z/TPF EE V1.1
z/TPFDF V1.1
TPF Toolkit for WebSphere® Studio V3
TPF Operations Server V1.2

IBM Software Group

TPF Users Group Fall 2005

z/TPF TCP/IP Control Block and Input Message Processing

Name : Jamie Farmer
Venue : Communications Subcommittee

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

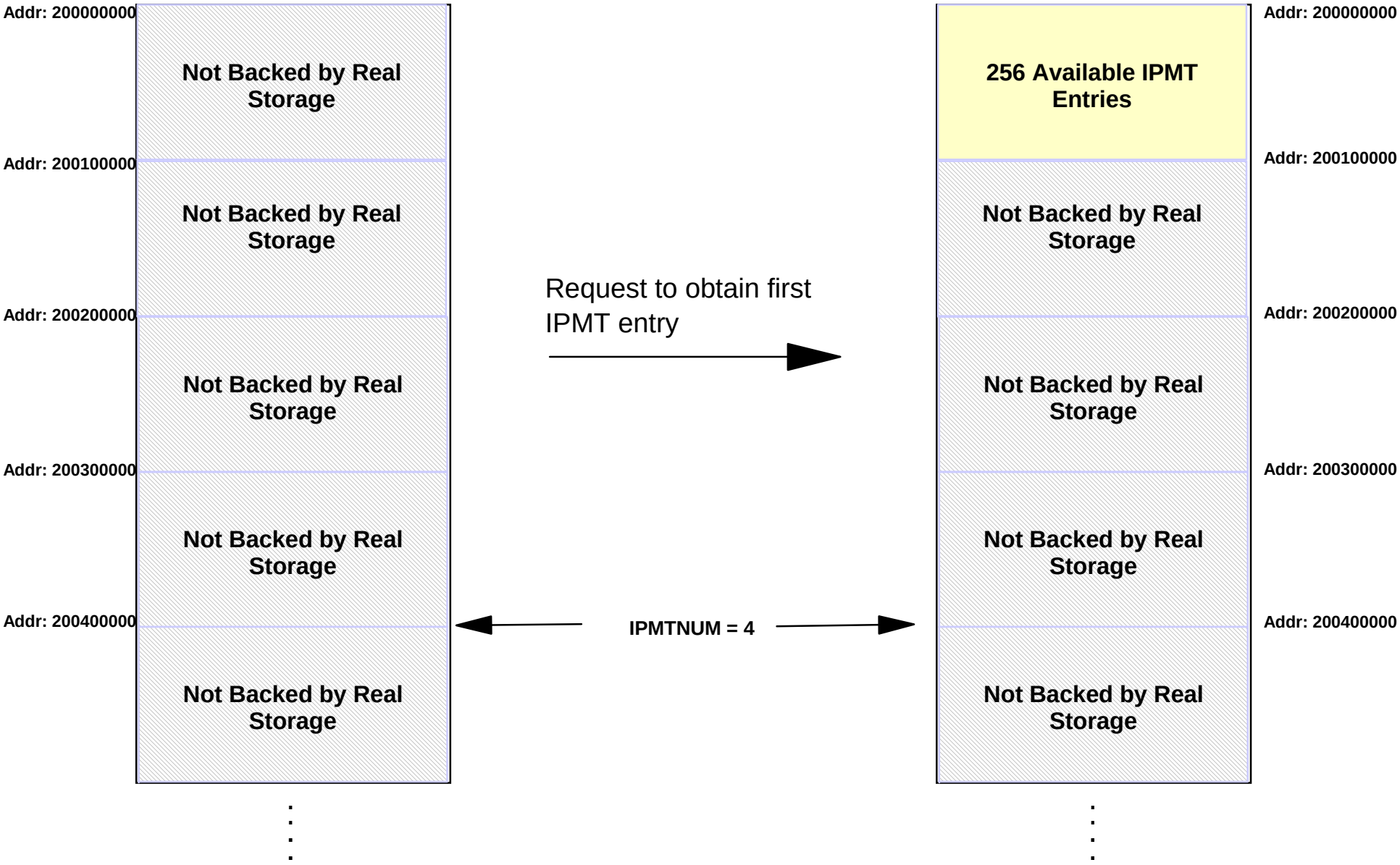
Socket Block and IPMT Table Allocation

- Tables now reside above the 2 GB bar
- Storage for tables backed by 1 MB frames
  - Only backed by real storage when needed
- IP Message Table (IPMT) entries are still 4K (same as in TPF 4.1)
- Socket Block Table entries are now 4K as well
- Ability to dynamically increase the maximum size of tables
  - Changes take affect immediately, IPL not required
  - Use the ZNKEY parameter with the IPMTNUM or MAXSOCK parameter specified

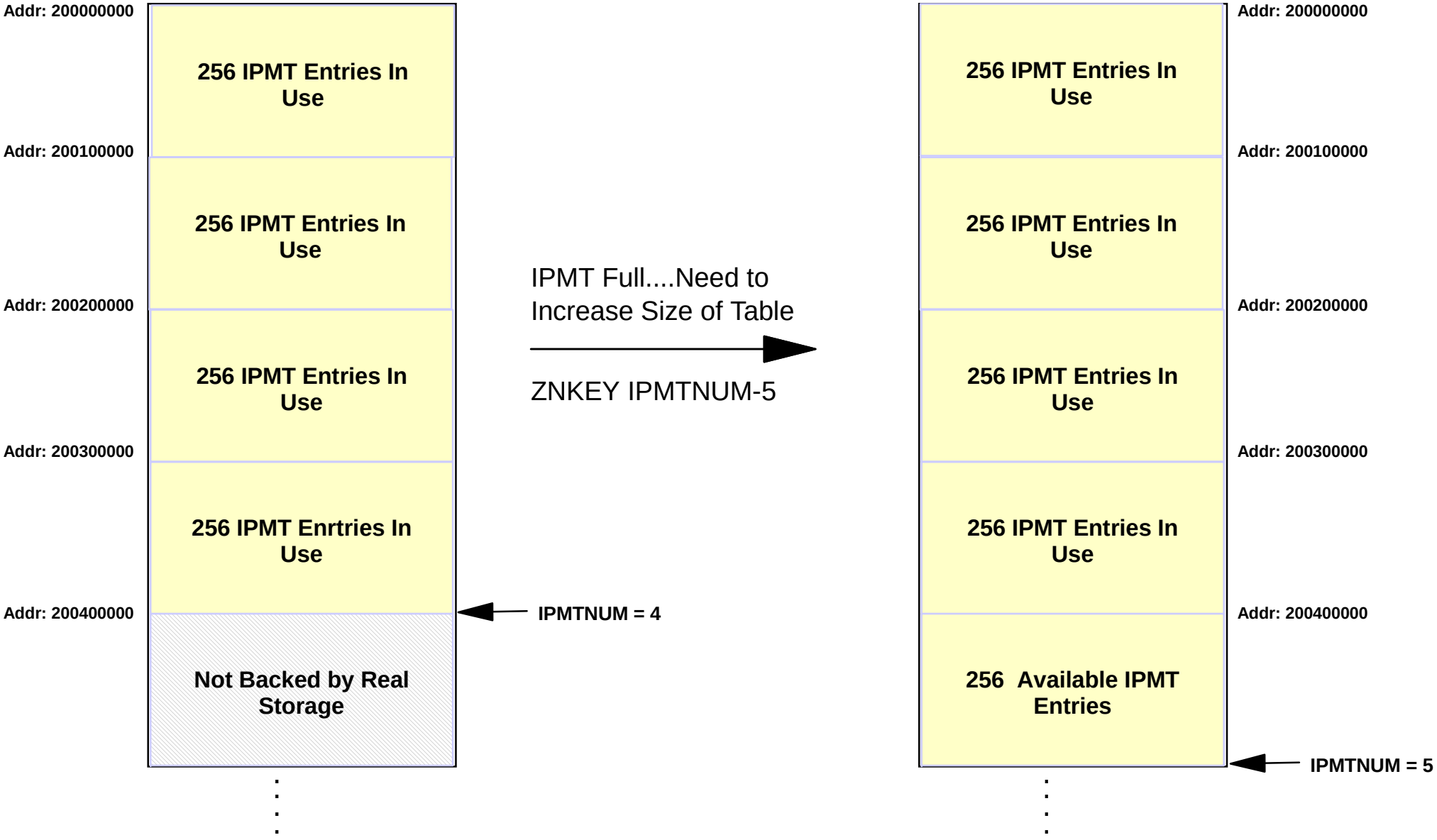
Table Allocation - The Details

- In restart, a very large virtual address space is set up for each table
- When needed, a 1 MB frame is obtained to back the first 1 MB of storage.
- TPF will continue to obtain more 1 MB frames as the storage is needed.
  - Continues until maximum size of table in CTK2 is reached.

IPMT Table Allocation Example



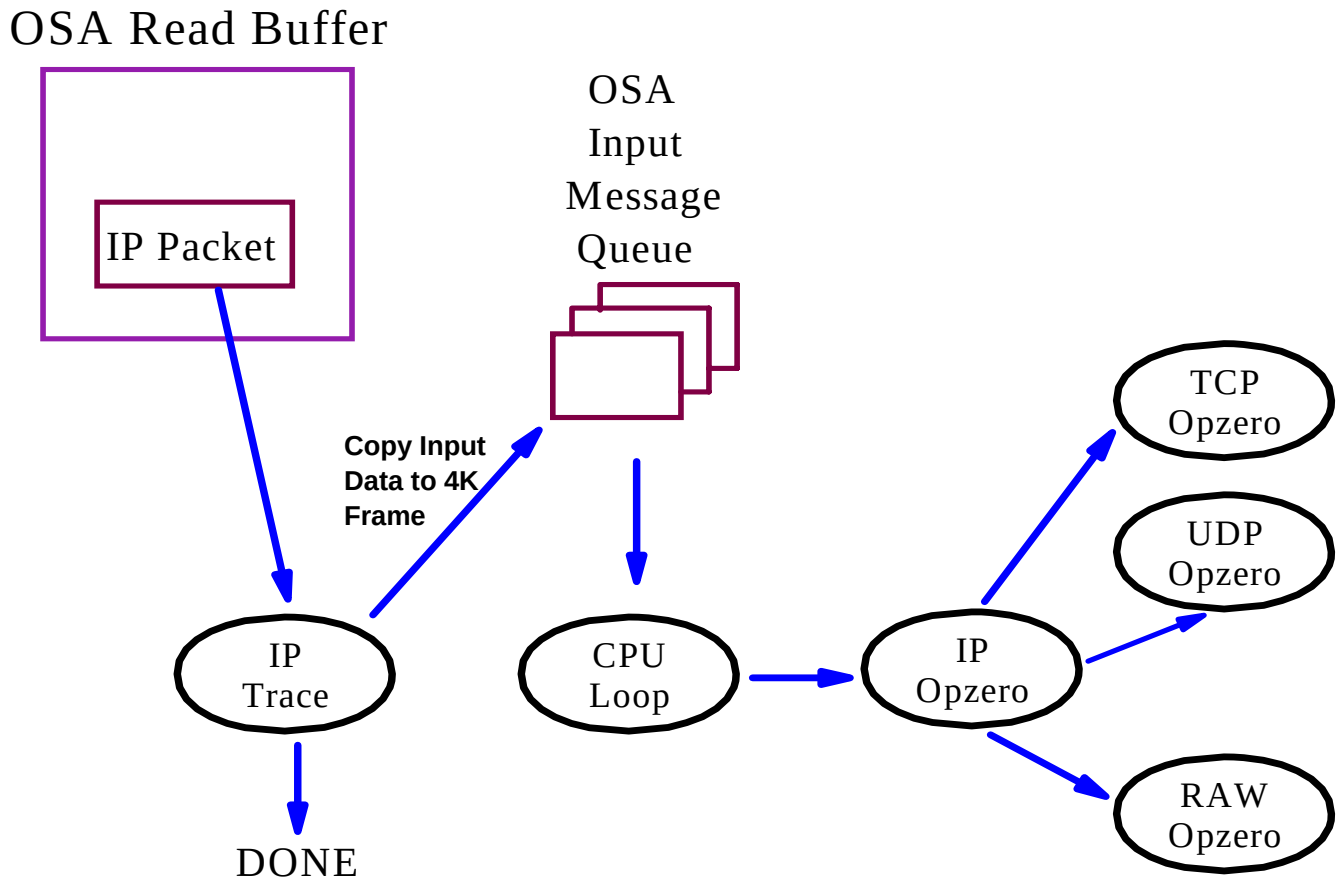
Example of Increasing Size of IPMT Table



OSA Polling Changes for z/TPF

- TPF 4.1:
  - New input messages received in the OSA read buffers are copied into 4K frames
  - Messages are placed on the OSA input message queue, which is a shared queue processed by all I-streams.
  - Special code exists in the CPU loop to process the OSA input message queue and discard messages if input list shutdown is caused by OSA input messages use of 4K frames.
- z/TPF:
  - New input messages received in the OSA read buffers are copied into IPMT blocks
  - Messages are placed on the shared input list
    - No special CPU loop code is needed anymore for processing OSA input messages.
  - The function of the OSA input message queue has changed - used less frequently
  - New OSA staging queue created to provide additional functionality

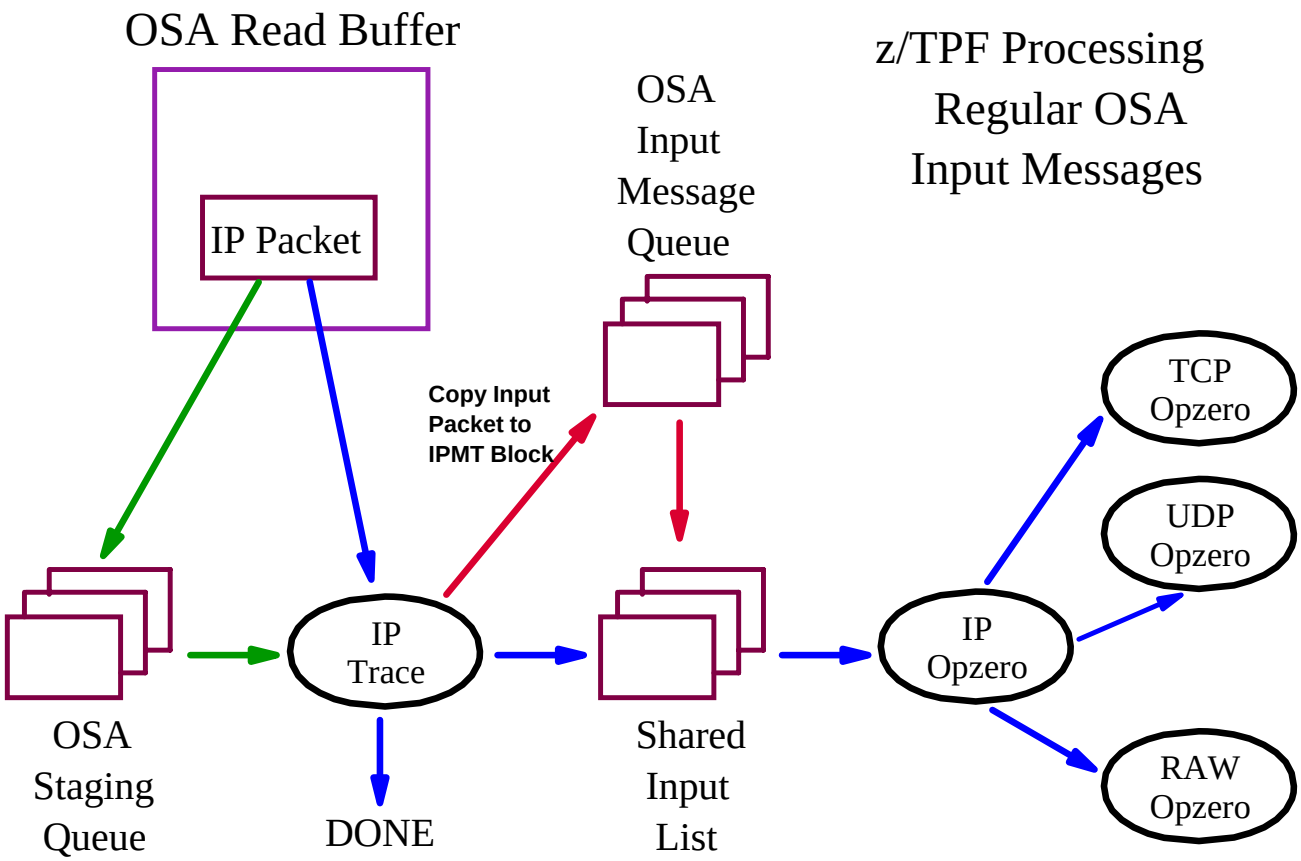
TPF 4.1 OSA Input Message Example



TPF 4.1 OSA Input Message Overview

1. OSA card places input message into OSA read buffer
   - Pre-carved storage in TPF
2. OSA Polling processes the OSA read buffers
   - Only done in CRAS state and above
   - Polling is skipped during input list shutdown and during system error (dump) processing
3. OSA Polling calls TPF's IP Trace routine
   - Starts processing packet and updates IP trace tables
4. Packets that contain data are copied into a 4K frame and placed on the OSA input message queue
5. The CPU loop checks the OSA input queue when processing the input list
   - One message is removed from the queue and passed to IP OPZERO for further processing

z/TPF OSA Input Message Example



z/TPF OSA Input Message Overview

1. OSA card places input message into OSA read buffer
   - Precarved storage, above the 2GB bar
2. OSA Polling processes the OSA read buffers
   - May be done in 1052 state if the OSA connection is defined as 1052 state capable.
     - Default is still to poll only in CRAS state or above.
   - OSA polling is now done during input list shutdown and dump processing - more on this later
3. OSA Polling calls TPF's IP Trace routine
   - Starts processing packet and updates IP trace tables
4. Packets that contain data are copied into an IPMT block and added to TPF's shared input list.
   - Maximum of 32 OSA message on shared input list at any time.
   - If there are 32 OSA messages on the shared input list, the message is added to the OSA input message queue
5. CPU loop on any I-stream will dequeue messages off the shared input list and pass the message up to IP OPZERO for further processing
   - If there are messages on the OSA input message queue, one message is moved to the shared input list

What is the OSA Staging Queue?

- Queue is used to save messages that cannot be processed right away
  - During Input List Shutdown
  - During System Error Processing
    - Except during catastrophic dump processing
- To ensure messages are processed in order, if any messages are on the staging queue, new ones will be queued behind them.
- Once the condition clears that is causing message build-up on the OSA Staging Queue, TPF starts processing the queued messages
  - Processing is throttled to avoid resource overload and flooding the system.

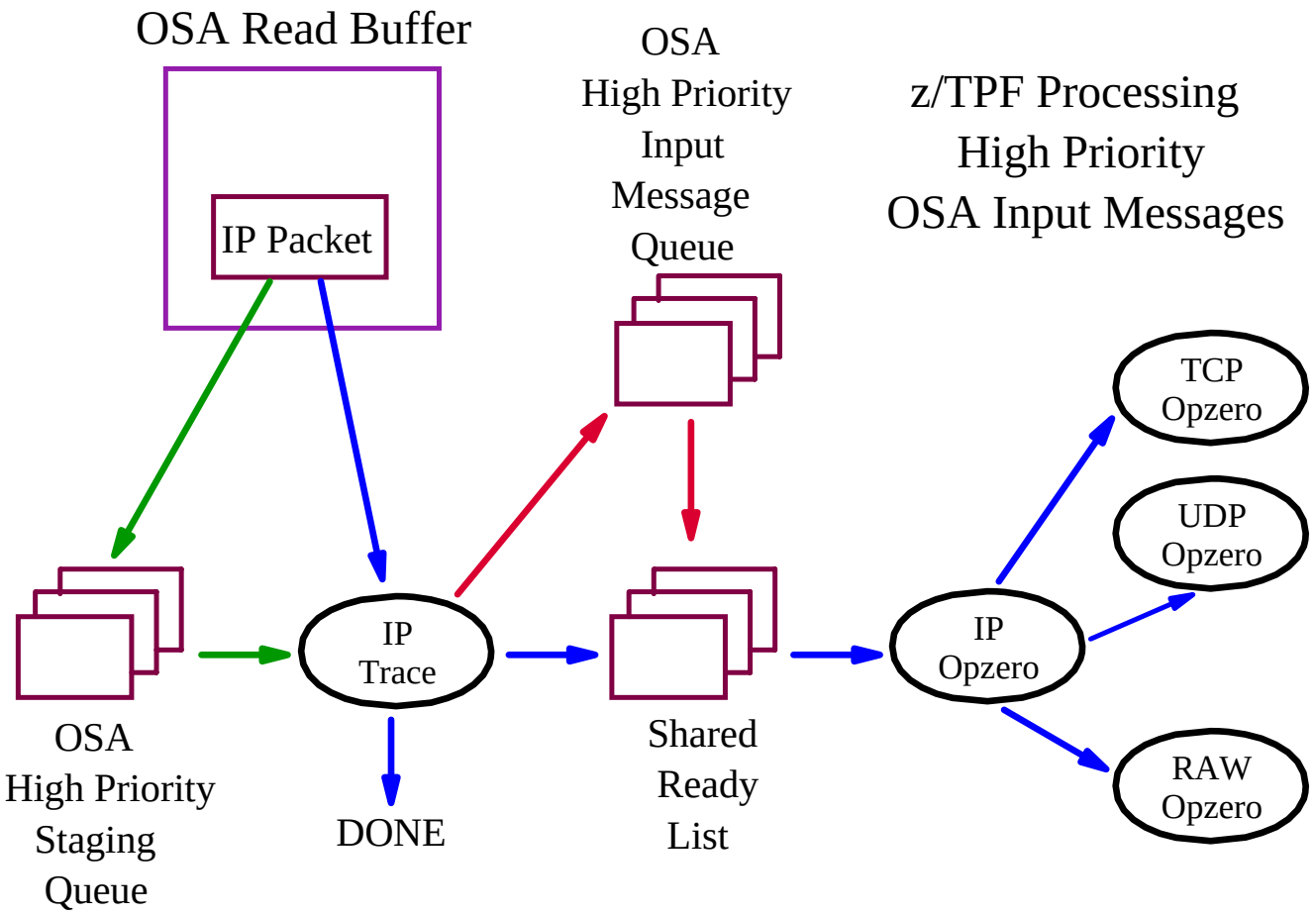
OSA Input Message Priority

- Ability to define input messages with a priority
  - Done by application using TPF's Network Services Database (NSD)
  - Can be overridden for individual sockets using the setsockopt() API
- Priority values of 2-9 for input messages is a "discard priority"
  - If the IPMT table becomes full (IPMTNUM in keypoint 2 is reached), TPF will begin to discard messages based on the priority
    - The higher the priority value, the likelihood of discard is increased (For example, messages with a priority of 9 are dicarded first).
  - These messages are processed equally
- A Priority value of 1 indicates a high priority message (or application)
  - Input messages of this priority will never be discarded.

z/TPF OSA High Priority Input Messages

- Specifying a priority of 1 for a socket or application defines its messages as high priority
  - TPF continues to read in and process high priority input messages even during input list shutdown.
- High priority input messages will never be discarded due to an IPMT full condition.
- Separate set of OSA input and staging queues are used to process high priority messages.

z/TPF OSA High Priority Input Message Example



z/TPF High Priority Input Messages Overview

1. OSA card places input message into OSA read buffer
   1. Precarved storage, above the 2GB bar
2. OSA Polling processes the OSA read buffers
   1. May be done in 1052 state if the OSA connection is defined as 1052 state capable.
      - Default is still to poll only in CRAS state or above.
3. OSA Polling calls TPF's IP Trace routine
   - Starts processing packet and updates IP trace tables
4. Packets that contain data are copied into an IPMT block and added to TPF's shared ready list.
   - Maximum of 32 OSA message on shared ready list at any time.
   - If there are 32 OSA messages on the shared ready list, the message is added to the OSA high priority input message queue
5. CPU loop on any I-stream will dequeue messages off the shared ready list and pass the message up to IP OPZERO for further processing
   - If there are messages on the OSA high priority input message queue, one message is moved to the shared ready list

Implications of High Priority Applications

- Caution must be taken when defining applications or sockets as High Priority
  - You must analyze the application design and resource usage
- Use in applications where the inbound message can be read by the application to free up resources.
  - Example: MQ Sender Channel
    - Sends a batch of MQ messages out to the remote node
    - These messages are saved in TPF core blocks until the MQ acknowledgment is received from the remote node
    - If normal priority, the MQ acknowledgement cannot be processed during input list shutdown. If processed the acknowledgement would free up resources (storage) and likely get TPF out of shutdown.
    - By marking the sender channel sockets as High Priority, we avoid this deadlock condition
    - This processing has already been implemented on z/TPF

Summary

- Dynamic allocation of IPMT and Socket Block tables
- Input Message Processing
  - Enhanced CPU Loop Processing
  - New TCP/IP Input Message Priority
    - Ability to discard inbound messages based on priority during an IPMT full condition.
    - Ability to read and process high priority messages even while in input list shutdown.
  - Polling is done even while in Input List shutdown and System Error processing.
    - Minimizes packet loss during these situations

Trademarks

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.