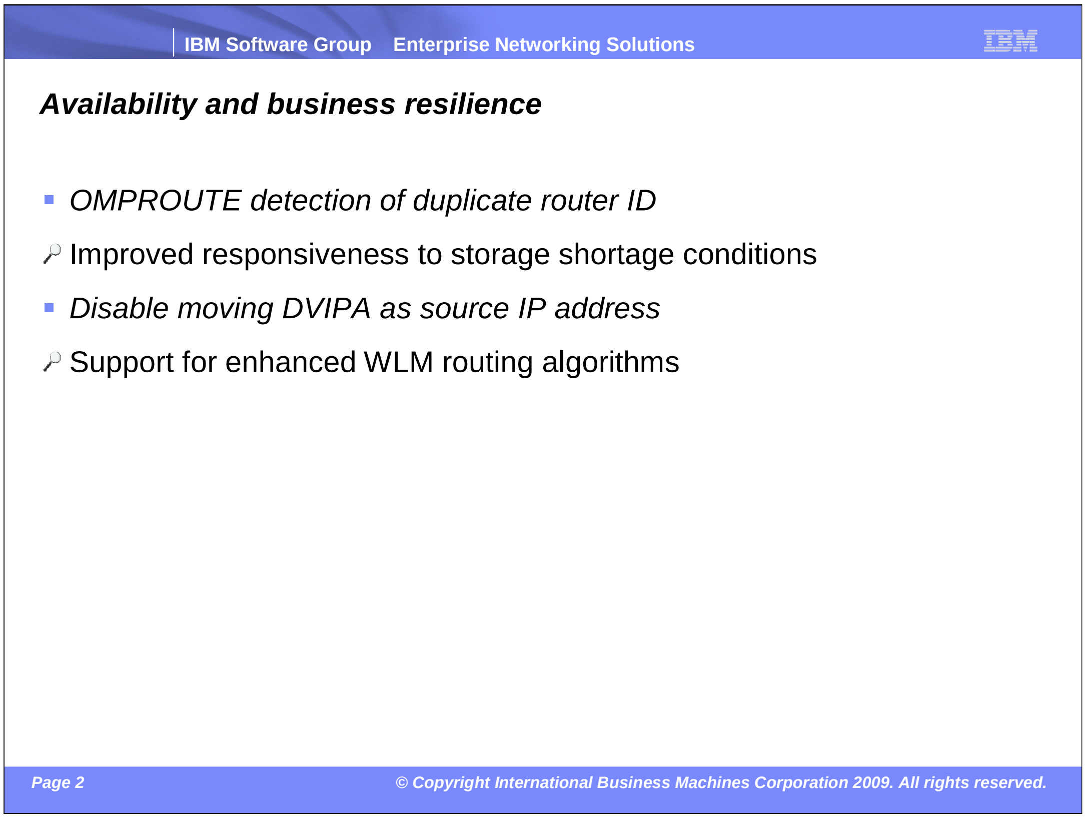
This presentation describes the enhancements to the Communications Server in z/OS V1R11 for availability and business resilience. This area covers enhancements that make the z/OS networking environment more resilient to various types of abnormal conditions that might impact the overall availability of the system.



The main enhancements within this area are those listed on this slide.

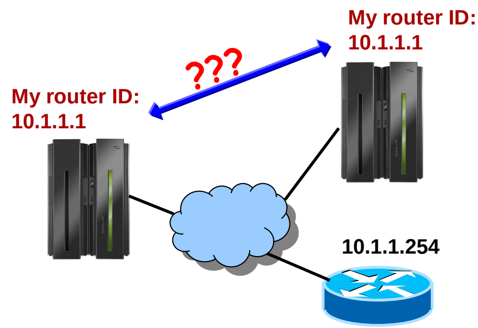
Special emphasis in this release has been put into making the TCP/IP environment able to better cope with situations where storage resources are constrained – typically caused by extreme workload conditions.

All items except the moving DVIPA item are discussed in more detail in this presentation.

Here is a brief description of the DVIPA function. When a non-distributed DVIPA moves from one stack to another (takeover and takeback), there is a possibility that it might transition from ACTIVE status to MOVING status. This occurs if there were any connections still active on the stack giving up the DVIPA. The stack that takes over the DVIPA will then forward any packets it now receives to the MOVING stack for those connections. The ultimate goal is typically for the MOVING stack to finish with its existing connections and then transition back to BACKUP status. However, the stack does not exclude MOVING DVIPAs from TCPSTACKSOURCEVIPA consideration. Therefore, it is possible this DVIPA will remain in MOVING status for an indefinite period. This violates the intent of MOVING status. For reference, there is another state a DVIPA can be in called QUIESCING. It is similar to MOVING in that when all connections to this QUIESCING DVIPA are done the DVIPA is removed. QUIESCING DVIPAs are NOT considered for TCPSTACKSOURCEVIPA consideration. As of V1R11, MOVING DVIPAs are no longer considered for TCPSTACKSOURCEVIPA consideration.

OMPROUTE to aid in detecting duplicate router IDs

- If multiple OSPF routers use the same router ID, routing problems will occur
  - Routes are continuously added and deleted by neighboring routers
  - Increased OSPF traffic as designated router floods new LSAs
  - Packet loss or connectivity loss depending upon routing environment
- Problem can be difficult to diagnose due to varied symptoms
- OMPROUTE will detect when another adjacent OSPF router is using the same router ID as this OMPROUTE instance
- Message EZZ8165I is issued to the console once every 10 minutes per OSPF version (IPv4 or IPv6)



Although router IDs should be unique, sometimes multiple OSPF routers are configured with the same router ID. This will cause routing problems.

The designated router is getting router LSAs from each router with different information. The router will update its routing table and then flood the updated LSAs to other routers in the area. This will cause routes to cycle from active to non-active states, or be constantly added then deleted from the network topography, causing excessive network disruption. Packets can be lost in a routing loop or dropped as these routes consistently change. This can cause intermittent ping timeouts or poor performance on connections. The symptoms will stop if the duplicate router is stopped. This type of problem can be difficult to diagnose.

The picture shows three OSPF routers, however two of them are using the same router ID.

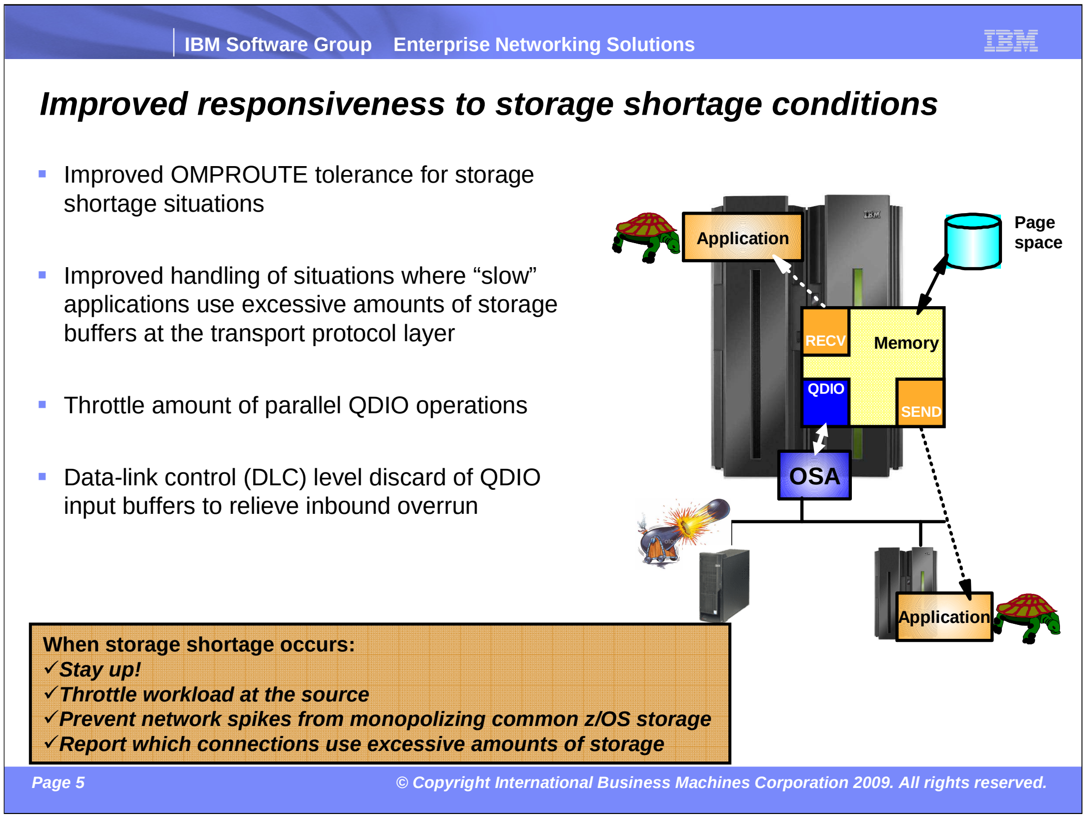
In z/OS V1R11, OMPROUTE will issue message EZZ8165I when OSPF packets are received from a adjacent router with the same router ID OMPROUTE is using. EZZ8165I is issued to the console once every 10 minutes per OSPF version. So if a router is using the same router ID for both IPv4 and IPv6 OSPF, message EZZ8165I is issued twice.

Actions to take when you see message EZZ8165I

1. Verify that the router ID in this OMPROUTE is correct
   - Do not use a DVIPA address for a router ID – EZZ8134I
2. Use neighbor displays to find who else is using the duplicate router ID
   - Use packet trace or sniffer trace
3. Configure the router with the correct router ID

Message EZZ8165I only detects this situation has occurred. Unfortunately, OMPROUTE cannot resolve this problem dynamically. The first step is to verify that the router ID being used by this OMPROUTE is correct. If the router ID in message EZZ8165I is not the expected router ID, the configuration needs to be verified. OMPROUTE should be configured with a router ID so the same router ID is used by this OMPROUTE instance. The router ID should not be a DVIPA address, as this address can be active on multiple TCPIP stacks. Message EZZ8134I is issued when OMPROUTE started if a DVIPA address was used.

If the router ID in message EZZ8165I is correct for OMPROUTE, someone else in the OSPF autonomous system is incorrectly using the router ID. The designated router should be checked first, using neighbor displays. You are trying to correlate the router ID with an interface address to determine which router is incorrectly using the router ID. A packet trace or sniffer trace can also be used to find the IP address. Once the router has been identified, the router can be configured with the correct router ID.

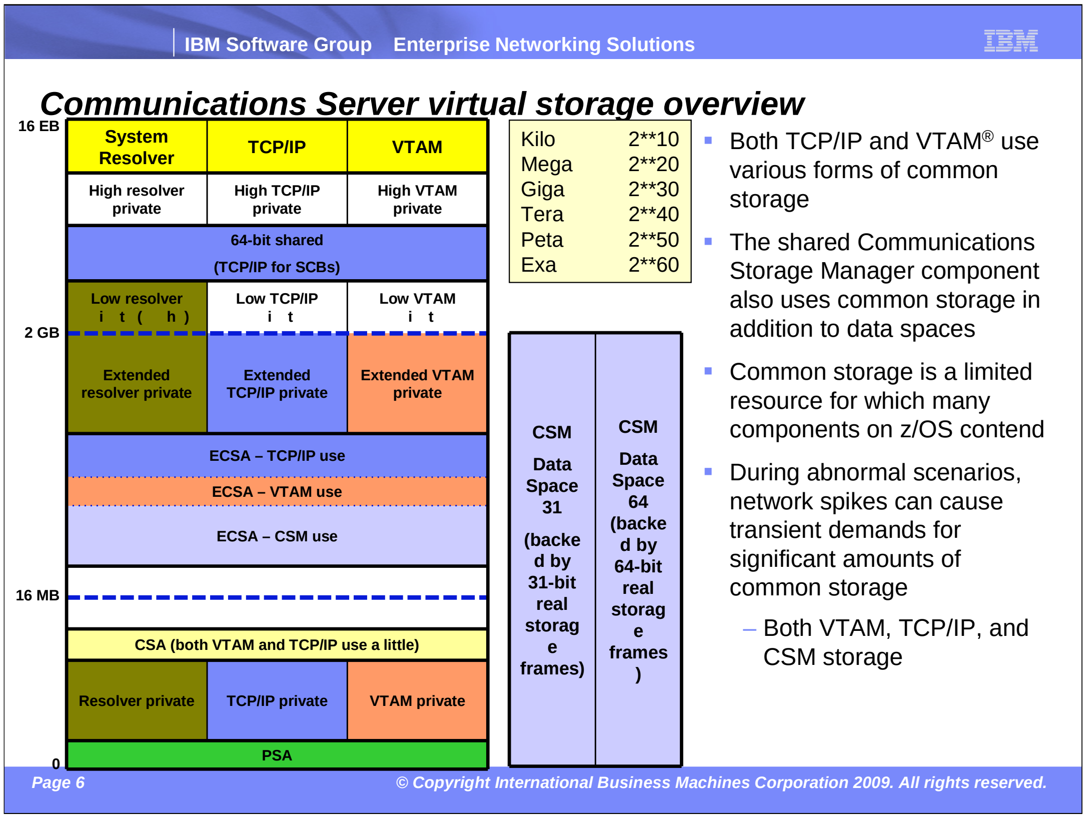


There have in the past been situations where OMPROUTE, in storage constrained situations, was not able to get storage for sending its periodic hello packets. This led to neighboring routers believing z/OS had gone down – even though it was still up-and-running, however in a very constrained state.

In this release z/OS Communications Server attempts to control situations that can lead to storage constraints by detecting early signs of extreme conditions for ECSA storage. If such signs are detected, z/OS Communications Server will attempt to limit inbound traffic at the source: the inbound QDIO network interfaces. This is done by capping the processor resources that are used for processing inbound QDIO buffers. z/OS Communications Server will further start dropping inbound packets (the normal IP-based recovery action of constrained nodes). The remote TCP will retransmit in an attempt to recover the lost packets.

OMPROUTE storage processing has been changed to ensure that OMPROUTE can get storage for sending its hello packets even in cases where the TCP/IP environment is severely storage constrained.

Another factor that can impact storage usage is applications that either do not read the inbound data that is queued for them in the receive buffers or accumulate large amounts of data in the send buffers.



This slide provides an overview of the Communication Server's virtual storage model with a focus on common storage since common storage is a restricted resource governed by various installation defined limits.

VTAM and TCP/IP each obtain ECSA storage. Most of this ECSA storage is reported as belonging to the MVS Master address space and not TCP/IP or VTAM because the storage is persistent. Note that an RMF report of common storage usage does not match what is reported by a D TCPIP,,STOR command!

Generally all Communications Server storage is obtained in key 6. You can see how much storage is owned by key six in an RMF monitor II virtual storage report. This is probably Communications Server storage (VTAM, TCP/IP, and CSM).

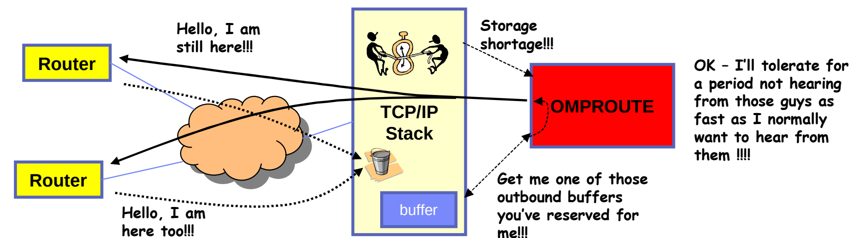
CSM is a communications server buffer pool manager which maintains the buffers in a combination of ECSA and data spaces Both data spaces are normal z/OS data spaces that each provide 2GB of virtual storage. The fact that one of those is named 'data space 64' does not mean it provides 64-bit virtual storage, it just means that this data space might be backed by 64-bit real storage frames. Some CSM users can be restricted to only work with 31-bit real storage frames, which becomes important when CSM buffers are fixed for I/O operations. TCP/IP allocates all of its CSM data space buffers in the 64-bit backed data space.

The best way to monitor CSM storage availability and usage is through the D NET,CSM and D NET,CSMUSE commands.

IBM Software Group Enterprise Networking Solutions

Storage shortages and OMPROUTE

- OMPROUTE and the TCP/IP stack work together to make OMPROUTE more tolerant of storage shortage conditions:
  - TCP/IP stack informs OMPROUTE of stack storage shortage conditions
  - During a storage shortage, OMPROUTE temporarily suspends requirement for periodic routing updates from neighbor routers
  - TCP/IP stack ensures that dispatchable units for OMPROUTE can always obtain the control blocks that they require
  - TCP/IP stack satisfies storage requests for OMPROUTE as long as storage remains available
- These changes temporarily keep OMPROUTE from timing out routes due to lack of routing updates from neighbor routers during a storage shortage
- Decreases likelihood of OMPROUTE exiting or failing to send routing updates to neighbor routers



Page 7

The changes to make OMPROUTE more tolerant of storage shortage conditions consist of changes to both OMPROUTE and the TCP/IP stack.

New notifications are sent to OMPROUTE by the TCP/IP stack to inform OMPROUTE when the stack enters or exits a storage shortage condition. OMPROUTE uses these notifications to temporarily suspend, during a storage shortage, the requirement that it receive periodic routing updates from neighbor routers.

The TCP/IP stack ensures that there are always control blocks available for dispatchable units doing work for OMPROUTE. In addition, the stack satisfies requests for stack storage made on behalf of OMPROUTE as long as storage remains available. Requests made on behalf of other applications are not satisfied during a storage shortage.

These changes temporarily keep OMPROUTE from deleting routes during a storage shortage due to not receiving periodic routing updates from neighbor routers. In addition, they decrease the likelihood that OMPROUTE will exit or fail to send routing updates to neighbor routers during a storage shortage.

If a storage shortage persists for longer than five minutes, OMPROUTE will automatically resume the requirement that it receive periodic routing updates from neighbor routers. At this point, routes will begin to be deleted from the stack route table.

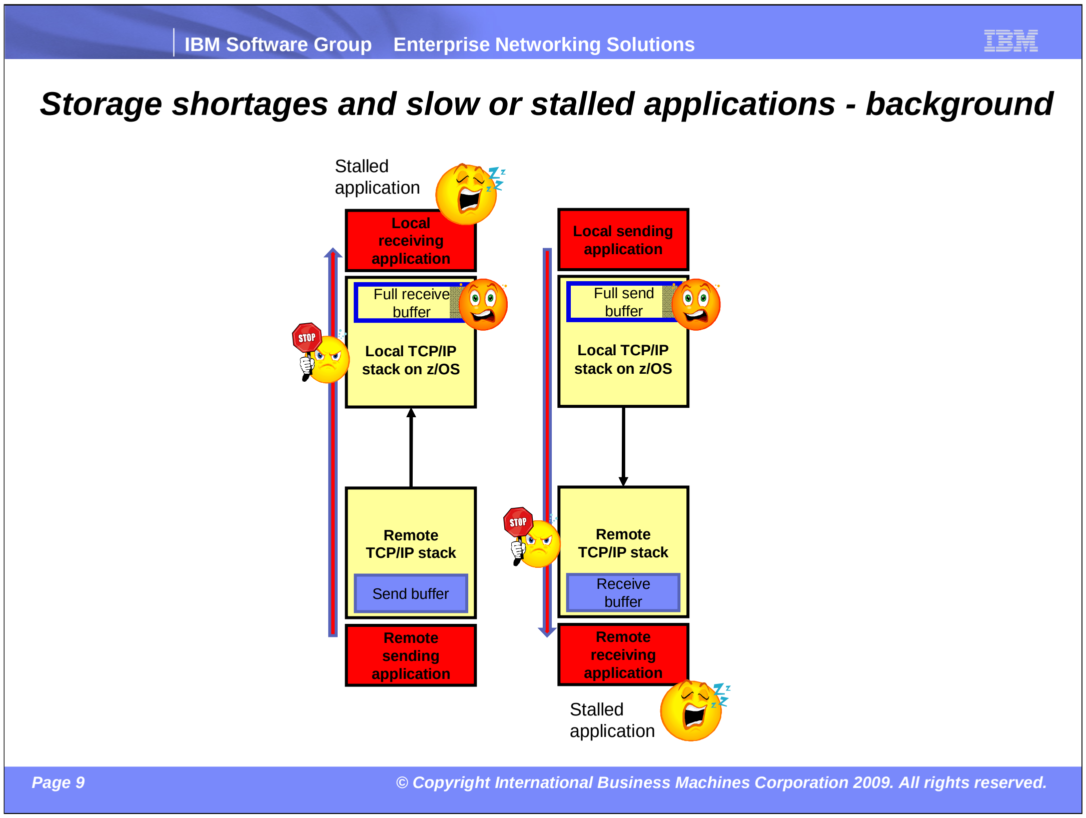
There is no configuration needed to enable the changes to improve OMPROUTE tolerance of storage shortages.

Storage shortages and OMPROUTE - considerations

- Display changes in OMPROUTE during a storage shortage
  - Time since last Hello
  - Age of RIP routes stops increasing
- Considerations
  - OMPROUTE is more tolerant of storage shortages but they still can cause problems
    - For example, if stack storage is totally exhausted
  - If storage shortage persists more than five minutes
    - OMPROUTE resumes period updates

There are a couple of things that you might see in OMPROUTE display reports during a storage shortage that you will not otherwise see. First, when displaying the detailed information for an OSPF neighbor, you might see the time since the last Hello packet was received from a fully-adjacent neighbor increase to be larger than the configured Dead Router Interval. Second, when displaying routes, you will see that the age of RIP routes stops increasing.

There are a couple of things to think about relative to this function. First, the changes that have been made are intended to improve OMPROUTE tolerance of storage shortages, but problems can still occur. For instance, OMPROUTE will still exit if stack storage becomes totally exhausted such that OMPROUTE can no longer send data. Also, if a storage shortage persists for longer than five minutes, OMPROUTE resumes the requirement that it receive periodic routing updates from neighbor routers. At that point, local routes begin to be deleted from the stack route table if the periodic updates from neighbor routers are not reaching OMPROUTE because they are being dropped by the stack.



The TCP layer of the stack maintains a receive queue and a send queue for each TCP connection. The receive queue holds data that has been received from the remote stack but has not yet been read by the local application. The send queue holds data that has been sent by the local application but has not been sent by the TCP layer to the remote stack or has not been acknowledged by the remote stack.

An application that is not reading data that it is being sent causes data to remain on the receive queue of its local TCP layer. In addition, once the receive queue of the local TCP layer is full, data remains on the send queue of the remote TCP layer.

Changes were made in V1R10 to reduce the amount of storage that is held due to old data on TCP receive queues. However, those changes had no effect on storage being held due to old data on TCP send queues. Two problems are addressed in this release in the area of storage used for the TCP queues.

First, an excessive amount of fixed storage was held on the TCP send queues when there were many connections with a lot of data on their queues. The problem was exacerbated when old data remained on the queues due to remote applications that were not reading. When fixed storage was constrained, as indicated by message `IVT5592I`, no action was taken on the TCP queues to resolve the constrained state. In addition, TCP applications failed due to the constraint.

Second, it was difficult to determine which TCP connections had old data on the queues. To find the offending connections, you had to scan all of the connections in a Netstat ALL report, looking for the connections with old data.

Storage shortages and slow or stalled applications

- Data in a send buffer is page fixed – awaiting IO operations to be initiated
  - When application is not making progress or fixed storage is constrained, all new data added to TCP send queue is marked as pageable
  - When storage becomes constrained, all unsent data on send queues for all non-local TCP connections is marked as page-able
  - Before data is sent to remote stack it is changed back to fixed, as required by the DLC
- To identify local applications using excessive storage, alerts now indicate which TCP queue is in a constrained state
  - Indicate old data on send or receive queue
  - Identify connection (connection id, job name, addresses, ports)
  - Constrained state entry and exit indicated
  - Issued to syslogd using TRMD

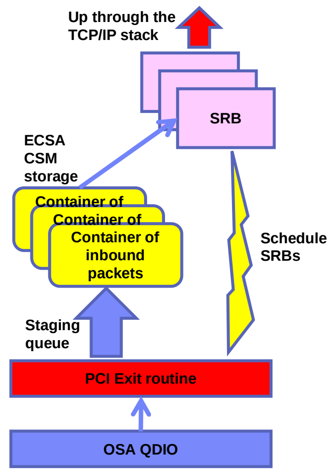
To address the problem where it was difficult to determine which TCP connections had old data on the queues, alerts are now issued to syslogd using TRMD. Alerts are issued when a TCP send or receive queue enters constrained state, indicating that there is old data on the queue. Alerts are also issued when the queue later exits constrained state. All alerts include the information necessary to identify the connection.

There are two thresholds that are used to indicate old data on a TCP send or receive queue. The first threshold is when the quantity of data on the queue equals the queue buffer size and the oldest data on the queue is at least 30 seconds old. The second threshold is when there is any amount of data on the queue and the oldest data on the queue is at least 60 seconds old. An alert indicating entry into constrained state is issued when a queue exceeds 90 percent of either threshold. An alert indicating exit from constrained state is issued when a queue falls back below 80 percent of both thresholds.

Storage shortages and QDIO device driver - background

Before z/OS V1R11, there was no limit on

1. Number of SRBs
2. Number of containers on the staging queue



The QDIO PCI exit is a traditional interrupt handler called by z/OS while holding the Unit Control Block (UCB) lock. Being an interrupt handler, it is not supposed to completely process the inbound data but instead perform a limited number of tasks to accept the interrupt and continue. It runs disabled for interrupts and uses the z/OS SCHEDULE macro to initiate enabled threads (SRBs) to complete read processing.

The PCI Exit routine packages inbound data, queues it onto a staging queue, replenishes read buffers, and schedules enabled processes (SRBs) to complete processing. The SRB threads continue read processing through the TCP/IP stack.

If the PCI exit finds the SRB available queue empty, it exits which allows the SRBs that are currently scheduled or dispatched to pick up the newly queued work.

Lock contention has stalled these threads, leading to tens of thousands of threads competing for system resources.

In earlier releases, there was no limit to the number of packets on the staging queues. Storage shortages have occurred due to high volumes of inbound packets.

Storage shortages and QDIO device driver actions in z/OS V1R11

- Number of parallel SRBs is now limited to:
  - For 1 Gigabit Ethernet, maximum of four threads per QDIO data device
  - For ten Gigabit Ethernet and HiperSockets™, maximum threads per QDIO data device is
    - Min(LPAR processors + 1, 4) * 2
- Use of CSM storage for containers on the staging queue is also now limited:
  - For 1 Gigabit speed OSA-Express
    - Two MB if CSM critical/constrained, else four MB
  - For ten Gigabit speed OSA-Express or HiperSockets
    - Four MB if CSM critical/constrained, else six MB
- If more data arrives than the current limit allows, packets are discarded

```
IST2273E PACKETS DISCARDED FOR jobname - READ QUEUE CONGESTION
```

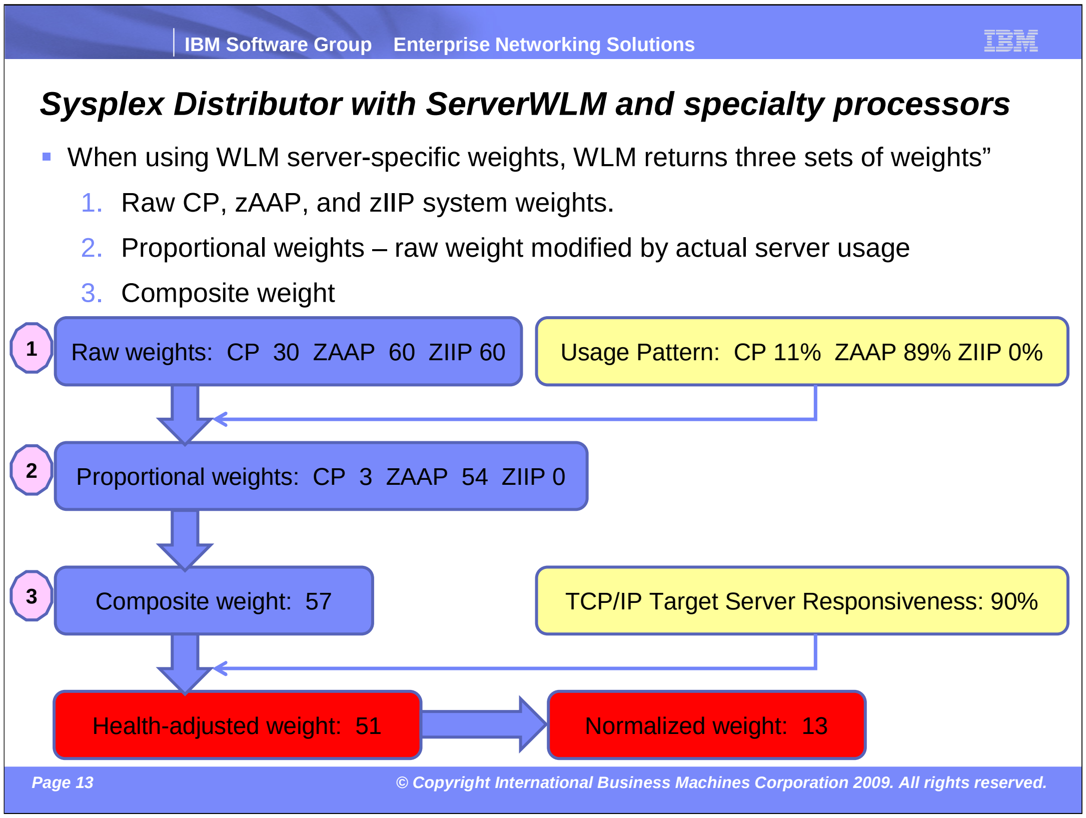
Page 12

Although multi-processing of inbound QDIO is necessary, allowing an unchecked number of threads to run is unacceptable. z/OS V1R11 implements an interface-speed dependent limit on the number of parallel SRBs that can be active processing inbound data for a QDIO interface. This limit is always enforced even when the system is not in a storage-constrained situation.

The staging queue contains a mixture of ECSA and CSM data space. Just before adding to the staging queue, the PCI exit sees if the queue already contains an excessive number of ‘containers’. Each container represent up to 64K of CSM data space containing inbound packets. If the staging queue has an excessive number of elements, Communications Server discards the containers instead of queuing. Containers are in ECSA so you can apply limits to ‘container’ consumption.

Discarding IP packets is an acceptable behavior in an IP environment when a node becomes congested. This is frequently done by routers.

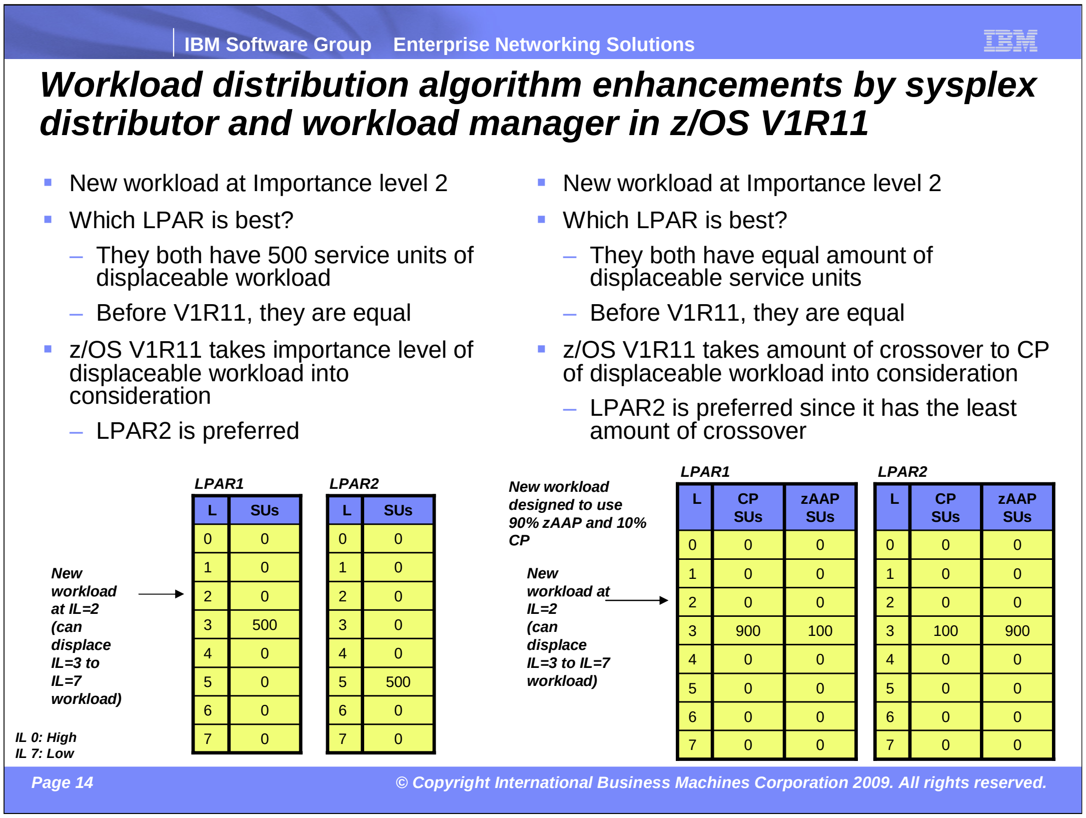
Message IST2305I is issued as part of the message group response to Display NET,TRL,TRLE=*trlename* and Display NET,ID=*trlename*. The *sbalcnt* indicates the number of containers discarded since TRLE activation. Since each container represents a 64K read buffer, this is also the count of the number of read buffers discarded.



When using WLM server-specific weights, WLM returns raw CP, zAAP , and zIIP weights. These weights represent a comparison of displaceable capacity for each processor type given the importance level and goals of the target servers’ service class. WLM returns proportional weights based on the actual usage pattern of the targeted servers for each processor type. WLM returns a composite weight determined from the proportional weights.

This example shows how the weights are determined for ServerWLM. The Raw weights, proportional weights, and composite weight are all calculated by WLM. Proportional weight is based on the current usage by the application of this processor. The composite weight is derived from the proportional weights.

The sysplex distributor will modify the composite weight by the Target Server Responsiveness fraction (how healthy is the server?) and then determine a normalized weight by dividing by 4.



WLM supports a new method to calculate relative weights of the target servers, to which Sysplex Distributor distributes incoming connections.

The new method provides two enhancements.

The first enhancement is for WLM to consider the relative importance of existing work that is displaced on each target as new work is distributed. Target systems with more lower importance displaceable work are favored over systems with higher importance displaceable work.

The second enhancement is to factor in the use of specialty processors (zIIP and zAAP) and the amount of cross-over that occurs. If two target LPARs has the same amount of zIIP and zAAP processor capacity, WLM can be requested to prefer the target that has the least amount of cross-over. Cross-over means zIIP/zAAP eligible workload that is processed on a general CP.

These enhancements apply to the SERVERWLM distribution method.

This support is based on the new METHOD=EQUICPU function in the IWM4SRSC WLM service.

Importance level changes in z/OS V1R11

- Importance level weighting factor of zero (IL0) means no change as compared to pre-R11 behavior
- Importance level weighting factors of one through three (IL1 through IL3)
  - gradually shifts new workloads towards LPARs with the lowest importance level work to displace
- Start with a moderate (IL 1) value initially
- Run the RMF workload activity report before and after a change

WLM categorizes work into eight importance levels with level zero being the most important and level seven being the least important (level seven contains unused or available services units).

Server-specific weight recommendations are based on a comparison of displaceable capacity given the importance level of the new work's service class. The importance level of the work being displaced is not considered.

WLM will support an importance level weighting factor which is used when comparing displaceable service units in each LPAR.

Four importance level (IL) weighting factors are supported:

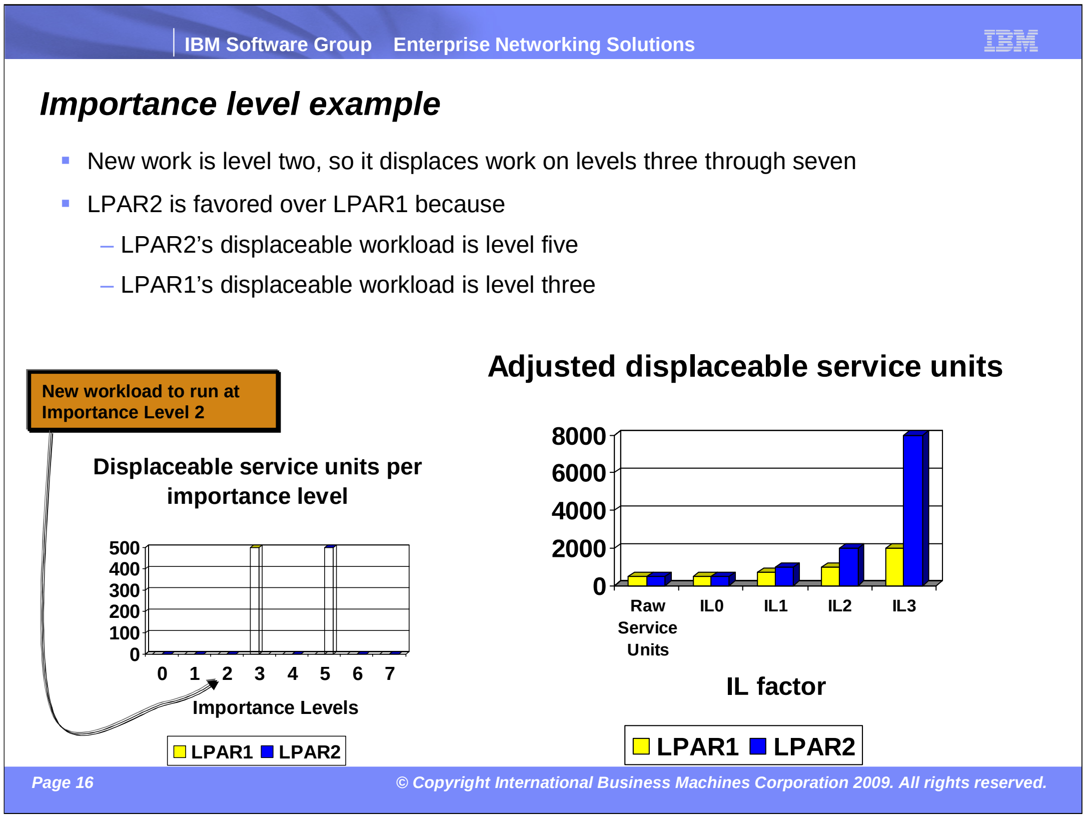
IL 0 – Ignore importance levels, this is the existing and default behavior.

IL 1 – Moderate, adjust the service units by the square root of the importance level difference of the New work and the Displaceable Service Units + 1.

IL 2 – Aggressive, adjust the service units by the importance level difference of the New work and the Displaceable Service Units + 1.

IL 3 – Exceptionally Aggressive, adjust the service units by the square of the importance level difference of the New work and the Displaceable Service Units + 1.

Using an IL value of two or greater will cause systems with comparatively higher importance displaceable workloads to receive a significantly lower percentage of the new workload. This can have an adverse effect on performance. As a guideline, use a Moderate (IL 1) value initially. Run the RMF Workload Activity Report before and after a change to understand how this affects performance.



In the example, since the new work is importance level two, it can displace lower importance work loads on levels three through seven. LPAR1 and LPAR2 each have the same server-specific weight since each system has 500 Service units of displaceable work. But LPAR2 should be favored over LPAR1 since LPAR2's displaceable workload is lower importance (level five) than LPAR1's workload (level three).

In the example, using a weighting factor of:

IL 0 Ignore - LPAR1 and LPAR 2 adjusted service units are unchanged.

IL 1 Moderate - Displaced work SUs * SQUAREROOT[(Displaced work IL) – (New work IL) + 1].

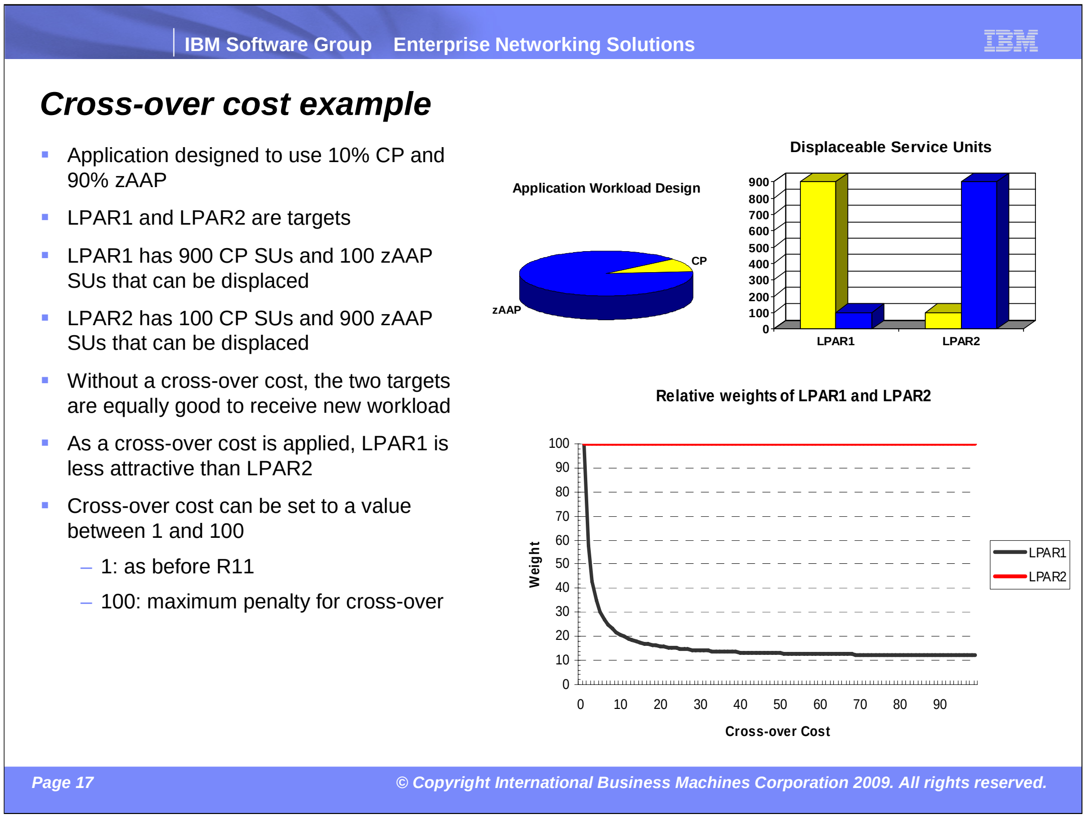
LPAR1 adjusted service units - 707 = 500 * SQUAREROOT[(3) – (2) + 1].

IL 2 Aggressive - Displaced work SUs * [(Displaced work IL) – (New work IL) + 1].

LPAR1 adjusted service units - 1000 = 500 * [(3) – (2) + 1].

IL 3 Exceptionally Aggressive - Displaced work SUs * SQUARE[(Displaced work IL) – (New work IL) + 1].

LPAR1 adjusted service units - 2000 = 500 * SQUARE[(3) – (2) + 1].



When server-specific distribution is used, a crossover cost can be configured. The crossover cost is used to penalize service units (SUs) that run on the CP instead of the specialty processor. The crossover cost range is between one (no crossover penalty) and 100 (heavy crossover penalty).

A JAVA workload is designed to consume CP SUs 10% and zAAP SUs 90%. The JAVA workload runs at importance level two; in the example both LPARs have the same displaceable capacity available, 1000 SUs.

For LPAR2, as 100 CP SUs run, the designed percentage of 900 zAAP SUs run, so there is no crossover cost needed.

For LPAR1, as 111 SUs are consumed, the designed percentage of 11 CP SUs (10% of 111) and the designed percentage of 100 zAAP SUs (90% of 111) are consumed. There is no penalty imposed on the 11 CP SUs. As the remaining 889 CP SUs run, 89 “pure”, non-crossover, CP SUs are consumed as 800 crossover CP SUs are consumed.

With a crossover cost of one, no penalty is applied against the 89 SUs. So the equivalent CP cost is 100 = 11 + 89.

With a crossover cost of five, the 89 CP SUs are divided by (percentage of CP intended usage) + (crossover cost * percentage of zAPP intended usage). So in this case, 19 CP = 89/((10%) + (5 * 90%)). Result is rounded to 1. The resulting *equivalent* CP weight is 30 = 11 CP SUs + 19 CP SU.

Configuring and displaying the new SERVERWLM options

- The new configuration parameters are
  - Only valid when server-specific recommendations are being used
  - Only used by WLM when all systems in the sysplex are V1R11 or later

- These parameters can affect performance
  - Importance Level values range from one (no impact) to three (aggressive weighting).
    - Use Moderate (IL one) value initially
  - Crossover cost values range from one (no impact) to 100 (crossover cost very expensive).
    - Use a low cost initially

```
VIPADISTRIBUTE
      DISTMETHOD SERVERWLM  PROCXCOST ZIIP 5 ZAAP 20 ILWEIGHTING 1
      201.2.10.11  PORT 8000
      DESTIP  ALL
```

```
NETSTAT VIPADCFG DETAIL
VIPA Distribute:
  Dest:         201.2.10.11..8000
    DestXCF:    ALL
      SysPt:    No  TimAff: No  Flg: ServerWLM
      OptLoc:   No
      ProcXCost:
        zAAP: 020  zIIP: 005
      ILWeighting: 1
```

This slide shows an example of how to use and display the processor cost and importance level weighting parameters. In the example, a procxcost zIIP value of five has been configured as the cost to penalize CP SUs consumed when zIIP SUs run on the CP, and zAAP cost of 20 (higher penalty).

Additionally, an ILWEIGHTING value of one has been configured. This will provide a moderate bias when comparing displaceable capacity at different importance levels.

The new configuration parameters can only configured with server-specific (DISTMETHOD SERVERWLM) distribution. WLM will only use the parameters if all systems in the sysplex are V1R11 or later.

These parameters can affect performance. As a guideline, use a moderate importance Level weighting (IL one) initially and a low crossover cost. Run the RMF Workload Activity Report before and after making a configuration change to one of these parameters.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

HiperSockets   VTAM   z/OS

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance 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 or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.