audience (Internal or External): External

Purpose

The purpose of this document is to provide the reader with knowledge to identify and troubleshoot Ethernet/Fast Ethernet and fragmentation issues, which can affect performance on a NetScreen device.

Assumptions

This paper was written with the assumption that the reader has prior experience with configuring and troubleshooting ScreenOS. The reader must also have previous experience with Ethernet and Fast Ethernet in twisted-pair environments. The reader should also have a fundamental knowledge of routing and the Internet Protocol (IP). All references to Ethernet within this paper apply to networks with twisted-pair cabling only.

Introduction

The complexity of networking increases everyday. Even the most basic network has many complex components. Troubleshooting performance problems can be a difficult task but if a practical approach is taken, problems can be identified quickly. ScreenOS provides a great number of tools that allow network administrators to identify and fix performance problems quickly and easily.

Interfaces

NetScreen devices support different interface speeds. Below is a partial list of models and corresponding speeds supported (Ethernet and Fast Ethernet only).

Model	Speed(s)
NS5/5XP	10
NS10	10
NS100	10/100
NS500	10/100

If you are experiencing connectivity or performance issues, it is important to check all physical connections. Check link lights and cabling first then check software configurations. If the NetScreen device is connected to interface ports that are manageable, check counter statistics for errors. Always be sure to check both ends of an Ethernet link because some link errors will only show up on one end of a link. When analyzing interface counters, it is important to get a "snap shot" of the current interface counter numbers first. Proceed by clearing the counters and then monitor for any increases. More information on NetScreen interface counters is covered in the following sections of this paper.

Interface Errors

Listed below are descriptions for some common Ethernet errors found in twisted-pair networks. It is important to note that in half-duplex environments, some data link errors such as alignment errors, CRC errors and collisions are normal.

Duplex Mismatch – One of the most common causes of performance issues on twisted-pair, Ethernet networks. A duplex mismatch is when one port on an Ethernet link is operating at half-duplex while the other port is operating at full-duplex. The result of a duplex mismatch can be extremely slow performance, intermittent connectivity problems and/or a complete loss of connectivity. A duplex mismatch occasionally happens when one or both ports on a link are reset and auto-negotiation doesn't function properly. Another cause could be changing the duplex on either end of a link but forgetting to force the link down so both ends will renegotiate with the new duplex setting. If you are using auto-negotiation, both sides of a link should have it on or off. It is also important to note that some 10MB implementations only support half-duplex.

Collision – On an Ethernet network it is possible for two devices to sense the wire and transmit at exactly the same time, resulting in a collision. After detecting a collision, the device waits a random delay and then attempts to re-transmit the packet. If the device detects a collision again, it waits twice as long to re-transmit the message, this is known as exponential back off. Excessive collisions can affect general performance and will result in traffic shaping allocating bandwidth improperly on a NetScreen device. Collisions can also cause alignment errors due to the frame not being completely copied to the wire, which may result in fragmented frames. Collisions should be minimal in full-duplex environments.

Alignment Errors – Alignment errors are typically the result of a duplex mismatch. They occur when a frame does not end with an even number of octets and has a bad CRC. If there is a duplex mismatch, you may see rapidly increasing alignment errors but the error may or may not show up on the interface counters, depending on which end of the connection is configured for half-duplex.

Interface Counters

Interface counters should be checked periodically to ensure there are no errors on your links. You can check all counters on a NetScreen device by issuing the command '*get counter stat*'. Below is a list of some important interface counters to watch for on ScreenOS 2.6.x and above:

crc err	Number of packets with a cyclic redundancy check error
align err	Number of packets with alignment error in the bit stream
no buffer	Number of packets dropped due to unavailable buffers
misc err	Number of packets with at least one error
coll err	Number of collision packets

Interface Settings

By default, all NetScreen devices auto-negotiate to determine duplex and speed. Sometimes auto-negotiation will not function properly and you may need to force an interface to a specific setting. The following physical parameters can be set on a NetScreen device:

'set interface **INTERFACE** phy **DUPLEX SPEED'**

INTERFACE	trust | untrust | dmz | mgt
DUPLEX	auto | half | full
SPEED	10MB | 100MB *not available on NS5 or NS10

Always be sure to force a link down after any changes are made to the physical interface settings. This can be done by unplugging the Ethernet connection.

Fragmentation

Another issue that can affect performance is fragmentation. Fragmentation happens when a packet is too large to be sent across a link. When this happens, the original packet is split into smaller packets, each containing enough information to allow the recipient to reassemble the fragments back to their original state. Fragmentation typically happens on a device that supports different media types, such as a router. Once a packet is fragmented, it will not be reassembled until it reaches its destination. Fragmentation is undesirable for numerous reasons, including:

- If any one fragment from a packet is dropped, the entire packet is retransmitted.

- Fragments impose processing load on the routers that have to split the packets.

- Fragments routers/firewalls may block fragments because they do not contain the header information from a higher layer protocol (i.e. TCP). Some devices require this information for filtering.

Maximum Transmission Unit (MTU)

To avoid fragmentation, it is sometimes necessary to modify the maximum transmission unit (MTU) for a packet. MTU is the maximum size packet or frame (specified in bytes) that can be sent in a packet or frame-based network. TCP uses MTU to determine the size of each packet in any transmission. Too large an MTU size may result in retransmissions if the packet encounters a router/network that cannot handle that large of a packet. An MTU size that is too small can result in additional packets being generated. This behavior can cause performance issues because of the added header overhead and the extra CPU cycles required to process the additional packets.

The standard MTU size for Ethernet is 1500 bytes. The de facto standard for the Internet is 576 bytes, however, most ISPs are now using an MTU size of 1500 bytes. In general, Internet users should follow the advice of their Internet Service Provider (ISP) about whether to change the default MTU value and what to change it to.

Below is a list of media types and their default MTU size listed in bytes:

Media Type	MTU
16 Mbit/Sec Token Ring	17914
4 Mbits/Sec Token Ring	4464
FDDI	4352
Ethernet	1500
IEEE 802.3/802.2	1492
PPP (typical; can vary)	1500
PPPoE	1492
X.25	576

Maximum Segment Size

In TCP communications Maximum Segment Size (MSS) is used to announce to another end-station the largest amount of data (in bytes) that should be sent in a single packet. The MSS setting is found in the "option" field of a TCP packet and should only be sent in the initial connection request (i.e., in segments with the SYN control bit set). Setting the MSS is optional and if this option is not used, any segment size is allowed. MSS can be used independently in each direction of data flow, which can result in different segment sizes in both directions. Most systems announce a MSS that is determined from the MTU on the local transmitting interface.

Path MTU Discovery

The Path MTU Discovery system (PMTU) was established to determine the smallest allowable MTU of any link on the current path between two hosts. The MTU size for the path between two hosts can vary since the routing may change over time. The path is not necessarily symmetric and can even vary for different types of traffic from the same host.

A host performs Path MTU Discovery by sending out as large a packet as possible. This packet is sent with the Don't Fragment (DF) bit set in the IP header. If a device is configured to participate in PMTU, when it receives a packet that is too large to forward on to the next link and the DF bit is set, the device will send an ICMP "Destination Unreachable - Fragmentation Needed" message to the source address. If the ICMP message makes it back to the host, it will adjusts the packet size. Unfortunately, devices on the Internet do not always forward these ICMP messages correctly for a variety of reasons which results in PMTU not working properly.

Typical Fragmentation Scenarios

On a NetScreen device, there are two common cases where fragmentation can occur:

IPsec Tunnel - IPsec requires the addition of an ESP and/or AH header to tunnel a packet. An ESP header is at least 36 bytes so if the original datagram is greater than 1464 bytes, the added ESP header will cause the original packet to be fragmented on an Ethernet network. Authentication headers can be even longer so fragmentation can also be expected with this protocol. Fragmentation through an IPsec tunnel can be corrected in ScreenOS by setting the Maximum Segment Size (MSS) size for all sessions in an IPsec tunnel ('*set flow tcp-mss*'). More information on adjusting the MSS option in a TCP packet is listed in the following section of this paper.

PPPoE Connection - PPPoE connections are PPP connections encapsulated in an Ethernet frame. PPPoE requires the addition of a PPP header (6 bytes) and a protocol ID field (2 bytes) which totals 8 bytes. If the original packet to be sent over the PPPoE connection is greater than 1492 bytes, the added PPP header will cause the original packet to be fragmented. This can be corrected in ScreenOS by configuring the Maximum Segment Size (MSS) for all traffic through the NetScreen device (*'set flow all-tcp-mss'*). More information on adjusting MSS option in a TCP packet is listed in the following section of this paper.

ScreenOS MTU and MSS Settings

The following settings can be used to combat fragmentation issues on a NetScreen device:

'set flow tcp-mss xxxx'

This setting applies to traffic through IPsec tunnels only. Configuring this command will cause the NetScreen device to interfere with the initial handshaking between two end-stations by rewriting the MSS option field in a TCP packet to be the value 'xxxx'. For example, if the setting 'set flow tcp-mss' is configured (the default size is 1400 bytes), the two end-stations will appear to be announcing an MSS size of 1400 bytes, which should be safe for normal IPsec operation. NetScreen recommends a setting of 1400 bytes to allow for IPsec headers but this setting can vary depending on the environment. Generally, this setting should be configured for all NetScreen devices with IPsec tunnels configured. If this setting is configured, we will only rewrite the MSS option if the proposed size is larger than what is configured on the NetScreen device. No modification is made if the original MSS is smaller.

'set flow all-tcp-mss xxxx'

This command is similar to 'set flow tcp-mss' except this command applies to ALL traffic through the NetScreen device. Setting this command will cause the NetScreen device to rewrite the MSS option (when present) to be whatever value 'xxxx' is. This command is typically used to correct fragmentation issues with PPPoE connections to a NetScreen device. If this setting is configured, we will only rewrite the MSS option if the proposed size is larger than what is configured on the NetScreen device. No modification is made if the original MSS is smaller.

'set flow path-mtu'

This command will allow a NetScreen device to participate in the Path MTU discovery process. This means that when the NetScreen device receives a packet that must be fragmented and the Don't Fragment (DF) bit is set, it will discard the packet and send an ICMP packet (Destination Unreachable - Fragmentation Needed) suggesting a smaller packet size.

Appendix

The following entries are taken from various sources, mostly RFCs. This is by no means a complete list of information. Instead, listed are some common components found in IP networking.

1.1 Assigned Internet Protocol Numbers

The Internet Protocol (IP) contains a field called Protocol to identify the next level protocol. Below Is a list of common IP protocols:

Decimal	Protocol	Description
1	ICMP	Internet Control Message
6	TCP	Transmission Control
17	UDP	User Datagram
47	GRE	General Routing Encapsulation
50	ESP	Encapsulation Security Payload
51	AH	Authentication Header

1.2 Well Known Port Numbers

Within TCP and UDP are defined service ports. Below is a list of common ports:

Service	Port	Description
FTP-DATA	20	File Transfer [Default Data]
FTP	21	File Transfer [Control]
TELNET	23	Telnet
SMTP	25	Simple Mail Transfer
DNS	53	Domain Name Server
TFTP	69	Trivial File Transfer
HTTP	80	HTTP
POP3	110	Post Office Protocol - Version 3
NNTP	119	Network News Transfer Protocol
NTP	123	Network Time Protocol
NETBIOS-NS	137	NETBIOS Name Service
NETBIOS-DGM	138	NETBIOS Datagram Service
NETBIOS-SSN	139	NETBIOS Session Service
SNMP	161	SNMP
SNMPTRAP	162	SNMPTRAP
IRC	194	Internet Relay Chat Protocol
IMAP3	220	Interactive Mail Access Protocol v3
LDAP	389	Lightweight Directory Access Protocol
HTTPS	443	HTTPS

2.1 Common ICMP Message "Type" Numbers

ICMP does not use port number. Instead, "type" fields are used. Below is a list of common ICMP messages:

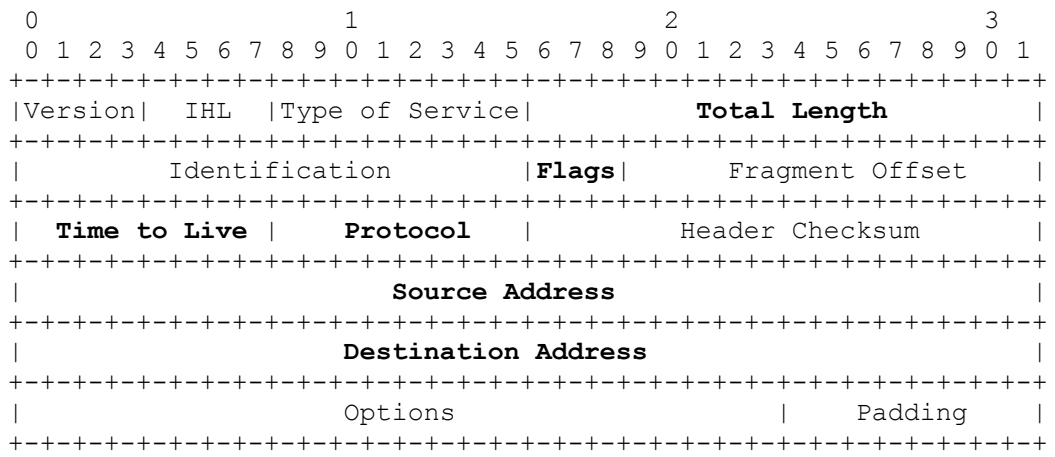
Type	Name
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo
30	Traceroute

2.2 ICMP "Code" Field – Destination Unreachable

Many of these ICMP types have a "code" field. Below are further details of the code field for ICMP Message Type 3, Destination Unreachable:

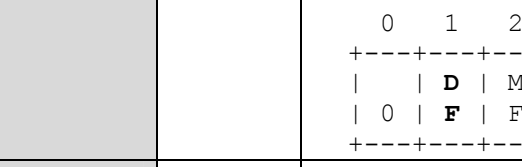
Code	Description
0	Net Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation Needed and Don't Fragment was Set
5	Source Route Failed
6	Destination Network Unknown
7	Destination Host Unknown
8	Source Host Isolated
9	Communication with Destination Network is Administratively Prohibited
10	Communication with Destination Host is Administratively Prohibited
11	Destination Network Unreachable for Type of Service
12	Destination Host Unreachable for Type of Service
13	Communication Administratively Prohibited
14	Host Precedence Violation
15	Precedence cutoff in effect

3.1 IP Header

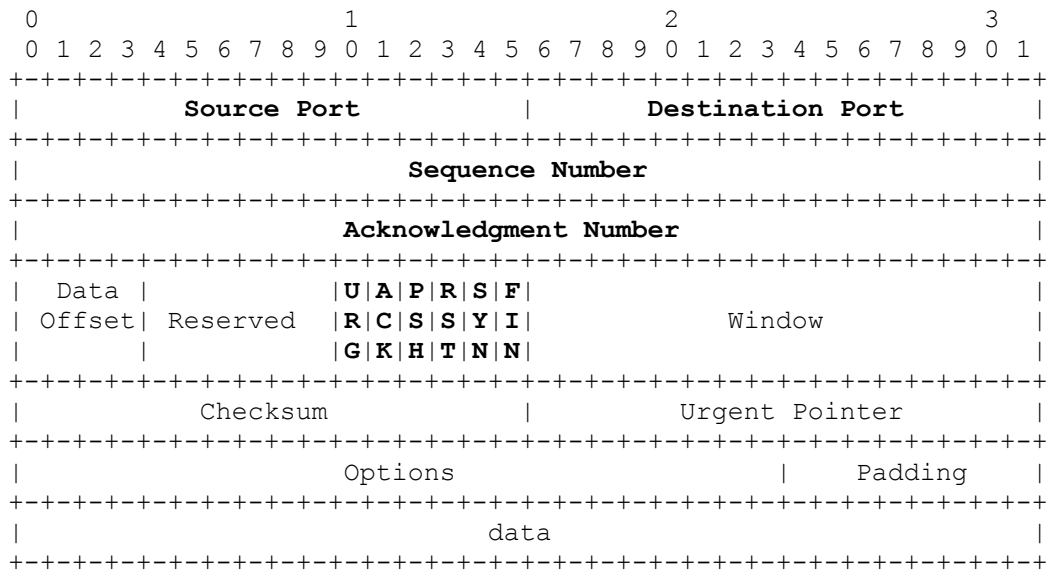


**Note that one tick mark represents one bit position

Notable Fields:

Total Length	16 bits	Total Length is the length of the datagram, measured in octets, including internet header and data. This field allows the length of a datagram to be up to 65,535 octets. Such long datagrams are impractical for most hosts and networks. All hosts must be prepared to accept datagrams of up to 576 octets (whether they arrive whole or in fragments).
Flags	3 bits	If bit number 1 (DF bit) is set to "1" then the packet cannot be fragmented: 
Time to Live	8 bits	This field indicates the maximum time the datagram is allowed to remain on the Internet. If this field contains the value zero, then the datagram must be discarded. The time is measured in units of seconds. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.
Protocol	8 bits	This field indicates the next level protocol used in the data portion of the internet datagram.
Source Address	32 bits	The source address.
Destination Address	32 bits	The destination address.

4.1 TCP Header



**Note that one tick mark represents one bit position

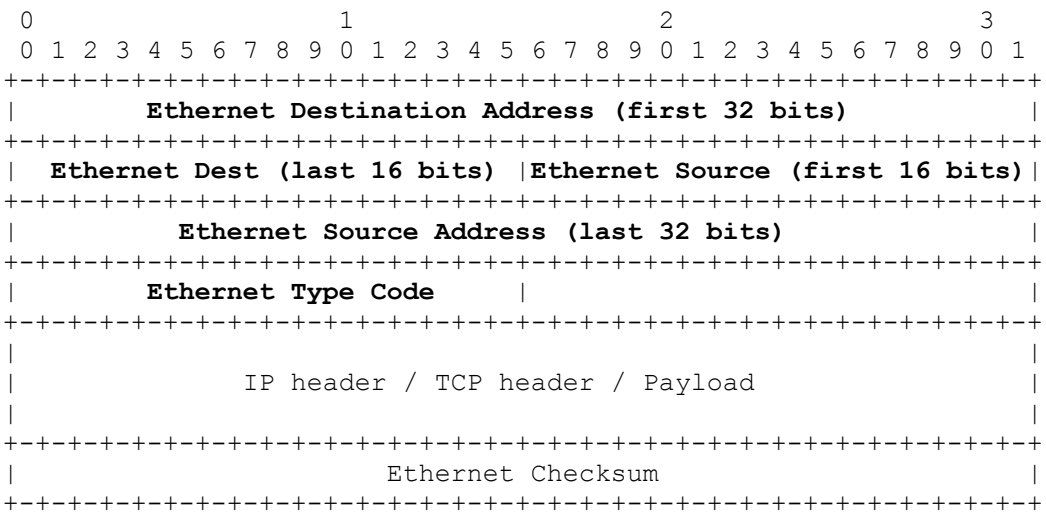
Notable Fields:

Source Port	16 bits	The source port number.
Destination Port	16 bits	The destination port number.
Sequence Number	32 bits	The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.
Acknowledgment Number	32 bits	If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

Control Bits (6 bits):

URG	Urgent Pointer field significant
ACK	Acknowledgment field significant
PSH	Push Function
RST	Reset the connection
SYN	Synchronize sequence numbers
FIN	No more data from sender

5.1 Ethernet Packet



**Note that one tick mark represents one bit position

Ethernet Hardware Address

Ethernet addresses are 48 bits, expressed as 12 hexadecimal digits. The first 6 digits are vendor specific and are assigned by IEEE. The first 6 digits are referred to as the Organizationally Unique Identifier (OUI) or 'company_id'. Listed below are some sample IEEE OUIs:

OUI	Vendor
0010DB	NetScreen Technologies
000130	Extreme Networks
000480	Foundry Networks

The last 6 digits of the Ethernet hardware address specifies the interface serial number specific to that interface vendor.

Ethernet Type Code

The 13th and 14th octets of an Ethernet packet (after the preamble) consist of the "Ethernet Type" field. This field is used to identify the protocol contained within the packet. Listed below are sample Ethernet Type codes:

Ethernet Type	Protocol
0800	Internet Protocol (IP)
0806	Address Resolution Protocol (ARP)
8035	Reverse Address Resolution Protocol (RARP)
8037	IPX (Novell Netware)
809B	EtherTalk (AppleTalk over Ethernet)
86DD	IP version 6

6.1 Windows Counters

From a command prompt in Windows, you can check fragmentation and other errors by using the netstat -s (use netstat -s | more to scroll by page).

```
C:\>netstat -s

  Packets Received                   = 182028
  Received Header Errors             = 0
  Received Address Errors            = 0
  Datagrams Forwarded                = 0
  Unknown Protocols Received         = 0
  Received Packets Discarded         = 0
  Received Packets Delivered         = 182028
  Output Requests                    = 183699
  Routing Discards                   = 0
  Discarded Output Packets           = 0
  Output Packet No Route             = 2
  Reassembly Required                = 0
  Reassembly Successful              = 0
  Reassembly Failures                = 0
  Datagrams Successfully Fragmented  = 0
  Datagrams Failing Fragmentation    = 0
  Fragments Created                  = 0

ICMP Statistics
                           Received    Sent
  Messages                 10          38
  Errors                   0           0
  Destination Unreachable  0           19
  Time Exceeded            0           0
  Parameter Problems       0           0
  Source Quenches          0           0
  Redirects                0           0
  Echos                    2           17
  Echo Replies             8           2
  Timestamps               0           0
  Timestamp Replies        0           0
  Address Masks            0           0
  Address Mask Replies     0           0

TCP Statistics

  Active Opens                        = 3118
  Passive Opens                       = 1
  Failed Connection Attempts          = 55
  Reset Connections                   = 582
  Current Connections                 = 5
  Segments Received                   = 161196
  Segments Sent                       = 161840
  Segments Retransmitted              = 651

UDP Statistics

  Datagrams Received    = 20748
  No Ports              = 81
  Receive Errors        = 1
  Datagrams Sent        = 21175
```

Appendix

The following sources were used for reference. Please refer to any of these sources for more detail.

RFC 791 Postel, J., "**Internet Protocol**", RFC 791, September 1981.

RFC 793 Postel, J., "**Transmission Control Protocol**", RFC 793, September 1981.

RFC 879 Postel, J., "**TCP Maximum Segment Size**", RFC 879, November 1983.

RFC 1191 McCann, J., Mogul, J. and S. Deering, "**Path MTU Discovery**", RFC 1191, November 1990.

RFC 1700 Reynolds, J. and J. Postel, "**Assigned Numbers**", RFC 1700, October 1994.

RFC 2402 Kent, S. and R. Atkinson, "**IP Authentication Header**", RFC 2402, November 1998.

RFC 2406 Kent, S. and R. Atkinson, "**IP Encapsulating Security Protocol (ESP)**", RFC 2406, November 1998.

RFC 2923 Lahey, K., " **TCP Problems with Path MTU Discovery**", RFC 2923, September 2000.