z/TPF Communications Enhancements

Name: Mark Gambino
Venue: Communications Subcommittee

AIM Enterprise Platform Software
IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

Overview

- This presentation covers recent z/TPF communications enhancements
- All APARs listed in this presentation will ship on PUT 4

I Hear You Knockin', but You Can't Come In

Starting an SNA APPN Session (Current Support)

- When an SNA remote LU is starting a session with z/TPF across an APPN network
  - The APPN Select a Host User Exit (UAPN) is called
- UAPN can choose:
  - The z/TPF host on which the LU-LU session will be established (if loosely-coupled)
  - The first hop link (ALS) over which to start the session
- UAPN can also tell the z/TPF system to select the z/TPF host and link to use

New APPN User Exit Option (APAR PJ31939)

- A new return code option has been added to UAPN indicating that z/TPF should reject the session request indicating the target (z/TPF) application is not available
- Can be used if you do not want this remote LU to log on at this time
- Can be used to reject certain session requests during peak periods or when the z/TPF complex is overloaded

The Boomerang Effect

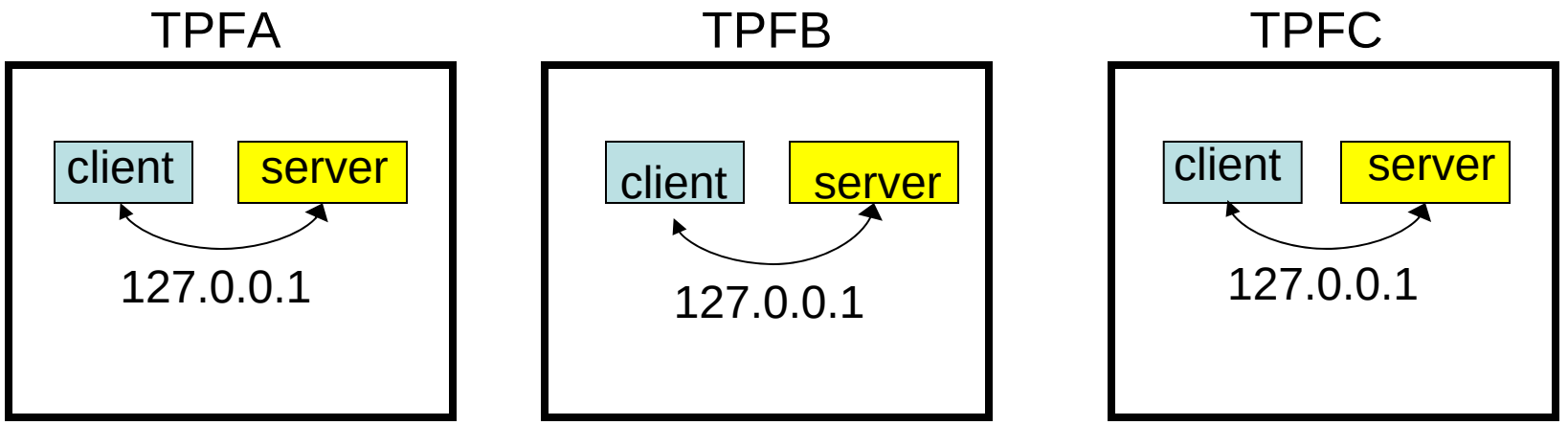
Local Sockets

- Client and server programs reside on the same host (same z/TPF processor)
- Client program issues *connect* (TCP) or *sendto* (UDP) specifying an IP address that is local to this z/TPF processor
- Outbound packet is built and placed on input list for processing
- No physical network interface (OSA-Express) is needed or used

Localhost Interface Support

- IP address 127.0.0.1 is defined by the architecture as the loopback address
- APAR PJ32025 adds support for the loopback address
- Operates the same as local sockets
  - No OSA-Express adapter is needed or used
- Makes it easier to port code where client and server application reside on the same host
- Makes set up easier for loosely coupled systems that have the client and server application on each host
  - Define the application name with 127.0.0.1 as its IP address in the etc/hosts file used by DNS client

Loopback Example



etc/hosts file

```
my_server.com    127.0.0.1
```

Loopback Considerations

- IP address 127.0.0.1 is automatically defined by z/TPF as a local IP address
- A socket bound to 127.0.0.1 cannot be used to communicate with external applications
  - Packets with 127.0.0.1 in the IP header never flow outside the local host (never flow on a real network)
- 127.0.0.1 cannot be set as the default local IP address
- If application does not bind to a local IP address and then issues connect to 127.0.0.1, the socket gets bound to 127.0.0.1 as well
  - Socket is not bound to the default local IP address in this case

The Art of Negotiation

TCP Packet Size Negotiation

- When TCP socket is started, the client and server nodes agree upon the largest size packet that can flow across this socket
  - Maximum segment size (MSS) option in the TCP header in the TCP handshake flows
  - MSS is the maximum amount of user data in a packet and does not include the sizes of IP and TCP headers
- Client node sends its MSS value in the TCP SYN packet that starts a TCP socket
- Server node responds with an MSS value in the SYN/ACK packet
  - This value may be the negotiated value (smaller of what the client and server support) or the largest MSS that the server supports
- Both the client and server nodes use the smaller of the MSS value it advertised and MSS value received from the partner node

Old TCP Clients

- Sending an MSS option in a SYN message is optional for clients
  - Modern clients do include the MSS option
- If the client does not provide an MSS option, the MSS value provided by the server in the SYN/ACK message is used for this socket

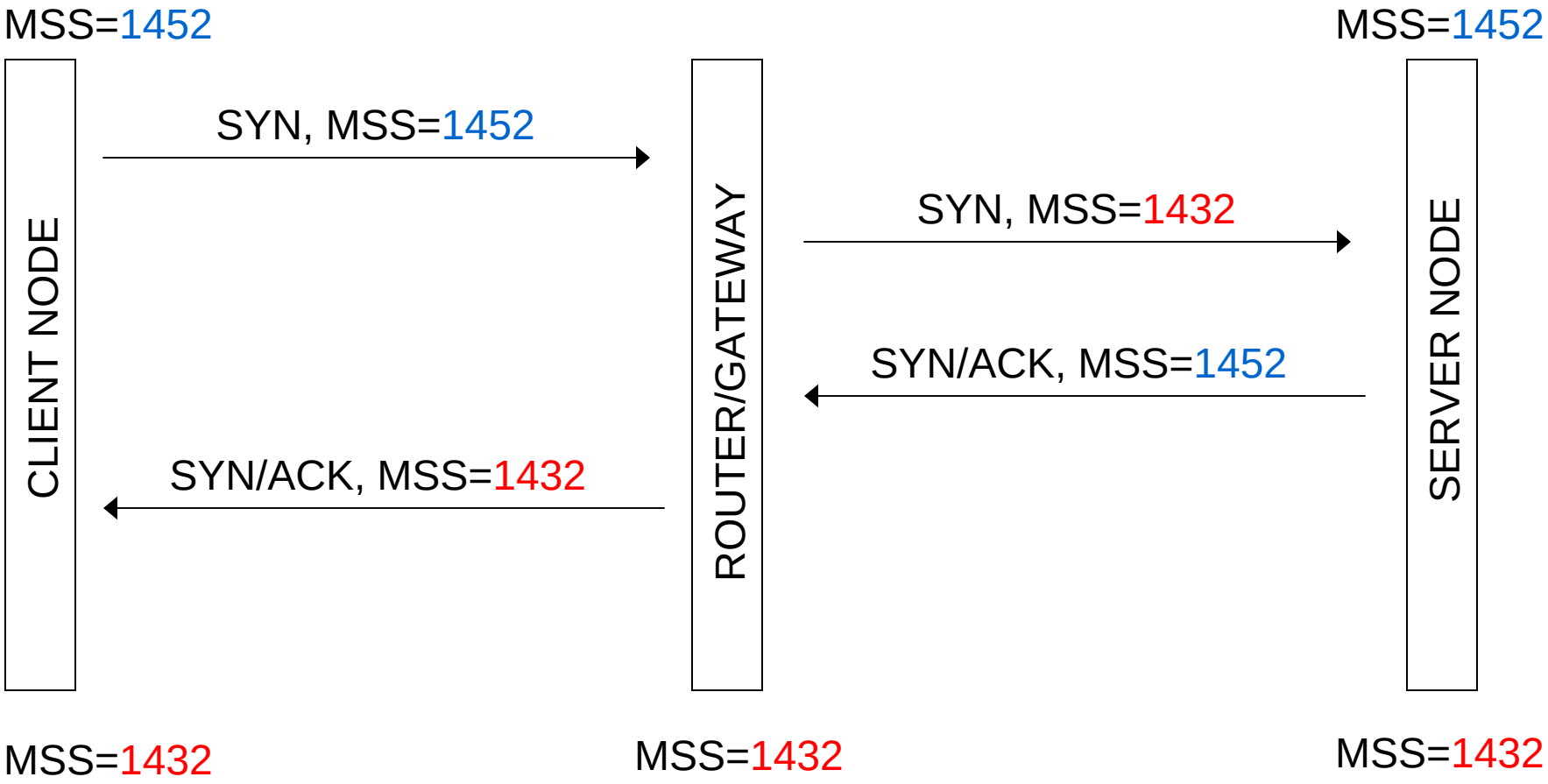
Routers and MSS

- Some router/gateway products modify (reduce) the MSS values in SYN and SYN/ACK packets
  - For example, if the router/gateway plans on adding information to the headers flowing on packets over this session
- This is OK when the client provides an MSS value in the SYN message
- Some router/gateway product enforce MSS rules
  - Discard packets larger than the agreed upon MSS value during the TCP handshake

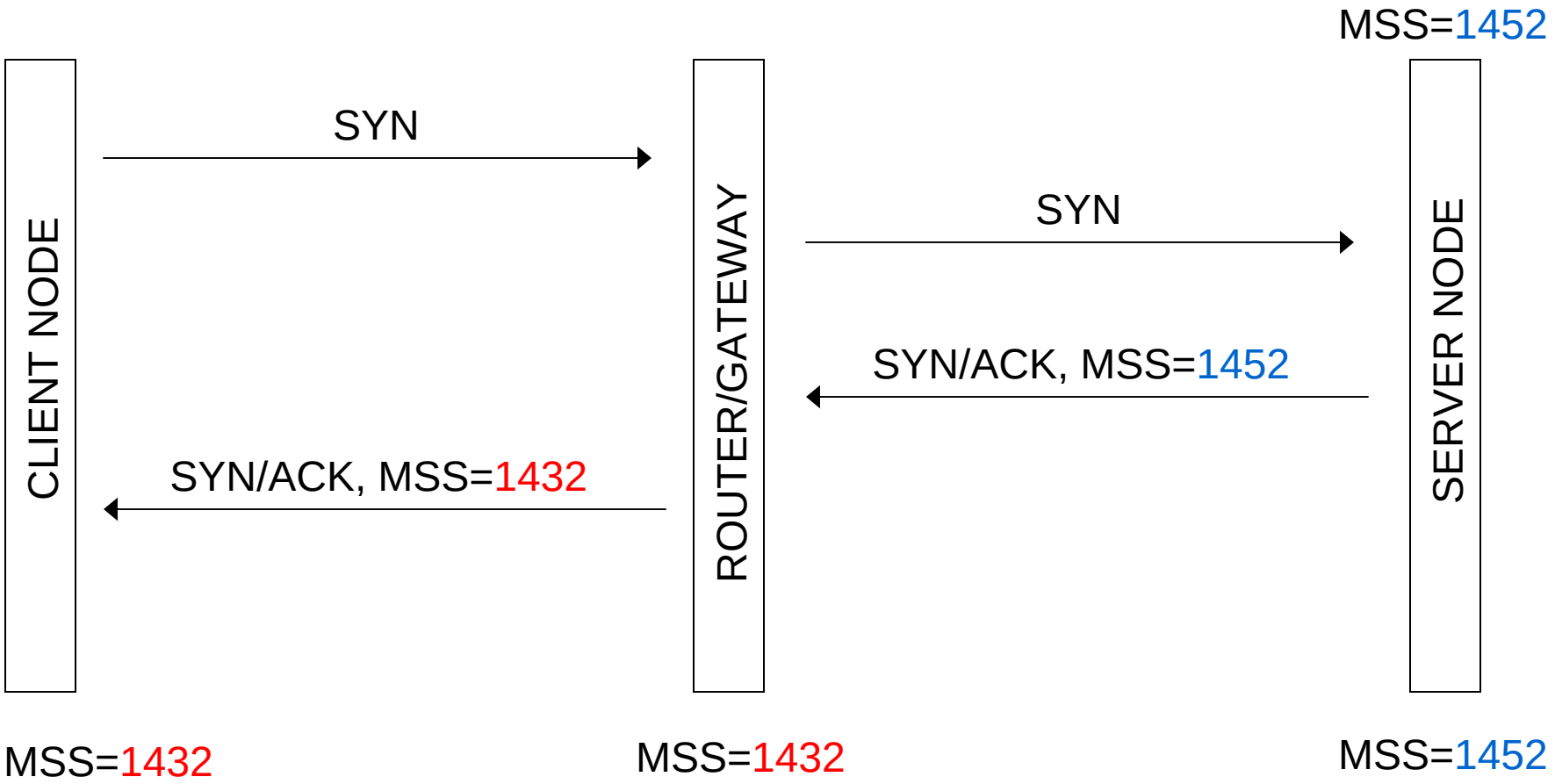
In the Following Example…

- Client and server are connected via networks that support up to 1492-byte packets
  - Maximum transmission unit (MTU) size is 1492
- Standard IP header size is 20 bytes
- Standard TCP header size is 20 bytes
- MSS value advertised by the client and server is:
  - MSS = MTU – IP header size – TCP header size
  - MSS = 1492 – 20 – 20 = 1452 bytes
- Router wants to add 20 bytes to the IP/TCP headers of each data packet; therefore, the largest MSS value the router can allow is:
  - Largest MSS = 1492 – 20 – 20 – 20 = 1432 bytes
  - MSS values above 1432 are reduced to 1432 by the router

Modern TCP Client, Semi-Intelligent Gateway → OK



Old TCP Client, Semi-Intelligent Gateway → Problem



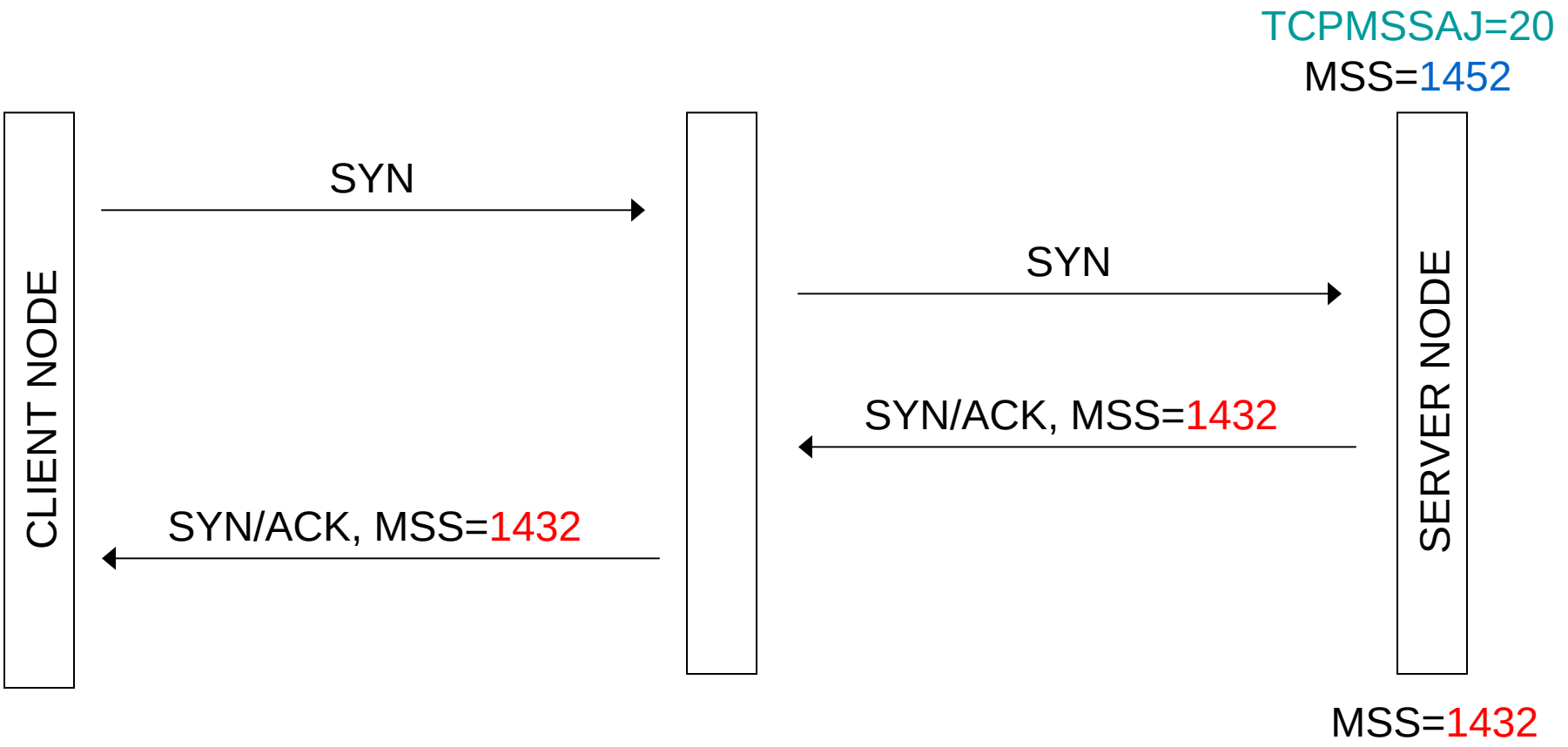
The Problem

- In the previous example, if the server sends a packet with more than 1432 bytes of user data, the router/gateway might discard the packet
  - This will eventually cause the socket to fail
- Reducing MSS value has become the default setting for some router/gateway products
  - This has resulted in problems with connections to old TCP clients
- Fixing the source of the problem is not easy
  - Changing the setting to not adjust MSS values is not always possible if the router is outside your network

The Solution

- APAR PJ31970 adds a new TCPMSSAJ parameter in CTK2
- Only used if remote client does not provide an MSS value in the SYN message that it sends to z/TPF
- The MSS value that z/TPF would normally send in the SYN/ACK message is reduced by the value of TCPMSSAJ

Old TCP Client, Semi-Intelligent Gateway, More Intelligent Server

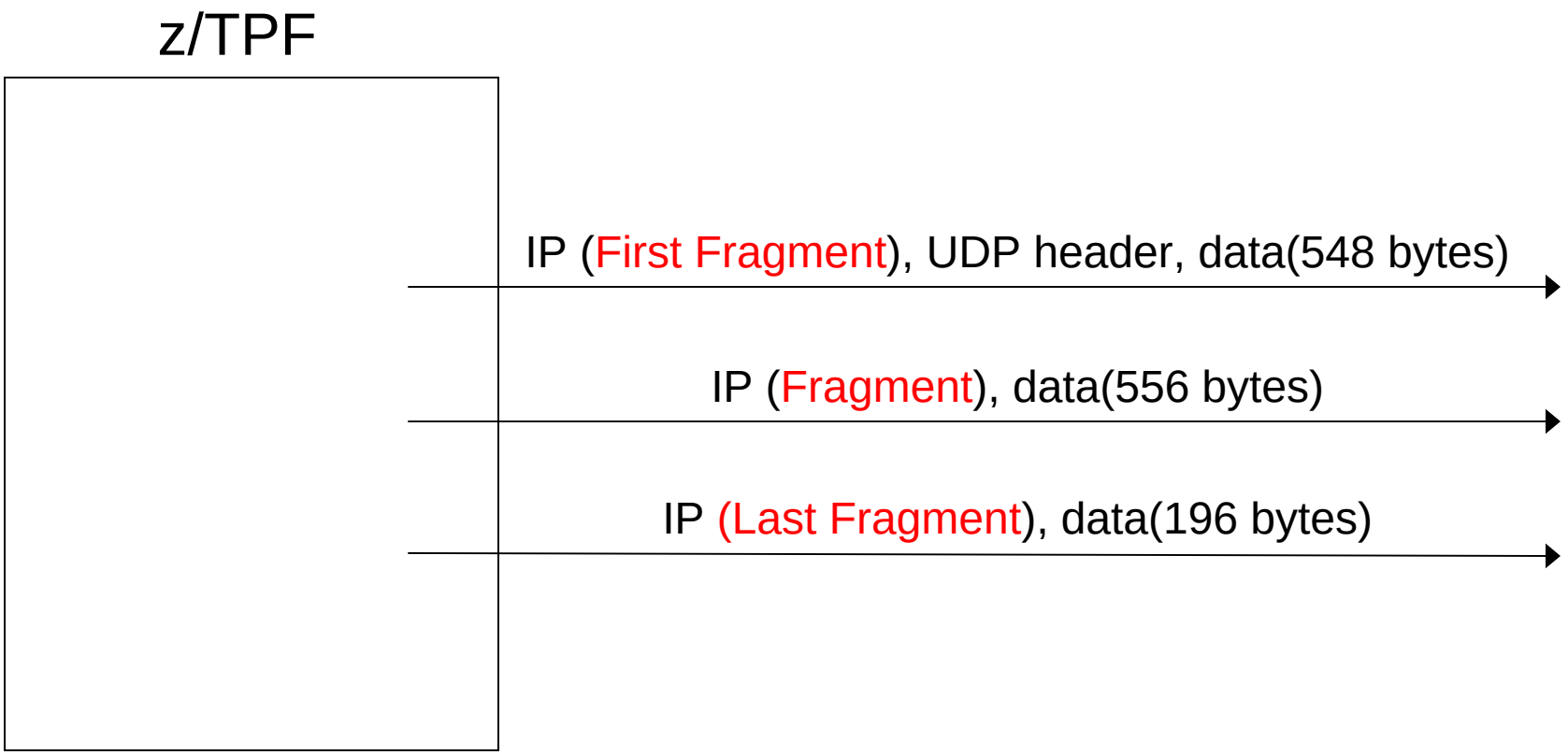


All For One and One For All

Existing UDP Packet Size Support

- TCP sockets agree upon the largest packet size (via MSS option of TCP handshake)
- There is no handshake or packet size negotiation with UDP sockets
- All IP devices (clients, servers, routers) must support packets of at least 576 bytes
- If a z/TPF application issues a send type API that would result in more than 576 bytes of data (including IP and UDP headers):
  - Data is sent as multiple packets (IP fragments)
  - Fragmentation is done by the source node (z/TPF) to avoid having to do fragmentation by routers in the network
  - Destination node must reassemble the fragments into the original message before delivering it to the application

Sending UDP Data with Existing Support



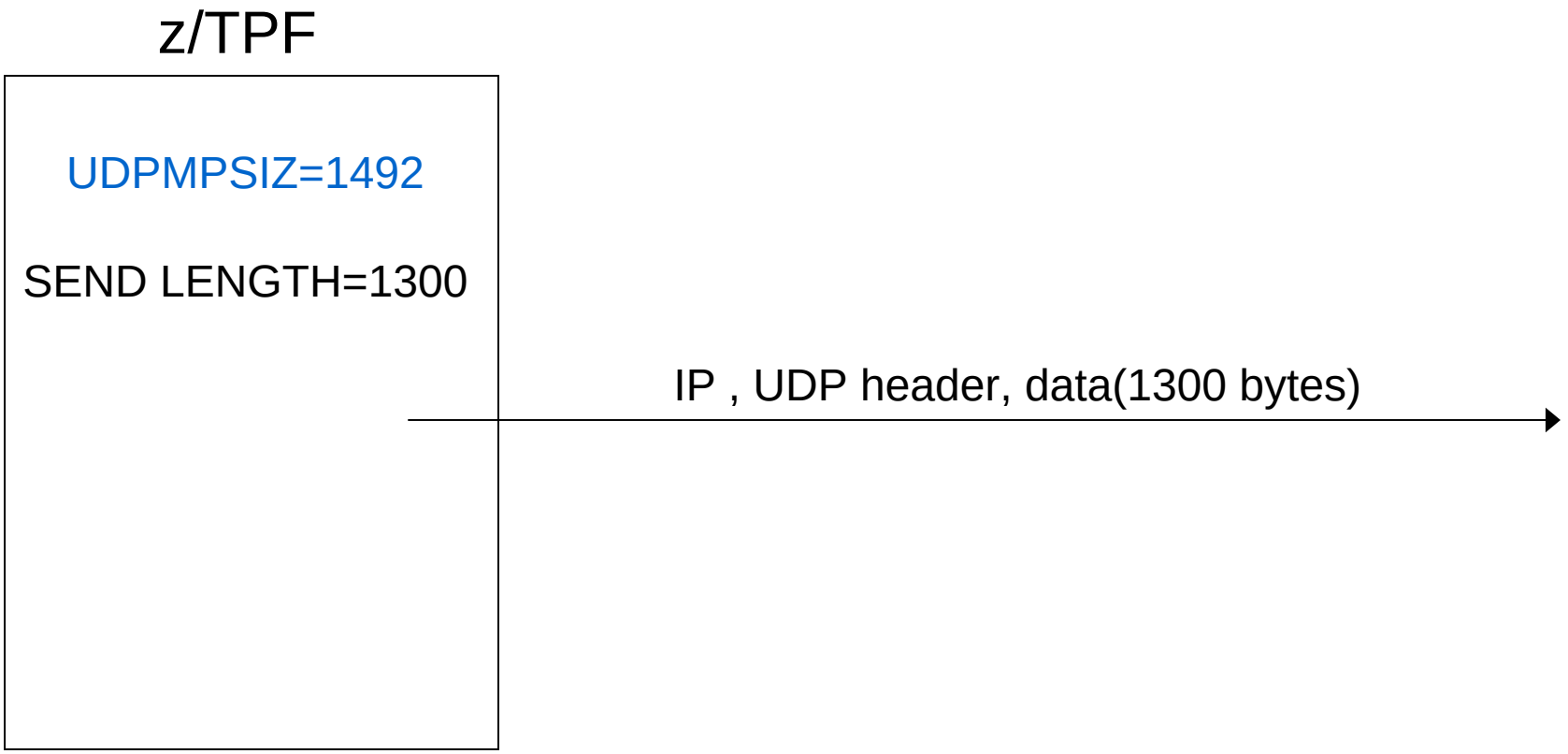
User Definable UDP Maximum Packet Size

- APAR PJ31970 allows you to customize the maximum UDP packet size sent by z/TPF
- New UDPMPSIZ parameter in CTK2
  - Default maximum packet size to use for UDP sockets
- New SO_UDPMPSIZ *setsockopt* option
  - Allows you to set the maximum packet size for a given UDP socket
- The actual maximum packet size used by a UDP socket is the smaller of the value defined for that socket (via SO_UDPMPSIZ value or UDPMPSIZ) and the MTU size of the OSA-Express connection

UDP Maximum Packet Size Considerations

- UDPMPSIZ defaults to 576
  - No change in behavior to existing support
- You should increase the maximum packet size value for UDP sockets where z/TPF sends large data and the networks between z/TPF and the remote node support larger packets
  - This reduces the number of packets that flow
  - Can eliminate fragmentation overhead (if all the data fits into one packet) or reduce fragmentation overhead
- Example where UDP application sends 20K of data
  - If UDPMPSIZ is 576, then 36 packets flow
  - If UDPMPSIZ is 1492, then 14 packets flow

Sending UDP Data with New Support



First 64-bit, Now 64K!

More UDP Message Size Information

- Data of each send type API for a UDP socket represents one message
  - Send type APIs include *send*, *sendto*, *write*
- If the message will not fit in one packet, multiple packets are sent using IP fragmentation
  - Remote end of the socket reassembles the fragments and delivers the entire message to the application when it issues a read type API
- IP fragmentation supports up to 64K of data
  - Therefore, maximum UDP message size is 64K

UDP Message Size Support Old and New

- Existing z/TPF support for UDP sockets
  - Send type APIs allow you to send up to 32K of data
  - Read type APIs allow you to receive up to 32K of data
  - Therefore, maximum UDP message size that can be sent or received is 32K
- New Support for UDP sockets (APAR PJ31970)
  - Can now send up to 64K of data on a send type API
  - Can now receive up to 64K of data on a read type API
  - Therefore, you can now exchange the largest UDP messages (64K) allowed by architecture

Considerations for Sending Very Large UDP Messages

- A send type API on a UDP socket is atomic
  - There must be enough room in the send buffer of the socket to hold all of the data passed on the API
- The socket send buffer size default is 32K
- If you have UDP applications that want to send messages larger than 32K
  - The application must increase the send buffer size of those UDP sockets using the SO_SNDBUF *setsockopt* option
- You should also increase the maximum UDP packet size (UDPMPSIZ) if possible

Considerations for Receiving Very Large UDP Messages

- When a fragmented UDP message is received from the network, it is reassembled in the socket receive buffer
  - If there is not enough room in the receive buffer, the message is truncated
- Socket receive buffer size default is 32K
- If you have UDP applications that want to receive messages larger than 32K
  - The application must increase the receive buffer size of those UDP sockets using the SO_RCVBUF *setsockopt* option

Closing Time

New Socket Closed User Exit (APAR PJ32046)

- Called whenever a socket is cleaned up
- Allows you to clean up user tables that contain the file descriptor (FD) of the socket that no longer exists
- Allows you to notify an operator or alert an automation platform to take action regarding a failed socket, application, or network component

Socket Closed User Exit Details (Segment UCLO)

- Inputs to the user exit include:
  - FD of the socket
  - Local IP address and port
  - Remote IP address and port
  - Protocol (TCP, UDP)
  - Reason why the socket is being cleaned up, such as:
    - The application issued a *close* API
    - z/TPF operator command ended the socket
    - The socket sweeper cleaned up the socket because it had not been used for a long period of time
    - z/TPF received a TCP reset (RST) for this socket

Summary

- Ability to reject SNA APPN session (PJ31939)
- Loopback IP address support (PJ32035)
- Ability to adjust TCP MSS values (PJ31970)
- Ability to set maximum UDP packet size at both system and socket level (PJ31970)
- Ability to send and receive UDP messages up to 64K in size (PJ31970)
- Socket closed user exit (PJ32046)
- All of these enhancements were requirements submitted by multiple customers

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.
- 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.