IBM Software Group

TPF Users Group Spring 2005

TCP/IP Support Enhancements

Mark Gambino

AIM Enterprise Platform Software

IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Flood Insurance

IP Fragmentation Processing

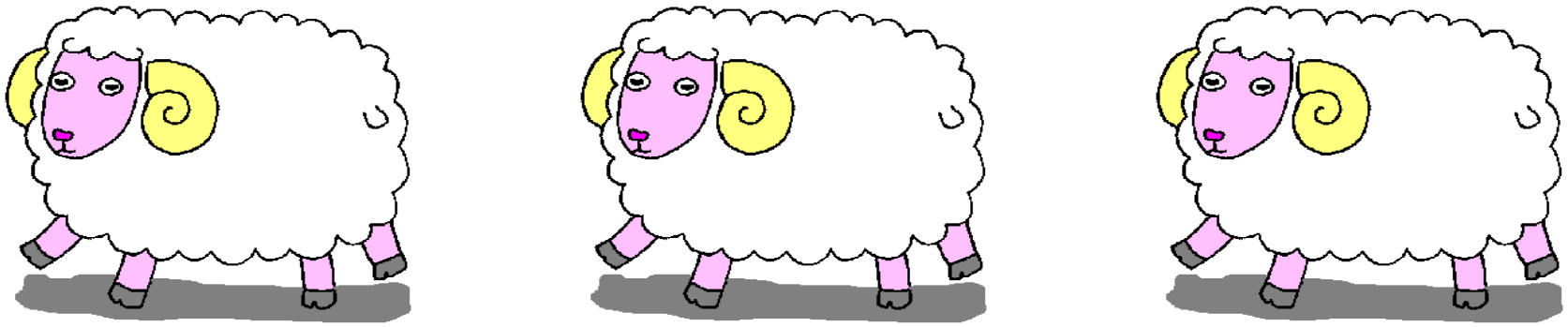
- When a message is too large to flow across a link in the network, the message is broken up into smaller packets called *IP fragments.*
- The IP layer in the receiver node of the socket reassembles the fragments, then passes the message to the protocol (TCP or UDP) layer.
- When the first fragment arrives, the IP reassembly timer is started. If the timer expires before all fragments of this message arrive, the fragments received so far are discarded.

IP Fragmentation Attack

- A denial or service (DoS) attack exists where many fragments (partial messages only) are sent to a node in a short amount of time, causing buffer depletion in the receiver node.
- Variations exist where only middle of message fragments are sent because the destination socket cannot be identified until the first in sequence fragment is received.
  - Only the first in sequence fragment contains the protocol (TCP or UDP) header, which is needed to identify the socket.

IP Fragment Flood Denial of Service Attack Prevention

- When data arrives for a socket in TPF, the data is queued in the IP message table (IPMT) until the application reads the data
  - Socket receive buffer size limits the amount of IPMT storage that one socket can use
- When IP fragments arrive in TPF, they are queued in the IPMT until all fragments arrive for this message (or until the IP reassembly timer expires)
- To prevent IP fragments flood attacks from depleting the IPMT, APAR PJ29978 added new CTK2 parameter:
  - MAXFRAG - defines the percentage of IPMT storage that can be used to queue inbound IP fragments
  - You can dynamically update the MAXFRAG value using the ZNKEY command
  - New counter added for fragments discarded because TPF is at the MAXFRAG limit
  - Can use new IP trace FRAG option (added by APAR PJ30131) to identify the potential attacker(s)



Duplication

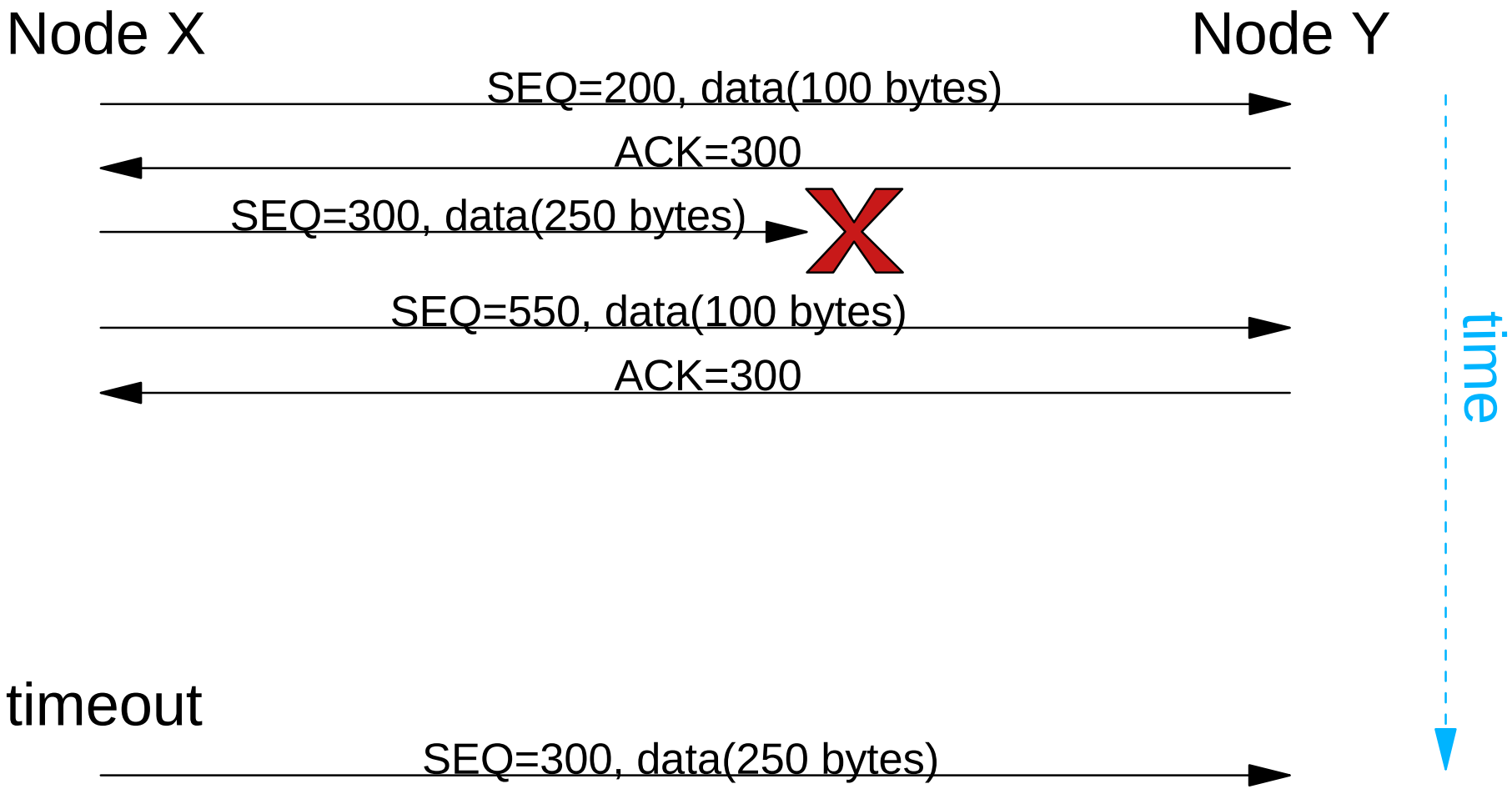
TCP Data Flow

- Each byte of data on a TCP socket has a sequence number
  - Used to ensure data is delivered in the correct order to the remote application
- Remote node acknowledges data received by sending an ACK value in its packets
  - The ACK can be piggybacked with data or can be a stand-alone ACK packet
- If no ACK is received, the data is assumed to have been lost in the network and the data is retransmitted
  - Normal TCP retransmit processing can impact throughput because the socket is idle for a period of time

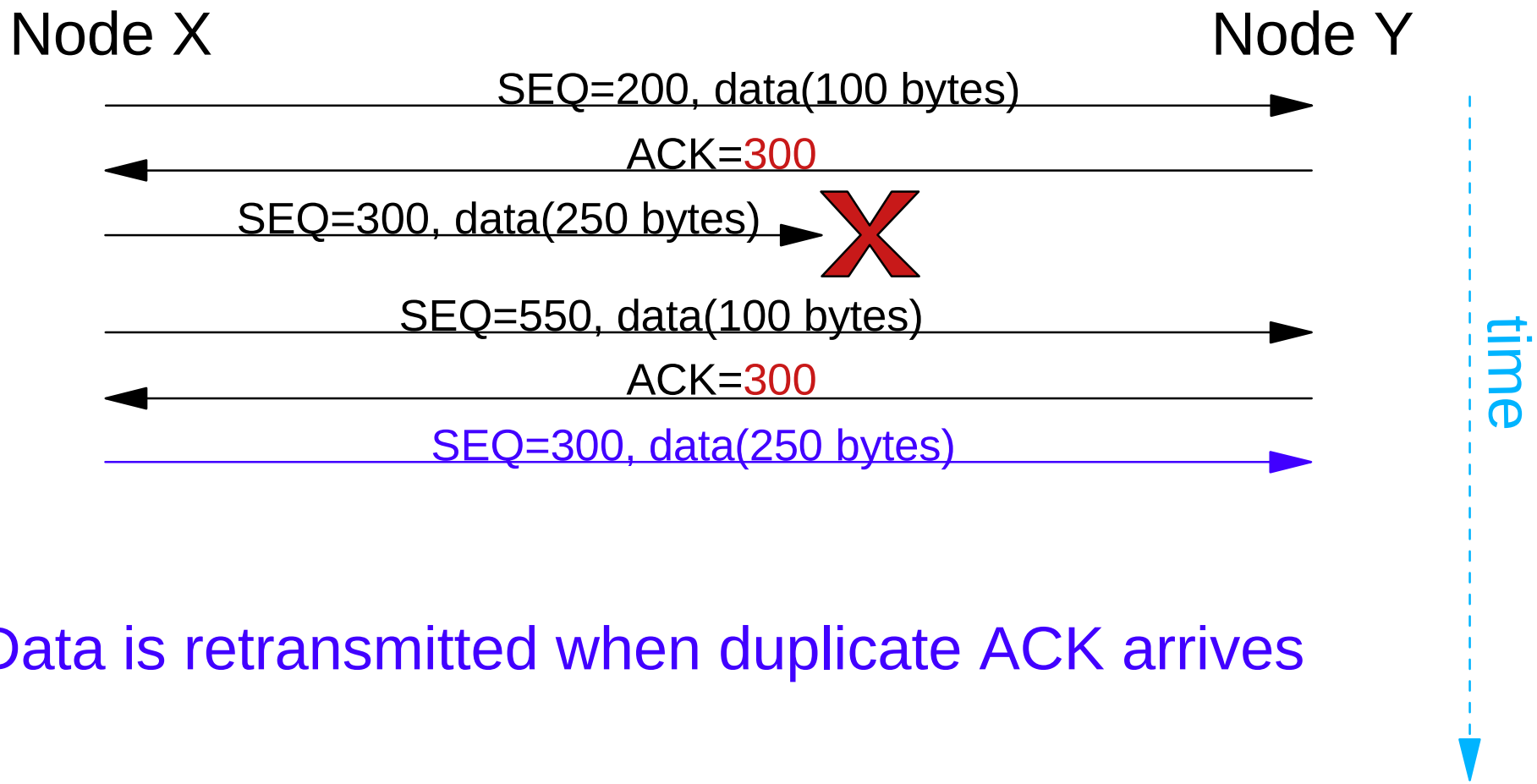
TCP Fast Retransmit Processing

- RFC 2001 introduced the concept of *TCP fast retransmit* processing
- Rather than waiting for a timeout, data is retransmitted if consecutive packets with duplicate ACK values are received
  - If data arrives out of order, a stand-alone ACK is immediately sent to indicate a possible packet loss in the network
  - Improves throughput on many sockets
- APAR PJ28344 (PUT 16) added TCP fast retransmit support to TPF

Normal Retransmit Processing



Fast Retransmit Processing

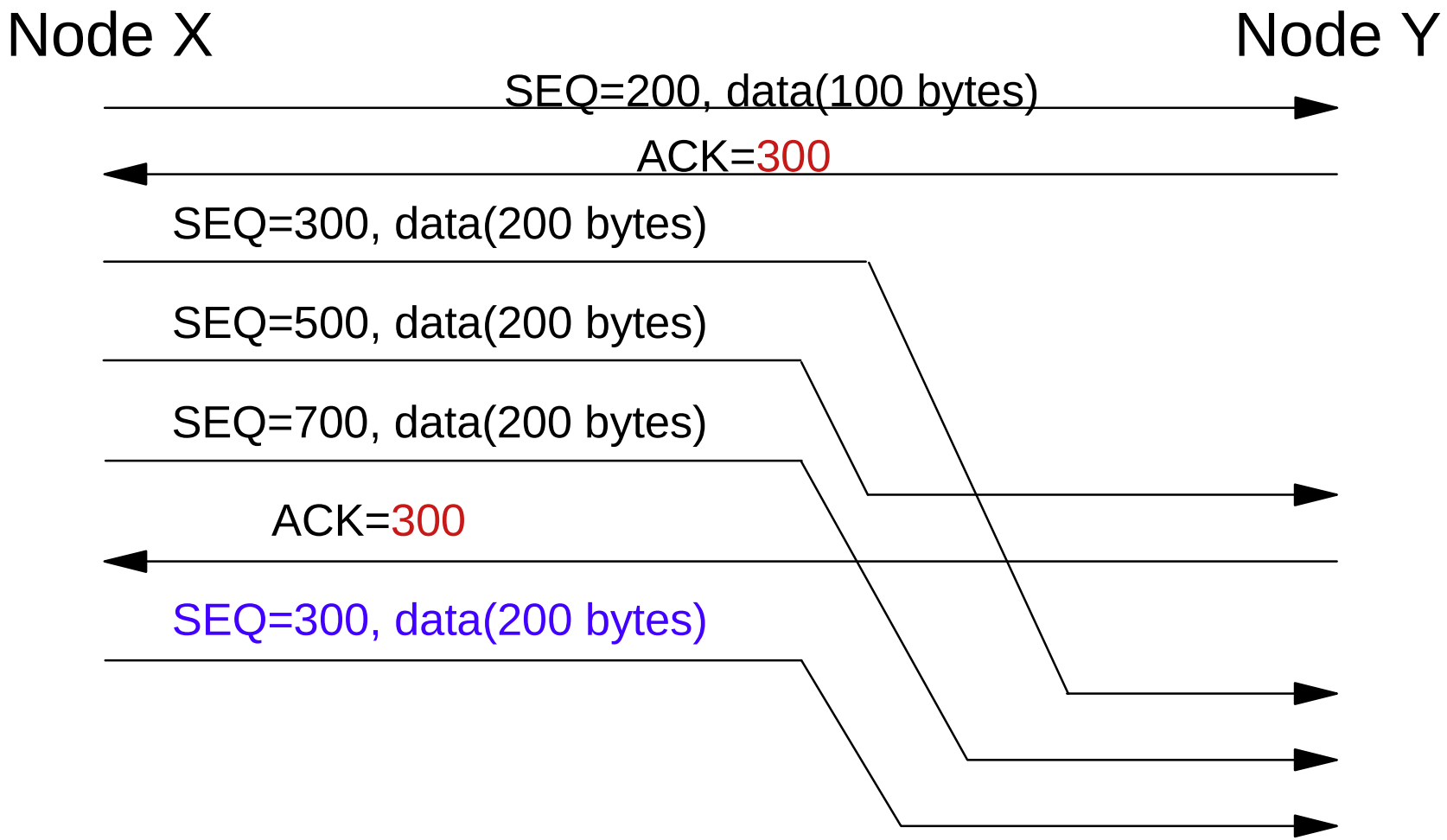


Data is retransmitted when duplicate ACK arrives

Network Routing Problems

- Non-optimal network routing can trigger fast retransmit processing even when no packets were lost
  - Can occur if the network is defined such that packets for the same socket flow across different routes
  - Packets arrive out of order at the remote end, making it look like packets were lost
  - Even if fast retransmit is not triggered, out of order data processing still causes extra overhead at the remote node

Network Routing Causing Unnecessary Fast Retransmit



Tuning TCP Fast Retransmit Support

- For sockets with non-optimal routing, you should increase the number of duplicate ACKs that must be received to trigger fast retransmit processing
- Note - increasing the number of duplicate ACK required reduces the likelihood of fast retransmit processing from being triggered in true error (packet loss) cases
- APAR PJ29978 adds the following options:
  - TCPDUACK - new CTK2 parameter that defines the number of consecutive duplicate ACKs that must be received on a TCP socket before data is fast retransmitted
  - SO_TCPDUACK option on *setsockopt* API - allows you to override the number of duplicate ACKs for a given socket
  - Number of duplicate ACKs range is 0 to 5
    - 0 means do not fast retransmit data on this socket

You've Got Options

Socket Options

- TCP server uses a listener socket to accept remote client connections
  - Client sockets inherit the properties of the listener socket
- TPF supports many standard and TPF-unique socket options that can be set using the *setsockopt* and *ioctl* APIs
- Sample socket options:
  - SO_RCVBUFF - the socket receive buffer size
  - SO_RCVTIMEO - timeout value for receive type APIs
  - TPF_AOR_BALANCE - AOR will create new ECBs on the least busy I-stream
  - SO_TCPDUACK - define the number of duplicate ACKs that must be received to trigger fast retransmit processing
- Many options are applicable to UDP servers as well

Setting Socket Options for INETD Servers

- APAR PJ30091 adds user exit USOC to allow you to set socket options for servers created by and controlled by the Internet Daemon (INETD)
- For TCP servers:
  - USOC is called after INETD creates the listener socket and before the first client socket is accepted
  - In USOC, you can issue *setsockopt* and *ioctl* APIs to set the options for this TCP server
  - Use the existing socket accept user exit (UACC) to override the options on a given client socket
- For UDP servers, USOC is called after INETD creates the socket and before the first message is read
- USOC is passed the name of the server application and the file descriptor (FD) of the server socket

Change of Address

Manual VIPA Move - Current Processing

- When a virtual IP address (VIPA) is moved from one processor to another in a loosely coupled complex, sockets are cleaned up internally within TPF
  - The remote node will find out that socket no longer exists when it sends its next packet to TPF
  - The remote node will reconnect and the new socket will be set up on the TPF processor that now owns the VIPA
- No notification (TCP RST) is set when the VIPA is moved
  - This prevents flooding the network with RSTs followed by a surge of TCP connection requests, both of which can effect traffic flowing on other sockets

Manual VIPA Move - Updated Processing

- Some customer applications are waiting for data from TPF and take a while to time out and recover
- Other customers have stateful firewalls that try to keep track of socket state information
- APAR PJ30102 adds new user exit UVMV to allow you to decide which sockets to send notification (TCP RST) to the remote node on when a VIPA is being moved
  - UVMV input includes the local and remote IP addresses and port numbers
  - Default logic is not to send a RST
- To prevent timing problems, the VIPA is not moved to the new processor until the requested RSTs have been sent on the processor from which the VIPA is being moved

Searching for Clues

IP Trace Options

- Offline IP trace (IPTPRT) facility creates a report of packets that meet a set of user specified input criteria
- APAR PJ30131 adds new input criteria options to improve your diagnostic capability:
  - DORIP - includes routing information protocol (RIP) packets. Default is now to not include RIP packets.
  - ZEROWIN - only include packets that have a 0 window value in the TCP header
  - FRAG - only include packets that are IP fragments
  - READ - only include input packets to TPF
  - WRITE - only include output packets from TPF
  - LIP - only include input and output packets that contain the specified local (TPF) IP address
  - RIP - only include input and output packets that contain the specified remote IP address

More New IP Trace Options

- More new input criteria:
  - LPORT - only include input and output packets that contain the specified local (TPF) port number
  - RPORT - only include input and output packets that contain the specified remote port number
  - RC option now allows RC=ALL, which indicates to only include packets that contain reason codes
- Each packet in the report now contains both the raw TOD clock value and a converted (human readable) TOD value:
  - For example, 16:34:20.56873
- New NODATA option causes only packet headers (IP header, TCP header, UDP header) to be displayed in the report

Does x'6E' mean > or n ?

Online IP Trace

- Many middleware packages send text data across the network in ASCII format
- Offline IP trace (IPRPRT) already has an option to display the data portion of IP packets in ASCII rather than EBCDIC
- APAR PJ30024 adds the option to display data in ASCII for the online IP traces, including:
  - System-wide IP trace display (ZIPTR)
  - Individual IP trace display (ZINIP)

Summary

- PJ29978
  - Prevents IP fragment flood denial of service (DoS) attacks
  - Allows you to tune TCP fast retransmit support
- PJ30091
  - Allows you to set up socket options for INETD servers
- PJ30102
  - Allows you to customize socket clean up during VIPA move processing
- PJ30131
  - Various enhancements to the offline IP trace facility
- PJ30024
  - Option to display online IP trace data in ASCII

Trademarks

IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Notes
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.