#SHAREorg

SHARE
Technology · Connections · Results

Diagnosing Sysplex Problems

Mark A Brooks
IBM

Wednesday, February 6, 2013: 3:00 PM-4:00 PM
Session Number 13402

mabrook@us.ibm.com

SHARE in San Francisco 2013

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM®	MQSeries®	S/390®	z9®	IBM® (logo)
ibm.com®	MVS™	Service Request Manager®	z10™	AIX® BladeCenter®
CICS®	OS/390®	Sysplex Timer®	z/Architecture®	DataPower®
CICSPlex®	Parallel Sysplex®	System z®	zEnterprise™	DS4000®
DB2®	Processor Resource/Systems Manager™	System z9®	z/OS®	DS6000™
eServer™	PR/SM™	System z10®	z/VM®	DS8000®
ESCON®	RACF®	System/390®	z/VSE®	POWER7®
FICON®	Redbooks®	Tivoli®	zSeries®	ProtecTIER®
GDPS®	Resource Measurement Facility™	VTAM®	zEC12™	Rational®
HyperSwap®	RETAIN®	WebSphere®	Flash Express®	System Storage®
IMS™	GDPS®			System x®
IMS/ESA®	Geographically Dispersed Parallel Sysplex™			XIV®

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license there from.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

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.

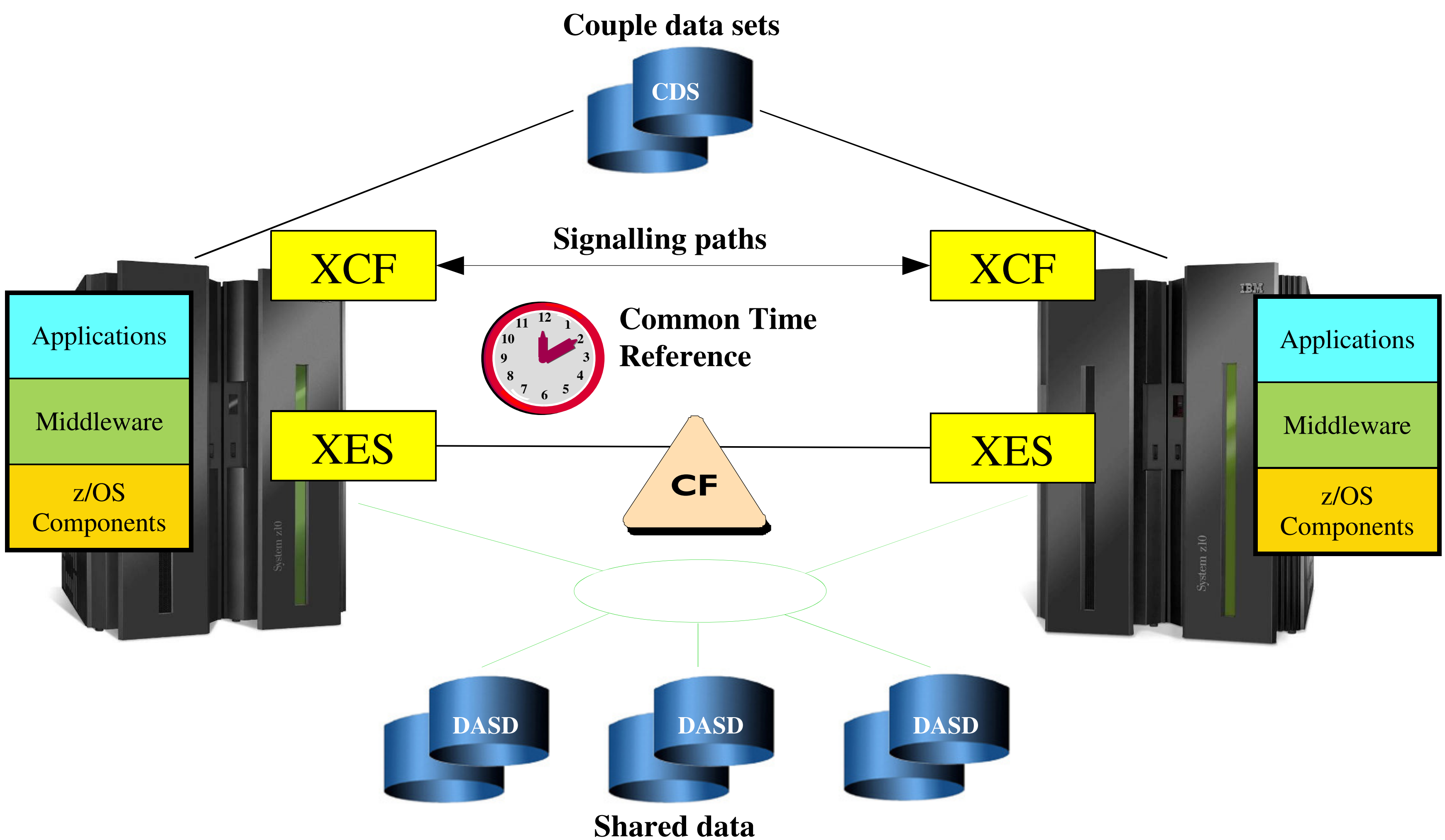
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Downloads available from SHARE website

- Presented Slides
  - Material to be presented during the session
  - "See handout" is a reference to slides in the handout that will not be presented

- Handout Slides
  - All of the above plus additional slides that will not be presented
    - Has a more complete treatment of key concepts
    - Includes examples of relevant messages, display output, and reports to illustrate what you would be looking at when diagnosing problems
  - If following along with the handout during the presentation, you can use the slide numbers in the lower left corner to keep pace (there will be lots of skipped slides)

Sysplex environment



Context

- In the sysplex, we have instances of z/OS, middleware, and applications running on individual systems
- These individual instances cooperate with one another to perform some function
- A given sysplex function may rely on the services of other sysplex functions, sometimes in rather non-obvious, even circular ways

Sysplex Problems

- Sympathy sickness (hangs)
  - If an instance of a sysplex function is unresponsive, it may:
    - Hold serialization on shared resource
    - Stop sending replies
  - Other instances may hang as a result
- Performance issues (delays)
  - Instance is responsive, but “slow”
  - May impact response time of individual requests, which at high request rates, can lead to significant throughput problems and/or large queue effects that also give the appearance of hangs

My thesis

- We do not have the expertise needed to do sysplex diagnosis
- We do not know:
  - Implementations, interactions, and dependencies
  - Causes, relationships, or relevance of various symptoms
  - How a given problem might impact the sysplex
  - All potential causes of a given symptom
- But we do know:
  - Sysplex application instances interact with one another
  - Most exploit sysplex services to do so
  - These services and their usage can be observed and analyzed
- So despite our imperfect knowledge, there is hope
  - We can make sure the sysplex infrastructure is sound
  - If applications can readily communicate and share data, we likely have a problem that will yield to traditional single system diagnosis
  - Hopefully our analysis will point us towards the vicinity of the culprit

Problem Taxonomy

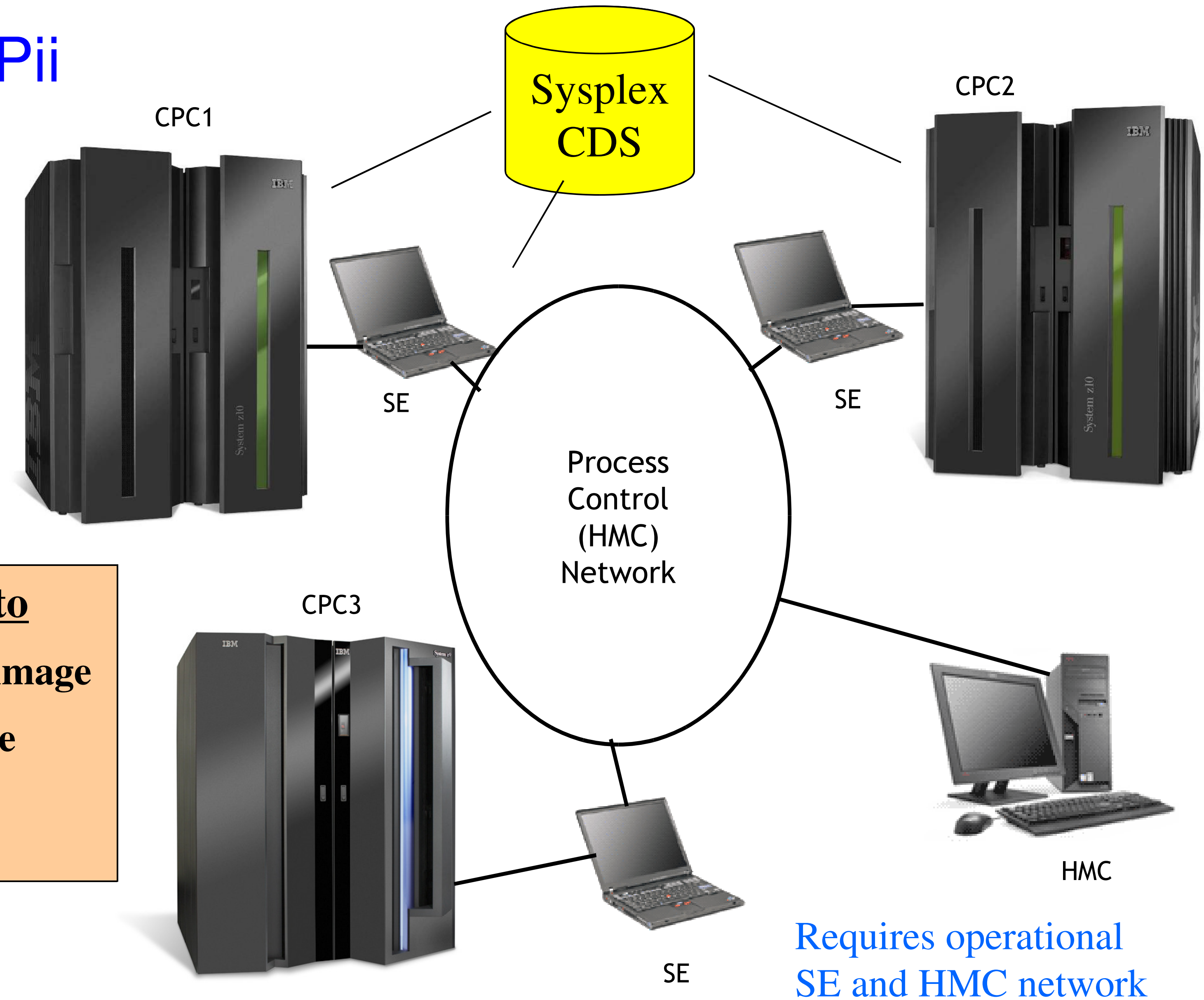
- **Dead System**
- Sick System
- Sysplex Fabric
- Sysplex Componentry
  - Coupling Facility
  - Signalling Service
  - Couple Data Sets
  - External Time Reference
- Configuration / Capacity
- Software Issues

Dead System

- A dead system can't participate in anything
  - Dead System = wait state and not removed from sysplex
  - The root cause of sympathy sickness often turns out to be a dead system that no one noticed !
- **So this should always be the first thing you check**
- Symptoms
  – IXC402D "system looks dead"
  – IXC102A "tell me when system is reset"
  – Other messages (see handout)

SFM With BCPii

z/OS Images
(not VM guests)



XCF uses BCPii to

- **Obtain identity of an image**
- **Query status of remote CPC and image**
- **Reset an image**

z/OS 1.11 SFM with BCPii

- SFM will automatically exploit BCPii and as soon as the required configuration is established:
  - Pairs of systems running z/OS 1.11 or later
  - BCPii configured, installed, and available
  - XCF has security authorization to access BCPii defined FACILITY class resources
  - z10 GA2 with appropriate MCL's, or z196, or z114, or zEC12
  - **New version of sysplex CDS is primary in sysplex**
    - Toleration APAR OA26037 for z/OS 1.9 and 1.10
    - Does NOT allow systems to use new SSD function or protocols

If you have the appropriate environment, SET THIS UP !
You will likely eliminate this entire class of "dead system" issues

Problem Taxonomy

- Dead System
- **Sick System**
- Sysplex Fabric
- Sysplex Componentry
  - Coupling Facility
  - Signalling Service
  - Couple Data Sets
  - External Time Reference
- Configuration / Capacity
- Software Issues

Sick System: Typical Root Causes

- Storage constraints
  - Not enough resource for system to run reasonably
  - Run away application
  - Defect
- CPU constraints
  - Not enough resource for system to run reasonably
  - SRBs looping
- Contention
  - ENQ, latches, local lock, spin locks
- DASD I/O issues
- Sick components

Run Time Diagnostics (RTD)

- Reviews critical messages in the log
- Analyzes contention
  - GRS ENQ
  - GRS Latches
  - z/OS UNIX file system latches
- Examines address spaces with high CPU usage
- Looks for an address space that might be in a loop
- Evaluates local lock conditions
- Performs additional analysis based on what is found
  - For example, if XES reports a connector as unresponsive, RTD investigates the appropriate address space

RTD can find many sick system issues. Give it a try.

Sick System: Storage constraints

- System may not run well if storage constrained
- Shortages tend to induce paging
  - Resolving page faults introduces delay and system overhead
  - Swap out of address spaces implies programs are not running
- Typical Causes
  - You need more memory
  - Runaway application
  - Page pack performance (see "Sick System: DASD I/O issues")
  - Defects

Sick System: CPU Constraints

- "Something" will not be running
- Usually the "something" is deemed to be less important, so it may seem that you are getting the desired result
  - Test system
  - Discretionary work
- But failure to run the less important "something" can induce sympathy sickness for the important work if it:
  - Holds serialization for which there is contention
  - Holds resources, or worse, accumulates them while not running
    - For example, XCF Signal buffers
  - Not participating in cooperative processes

CPU Constraint Considerations

- Latent demand?
  - If LPAR busy less than MVS busy, the physical processor is being ripped away despite the fact that MVS has work to do
- Blocked work?
  - If ready work is only getting “trickles”, the box is pretty much saturated
- Running high importance interactive/transaction oriented workloads at more than 90-92% busy is asking for trouble

- Did something change?
  - Configuration (upgrade, number CPs, capping, CF, ...)
  - Workload
  - CPU utilization by some job/space

Sick System: Contention

- ENQ contention
  - Often arises as the result of submitting some new batch job that serializes same resource as existing work/job, or
  - Submitting multiple batch jobs that need to use the same data sets
- Latch contention
  - Often arises as the result of work load changes
  - USS, RRS, Logger are examples of users of latches
  - Latches are local but can induce sympathy sickness:
    - Get latch, send message, <delay?>, get response, release latch
    - <delay?> Get latch, formulate response, send response, release latch
  - Can give rise to a tangled web of sympathy sickness which can be quite difficult to diagnose (stay focused on the holder)

Sick System: Contention ...

- Local Lock
  - Some applications use local lock for serialization
  - Many system services often need it as well
    - GETMAIN, ATTACH, …
  - Is holder running? enough?
  - Long queue effects?
- Spin Locks
  - If RSM lock, could be sign of real storage shortage
  - Hardware errors?
  - Long queue effects?
  - Software defect?

Contention considerations

- Contention is often induced by other problems
- Experience suggests:
  - Changing mix of batch jobs often induces ENQ contention
  - Spinning on RSM locks often implies real storage shortages
  - Workload changes often induce competitive contention
  - Lack of dispatch time can induce persistent contention
  - USS latch contention often arises due to issues on peer system
- So at this point in our methodology, if root cause is not obvious
  - *As it might be for batch jobs or real storage shortages*
- Take note of the contention, but continue diagnosis
  - *Who holds the resource? Why aren't they making progress?*

Sick System: DASD I/O issues

- Performance, response times, throughput, even functionality may be impacted if applications encounter errors or delays while accessing data on DASD
- Experience suggests that potential for such delays exists with:
  - Synchronous mirroring
  - Slow DASD
  - Workload changes
    - Changes in request rates, device contention, ...
- Of particular concern for sysplex are impacts to
  - Couple Data Sets
  - Data sets used by Logger
  - Page packs

Sick System: Sick Components

- Experience suggests that the system will not be running well if the following components are having issues (no particular order)
  - Logger
  - RACF
  - JES3
  - JES2
  - RRS
  - Unix System Services
  - Consoles
  - GRS
  - SMF

Many of these critical components exploit sysplex services. So bear in mind that they could be sick due to sysplex issues we have not yet covered.

My idealized methodology is trying focus on internal problems that impact the component independently of the sysplex infrastructure. In the real world we may not be able to achieve such isolation.

- Certainly others, but these tend to be most prevalent with respect to having single system issues that impact the sysplex

Problem Taxonomy

- Dead System
- Sick System
- **Sysplex Fabric**
- Sysplex Componentry
  - Coupling Facility
  - Signalling Service
  - Couple Data Sets
  - External Time Reference
- Configuration / Capacity
- Software Issues

Sysplex Fabric

- Consists of the various cables, links, channel paths, CHPIDs, and adapter cards that provide physical access to the sysplex componentry
- Along with the parmlib members and policies that govern logical access to the sysplex componentry
- Performance, response time, throughput, even functionality may be impacted if access to the sysplex componentry is impeded by
  - Error prone connections
  - Loss of connectivity
  - Outright loss or lack of access
- So now we make sure that the expected sysplex componentry exists and is accessible to each system in the sysplex

Sysplex Fabric: Couple Data Sets

- Physical access
  - See: “Sick System: DASD I/O issues”
  - IODF
- Logical access
  - COUPLExx COUPLE statement for sysplex CDS
  - COUPLExx DATA statements for function CDS's
  - SETXCF COUPLE command
- Detecting Fabric Problems
  - D XCF,COUPLE,TYPE=ALL (see handout)
  - XCF Messages (see handout)
  - IOS messages (see handout)
  - Exploiter Messages (see handout)

Do the various CDS exist?
Have an alternate?
Accessible to all systems?

Using right COUPLExx?
What changed?

Sysplex Fabric: Coupling Facility

- Physical Access
  - CF
  - Links

Does the CF exist?
Accessible from all systems?
Using the right CFRM policy?

- Logical Access
  - CFRM Policy

- Detecting Fabric Issues
  - D CF — physically connected?
  - D XCF,CF — logically connected?
  - D XCF,POLICY,TYPE=CFRM — right policy?
  - XCF Messages (see handout)
  - IXLERLOG LOGREC entries for link issues (see handout)
  - Check for flashing icons on the HMC — hardware issues?
  - HMC command to display CF — CF exist? Does it respond?

Sysplex Fabric: CF Structures

- Physical Access
  - Do expected structures exist?
- Logical Access
  - CFRM Policy
  - IXLCONN
- Detecting Fabric Issues
  - D XCF,CF
  - D XCF,POLICY,TYPE=CFRM
  - RMF CF Activity Report
  - XCF/XES Messages (see handout)

Which structures do you normally run with?
Do they exist?

Running with expected CFRM policy?
Any failed structure allocations?

Sysplex Fabric: Signalling Paths

Are CF structures used for signalling accessible?

Are CTC devices used for signalling online and operational?

- Physical Access
  - See “Sysplex Fabric: CF Structures”
  - CTC devices
- Logical Access
  - COUPLExx PATHOUT statements, or SETXCF START,PATHOUT
  - COUPLExx PATHIN statements, or SETXCF START,PATHIN
- Detecting Fabric Issues
  - D XCF,PO
  - D XCF,PI
  - Messages (see handout)
  - Apply “Sysplex Fabric: CF Structures” to signal structures
  - Check for IOS messages related to signal path CTC devices

Sysplex Fabric: External Time Reference

- Physical Access
  - Coordinated Time Network
  - Timing links
- Logical Access
  - CTN ID
- Detecting Fabric Issues
  - D XCF,SYSPLEX,ALL
  - D ETR,DATA
  - Messages (see handout)
  - HMC

If z/OS image loses access to ETR, the system is in a wait-state.

So my “fabric detection” is either
- Proactive prevention
- Post mortem analysis

If lose ETR, “live” investigation is via the HMC and other systems that still have ETR access

Timer links operational?
Is the CTN ID correct?
PTS operational? BTS? Arbiter?
Is CTS the one you want?

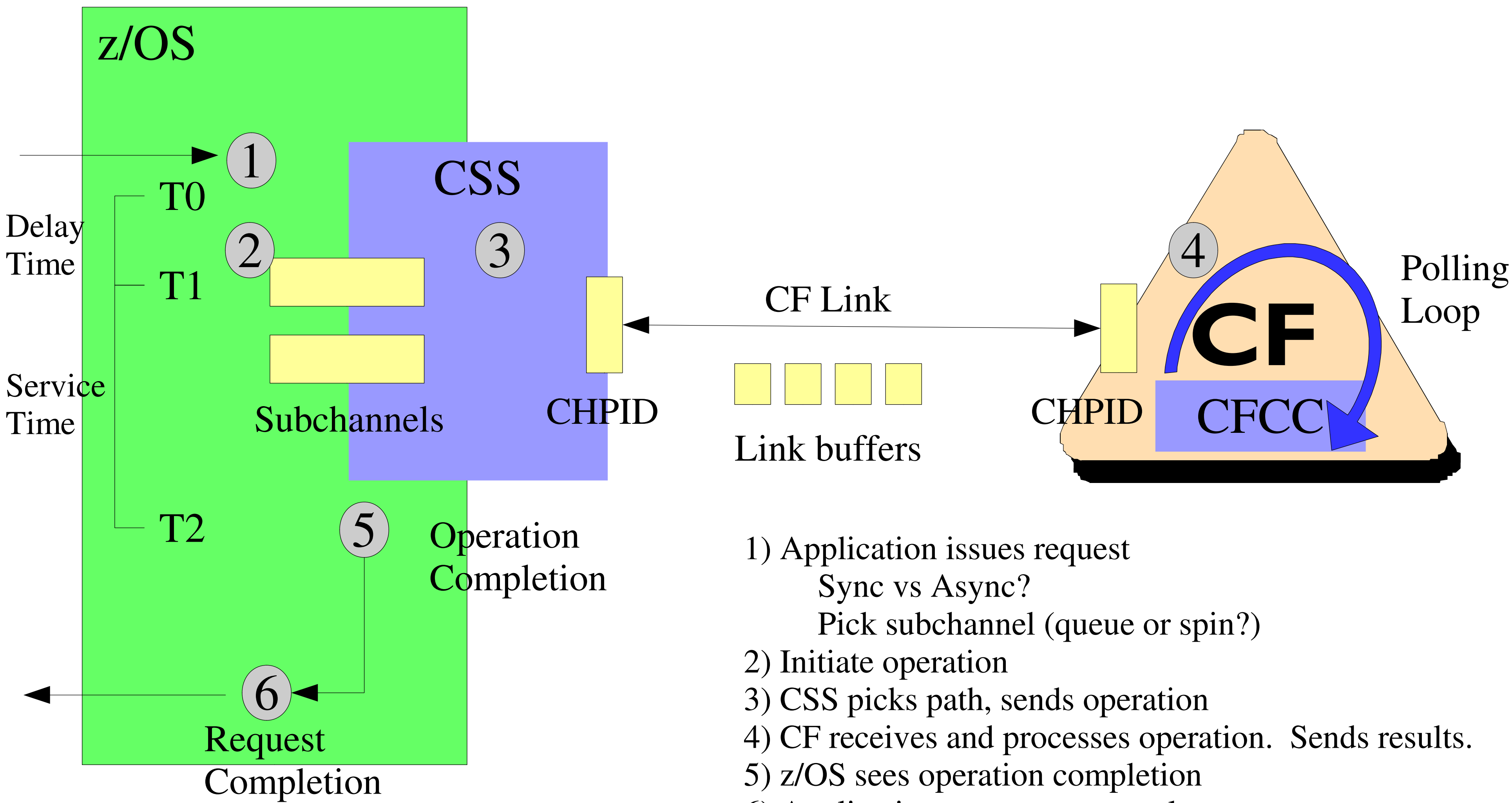
Problem Taxonomy

- Dead System
- Sick System
- Sysplex Fabric
- Sysplex Componentry
  - **Coupling Facility**
  - Signalling Service
  - Couple Data Sets
  - External Time Reference
- Configuration / Capacity
- Software Issues

Methodology Recap

- At this point, we should have the following conditions:
  - All systems are operational
  - All sysplex componentry is accessible via error free connections
- We suspect there may be issues with sysplex componentry
  - Performance issues
  - Configuration or capacity issues
- So we need a detailed understanding of the how the sysplex componentry works so that we can determine
  - How such problems might arise
  - How they might be observed
  - How they might be resolved

Coupling Facility Request Processing



1) Application issues request
    Sync vs Async?
    Pick subchannel (queue or spin?)
2) Initiate operation
3) CSS picks path, sends operation
4) CF receives and processes operation. Sends results.
5) z/OS sees operation completion
6) Application gets request results

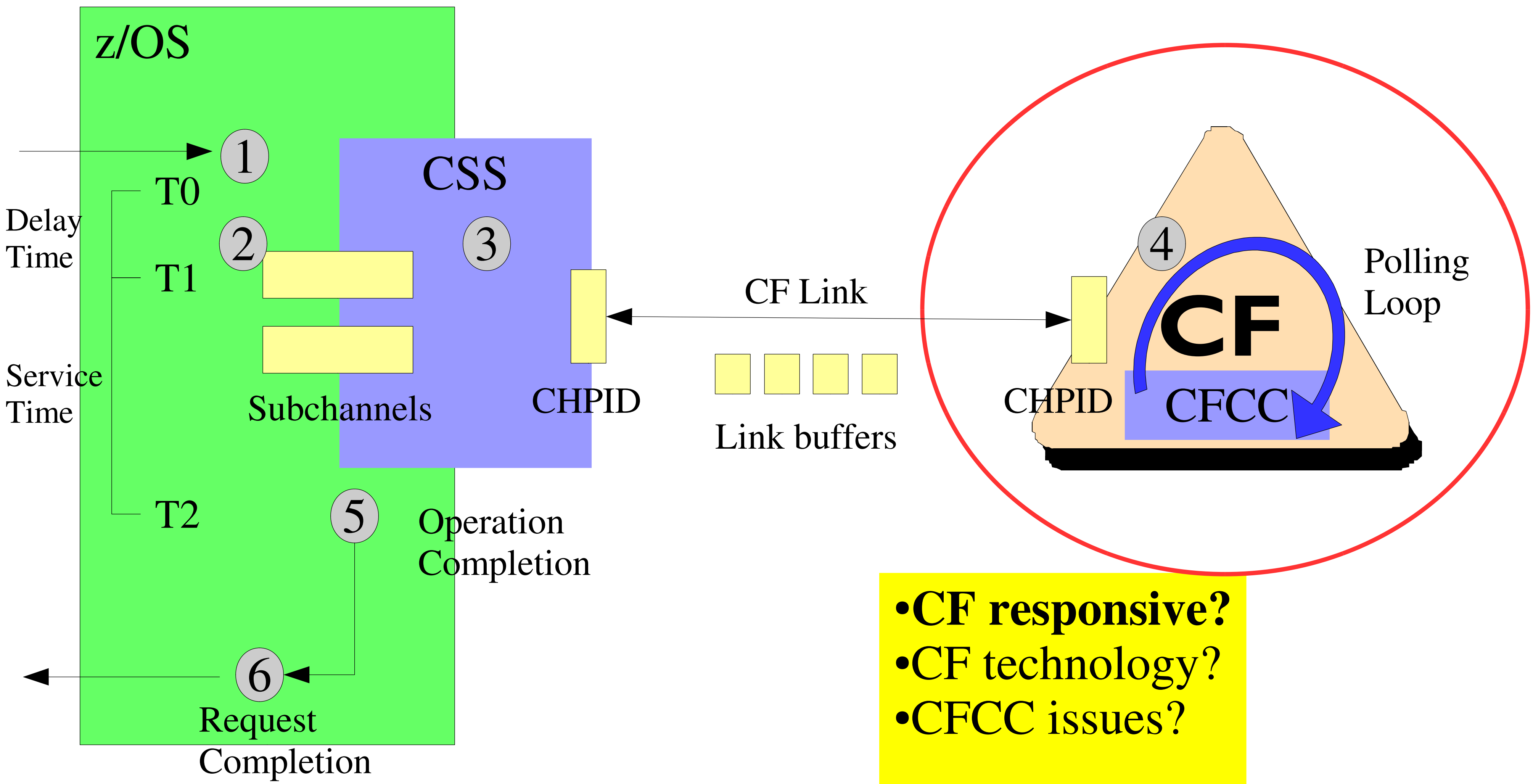
CF Service Time Considerations

- Delay Time is time spent waiting for a subchannel
- Service Time is delta between sending operation and observing its completion
- Service time impacted by
  - Coupling Facility
    - Technology
    - Utilization
    - Contention
  - CF Links
    - Technology
    - Path busy conditions
    - Distance
  - Completion recognition

You get one number that encompasses all these factors.

One or more could be the issue.

Coupling Facility Request Processing



- **CF responsive?**
- CF technology?
- CFCC issues?

CF Responsive?

- LPAR configuration must allow the CF to be sufficiently responsive so that the polling loop can run to receive commands in a timely manner
  - Dedicated CPs recommended
  - Shared CPs need sufficient weight
    - >50% for simplex or user managed duplexing
    - >95% for system managed duplexing
  - Dynamic Dispatch can give rise to erratic, elongated response times
  - Sharing CPs between z/OS and CF can impact response time

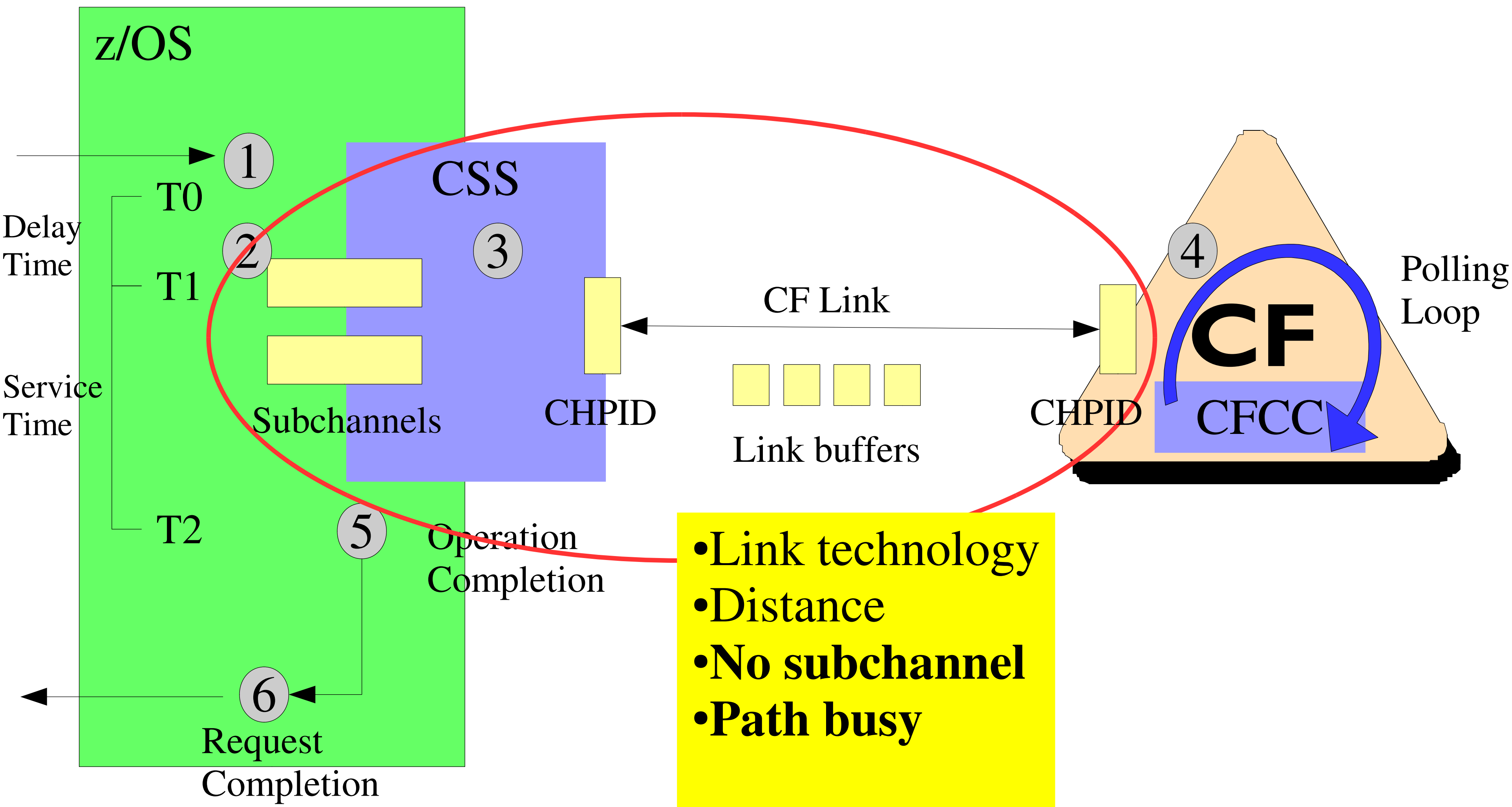
- CF utilization guidelines
  - <30% busy if single CP, otherwise <50%
  - Ensure sufficient capacity to handle structures that might be rebuilt into the CF as the result of failures or maintenance on a peer CF

Detecting Coupling Facility Issues

- First resolve existence and accessibility issues
  - See “Sysplex Fabric: Coupling Facility”
- Performance
  - RMF CF Activity Report
  - RMF Partition Data Report
  - RMF Monitor III – CF Overview
  - RMF Monitor III – Sysplex CF Views

Getting enough physical CP?
Timely dispatch?
CF utilization within guidelines?

Coupling Facility Request Processing



- Link technology
- Distance
- **No subchannel**
- **Path busy**

CF Link Considerations

- Link technology and distance
  - Faster links improve transmit portion of service time
  - Distance increases service time by 10 mics/km
- No subchannel conditions
  - Bursts? Sustained load? Tuned due to path busy?
- Path busy conditions
  - Number of requests exceeds number of available link buffers
  - XES may tune number of subchannels to avoid this condition
  - Distance (link buffers in use for longer)
- Configured correct number of subchannels/CHPID?
  - 32 for HCA2-O LR or HCA3-O LR, otherwise 7

Detecting CF link issues

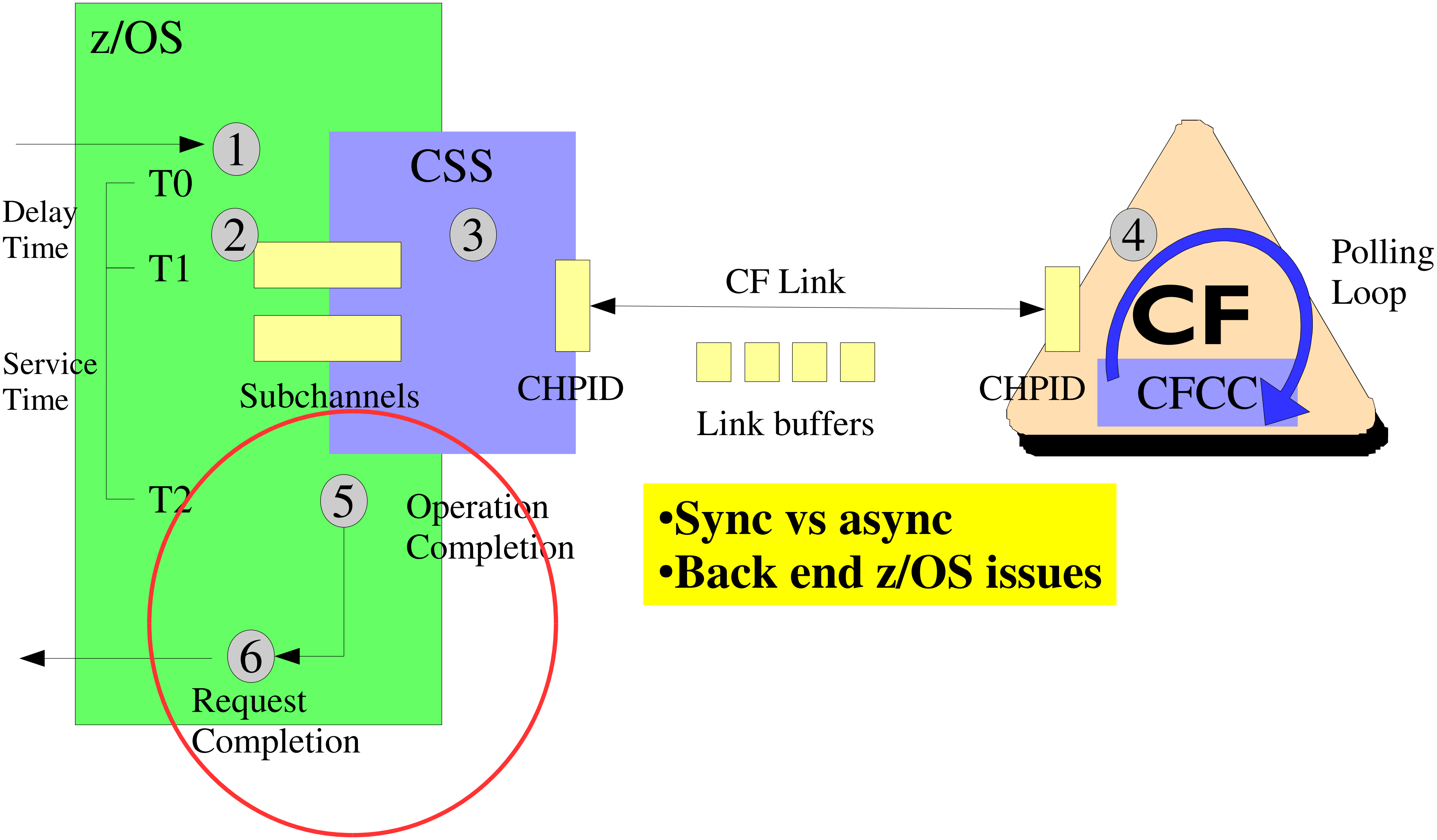
- First eliminate all physical link errors
- First resolve or eliminate CF responsiveness issues
  - Unresponsive CF can induce link problems
  - Link buffers “linger”, which can induce path busy conditions
- RMF Report of CF Activity
  - Delayed requests implies “no subchannel”
  - Reports “path busy” conditions
- D CF,CFNAME
  - Shows configured links
  - How many subchannels available? Being used?

Methodology Concern

- Subchannel and path busy conditions imply that there are not enough subchannels and link buffers to satisfy the workload
- Typically resolved by
  - Adding links, or CHPIDs (for IFB)
  - Upgrading link technology
- But …
  - Has workload grown?
  - Experiencing a spike? Just tolerate it?
  - Runaway application?

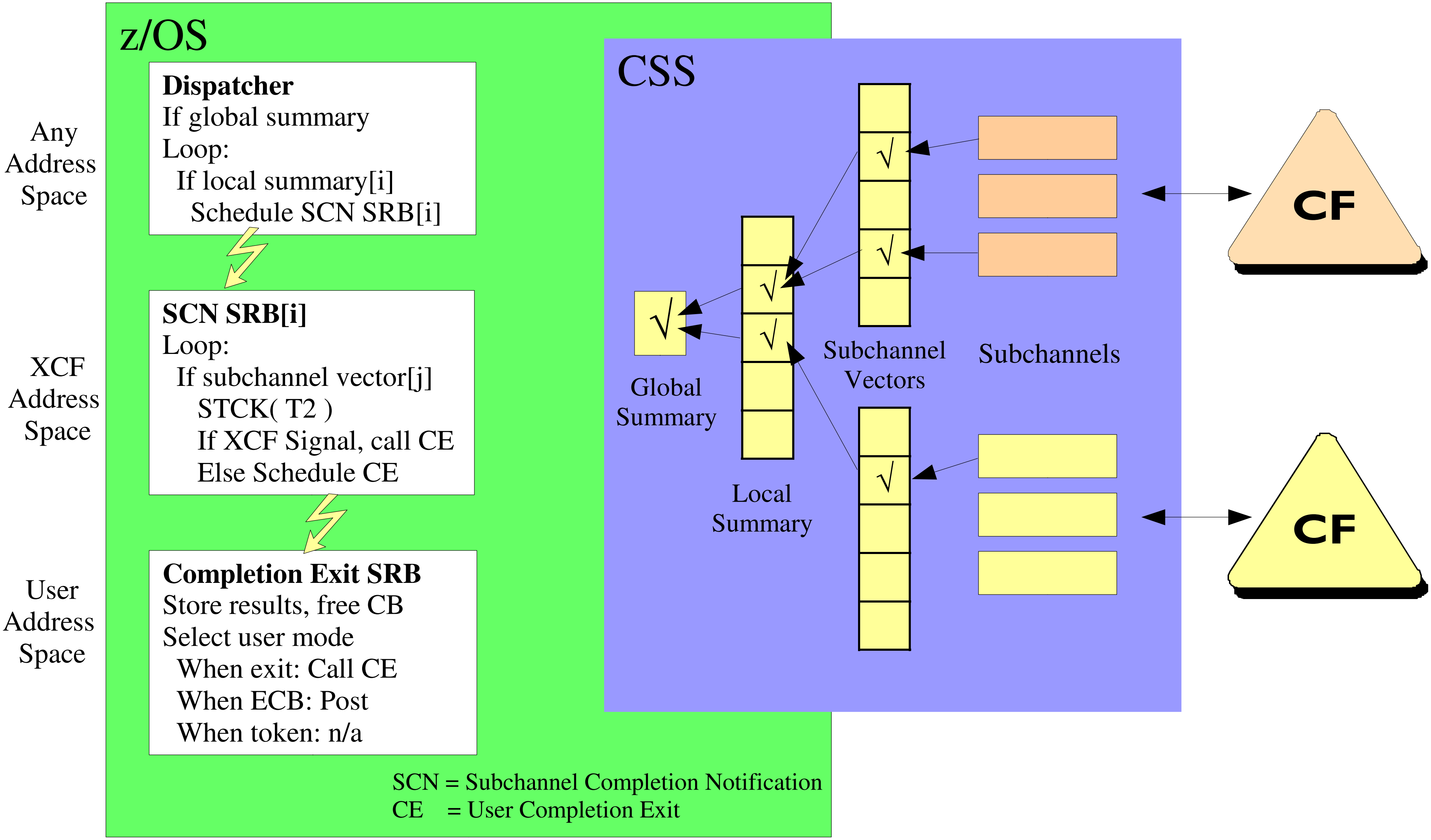
Know your workload.
What is normal for you.
What changed?

Coupling Facility Request Processing



- **Sync vs async**
- **Back end z/OS issues**

Asynchronous operation completion



Request Completion Issues

- Mismanagement of summary bits and subchannel vectors
  - (Extremely rare, only seen this once. I mention to be complete.)
  - Subchannel completion vector issues are likely “repaired” by synchronous requests, or an internal monitor
- Loss of physical processor
  - May elongate asynchronous CF service times
  - May elongate application response times
- Low loads
  - MVS may ask LPAR to take it out of its no work wait less often
  - If dispatcher runs less often, takes longer to notice completion
  - Which increases async service times

Request Completion Issues ...

z/OS Dispatching Issues

- Is application address space getting dispatched enough?
  - Won't show up in CF service time measurements
  - But application might appear to be sluggish, and
  - Increases dwell time of XES control blocks (common storage)
- XCF runs at high dispatch priority, so not usually an issue
  - Unless there is a more global issue
    - Storage constraints, spin loops, …
    - But we already eliminated those issues, right?

CF Request Response Time Summary
What does your "one number" tell you?

- Time z/OS spends waiting for subchannels

Delay time

Service time

- Time spent resolving path busy conditions
- Time spent transmitting request from z/OS to CF

Local

- Time spent waiting for physical dispatch of CF to receive request
- Time CF spends preparing, processing, and completing the request

Global

- Time spent transmitting results from CF to z/OS
- Time spent waiting for physical dispatch of z/OS to receive results

Local

Service time

Back End time

- Time spent in back end application completion processing

CF Structure Specific Concerns

- Properly sized?
  - Always resize after CF/hardware upgrade
  - Resize as workload changes
  - Is CFRM policy change still pending for new size?
  - For lock structures: Contention? False contention?
- Hitting full thresholds?
  - Whether an issue depends on application
- Being altered?
- Being rebuilt?
- Newly allocated?
- Application design issues
  - Perhaps the implementation induces contention/queueing in CF

See "z/OS Hot Topics" Issue 26 Lead article on CF Sizer

Sizing issues often arise long after CF upgrade for which structure was not resized

Contention implies delay, and induces signalling activity. Can fix false contention by increasing structure size appropriately.

Activity is quiesced (delayed) while in rebuild

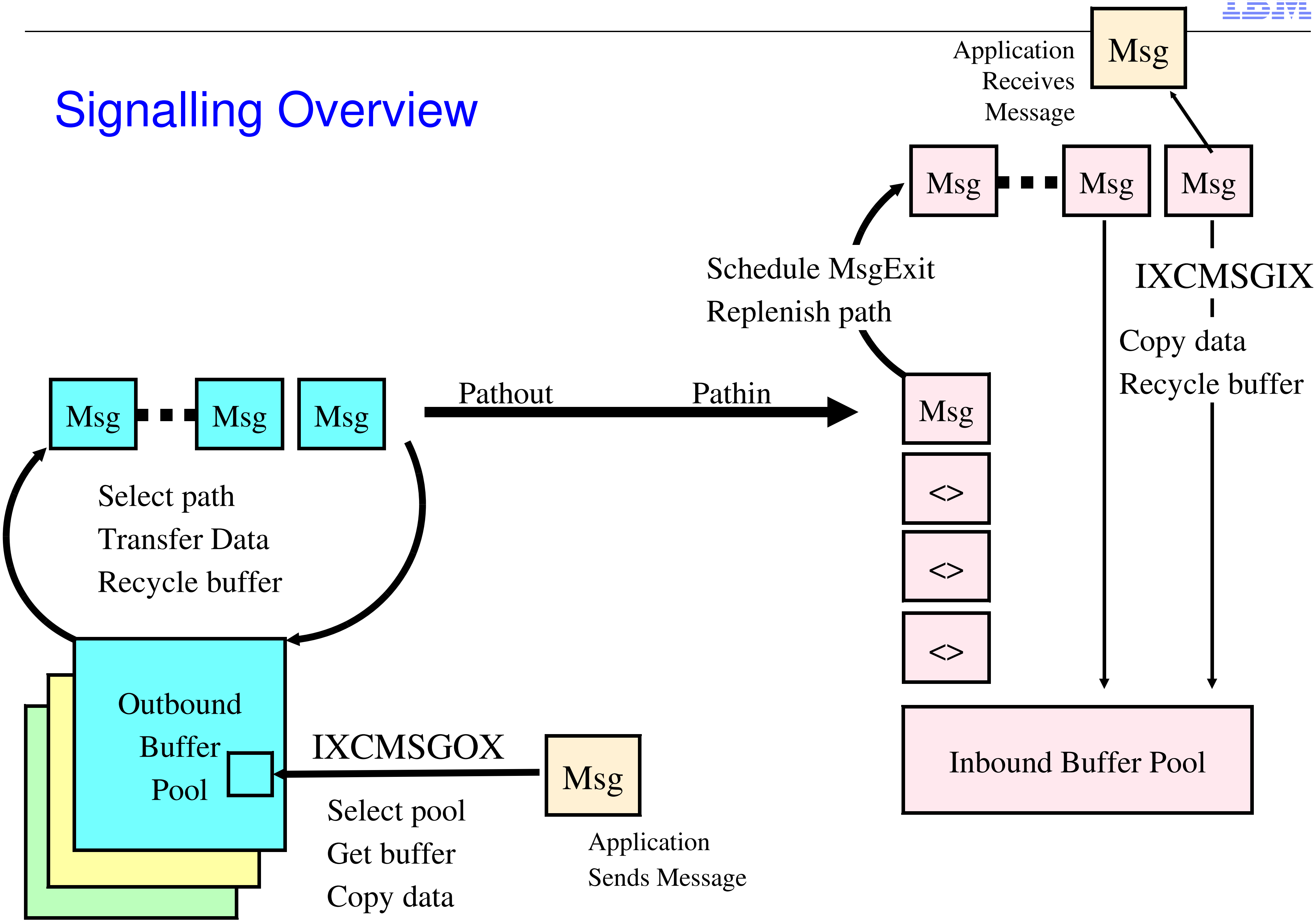
CF Structure Specific Diagnosis
Be Prepared

- Know your workload
- Periodically review structure usage
  - Request rates
  - Service times
  - Storage consumption (within the structure)
  - For lock structures, contention and false contention rates
- Relate changes in structure usage to workload variations
  - Track any “odd” behavior
- Maintain a list of which applications use which structures

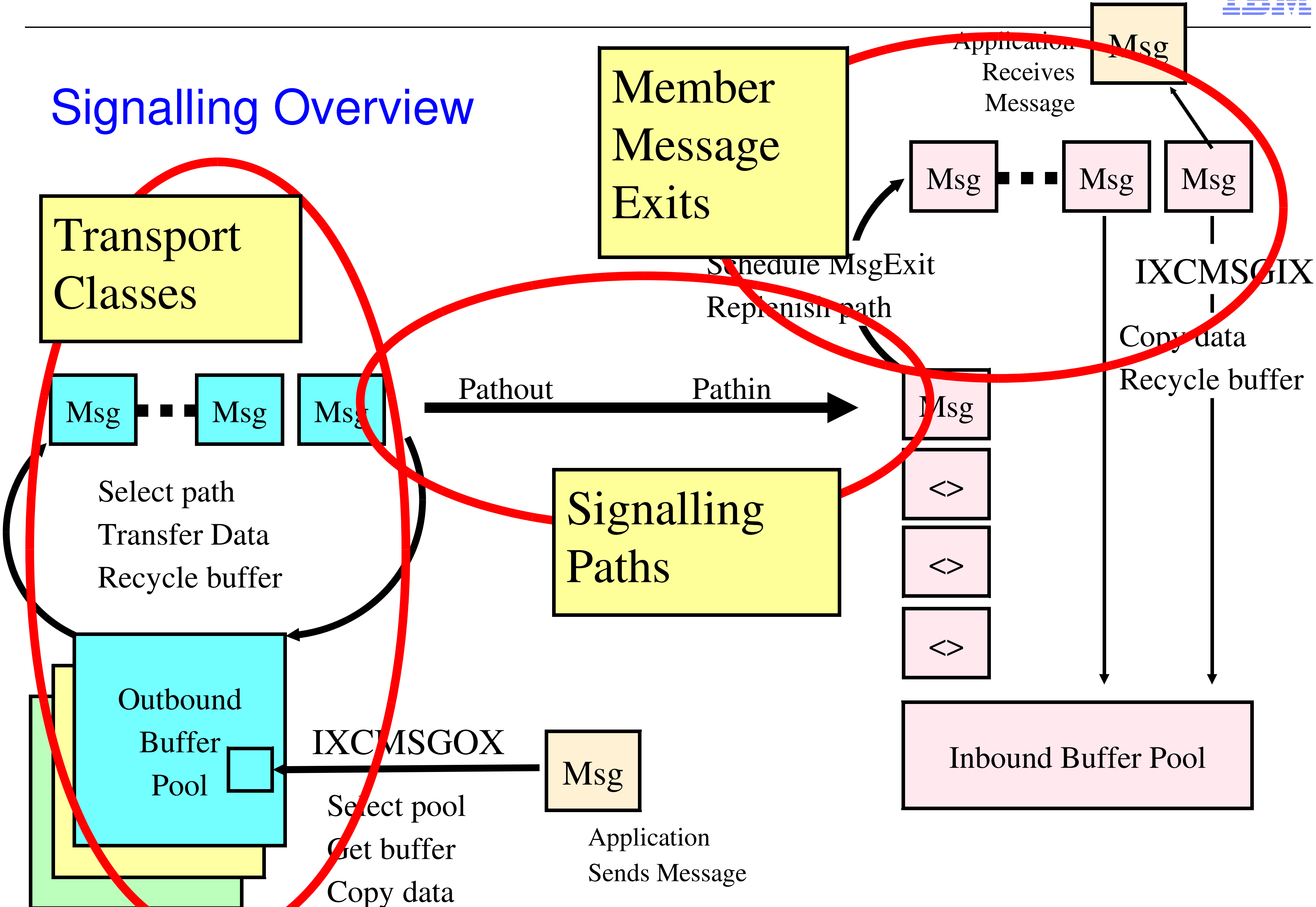
Problem Taxonomy

- Dead System
- Sick System
- Sysplex Fabric
- Sysplex Componentry
  - Coupling Facility
  - **Signalling Service**
  - Couple Data Sets
  - External Time Reference
- Configuration / Capacity
- Software Issues

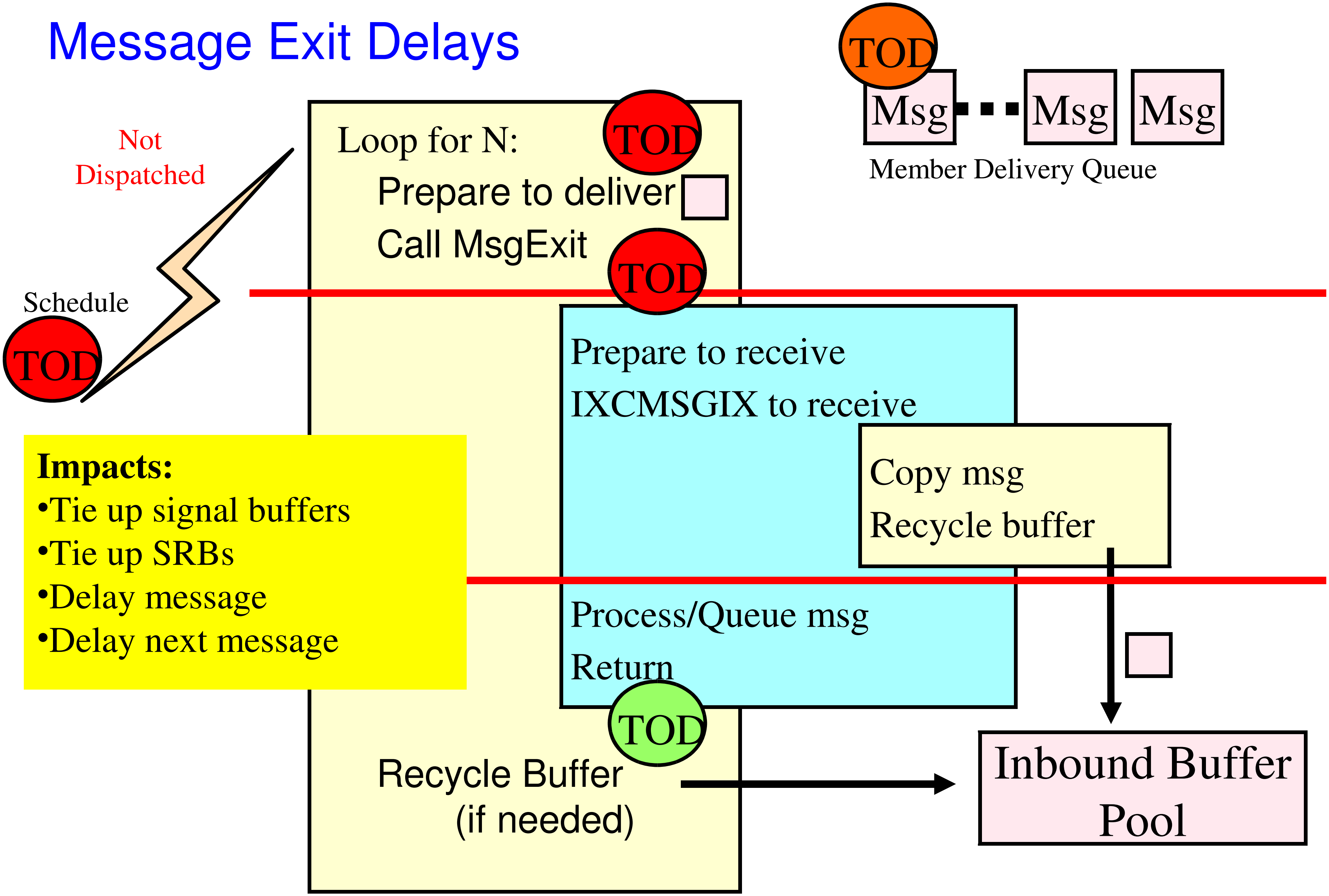
Signalling Overview



Signalling Overview



Message Exit Delays



Impacts:
- Tie up signal buffers
- Tie up SRBs
- Delay message
- Delay next message

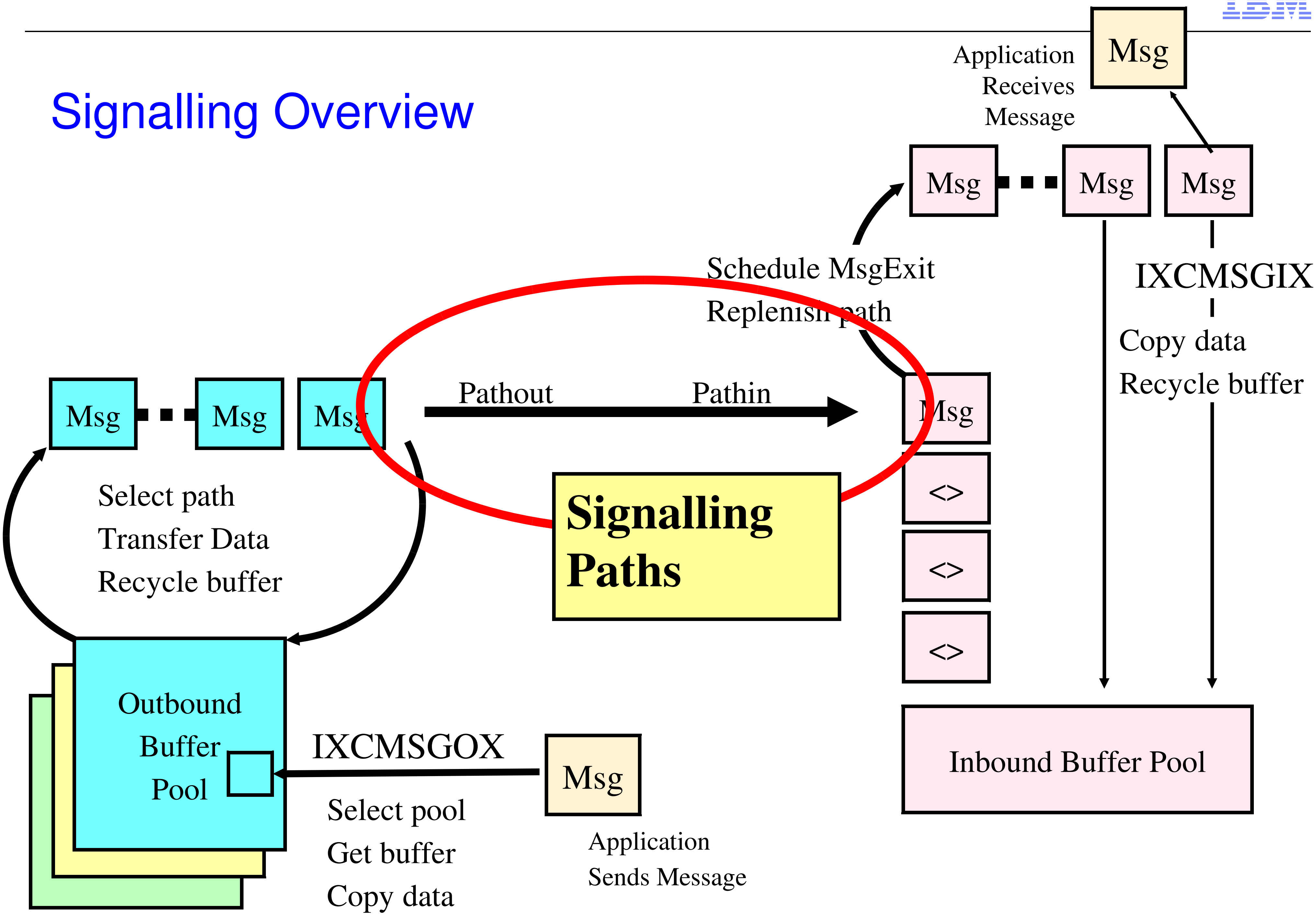
Typical Causes of Message Exit Delays

- CPU Constraints
  - Higher priority work winning
  - Insufficient LPAR weight
- Storage Constraints
  - Page faults
  - Exhausted private
  - Fragmentation
- Contention
  - Waiting for ENQ, local lock, or latch
- Signal Volume
- Defects

Without a timely dump, virtually impossible to diagnose stalls, particularly those of short duration

Try Run Time Diagnostics

Signalling Overview



Signal Path Concerns

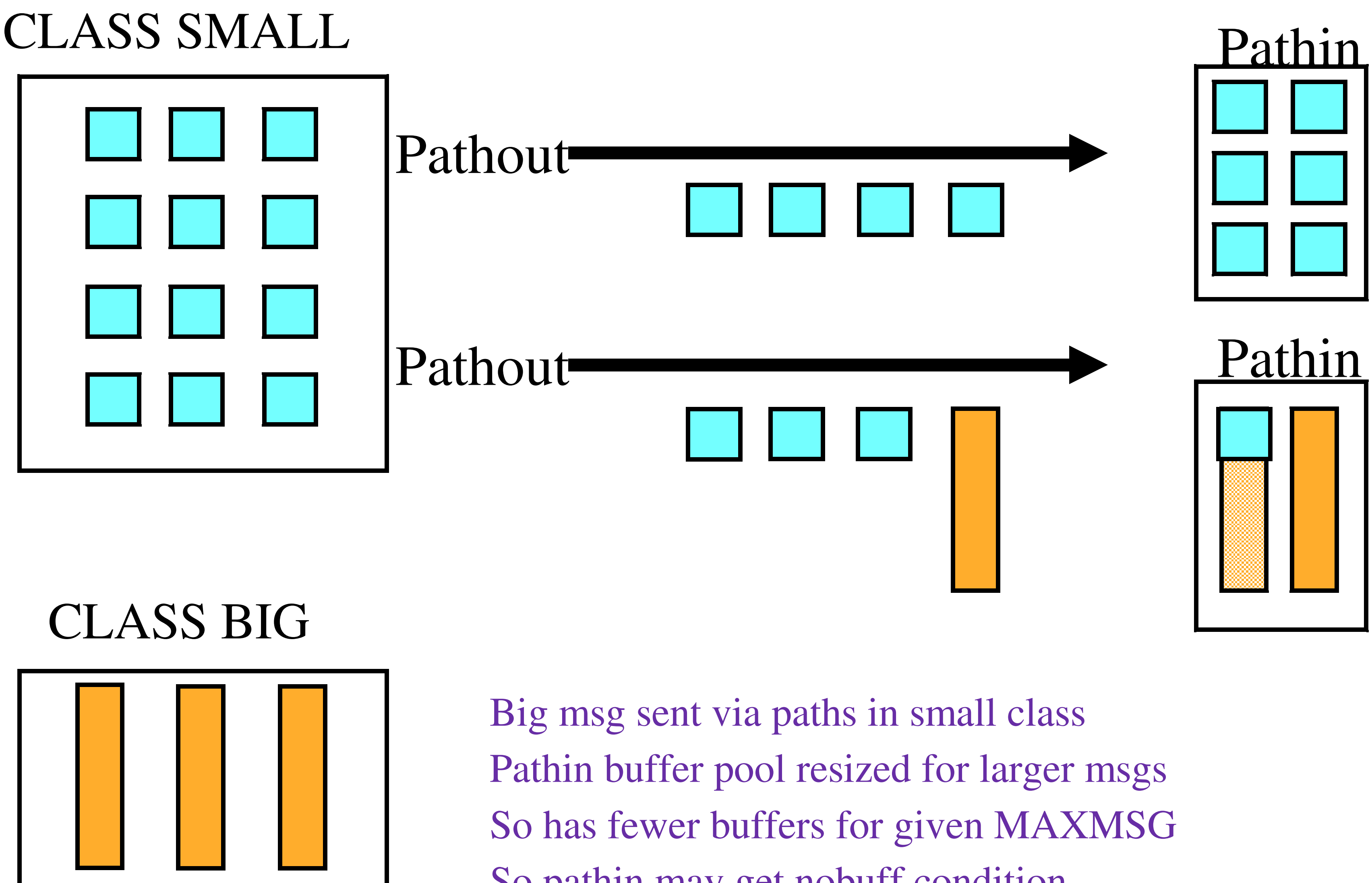
- Target system operational?
- Path operational?
  - Restart? Stop? Rebuild?
- Inbound “no buffer” conditions
- Transfer time

Signal Path Concerns: Inbound Buffers

- Stalled members consuming buffers?
- Lack of Transport Class Segregation on sending side?
  - *Every transport class must have a path for every target system*
  - *Signals may not be flowing via the expected paths*
- Transport Class Length relative to PATHIN MAXMSG?
  - *Number of buffers for given MAXMSG decreases as class length increases*
  - *If you define/modify class length, you may need to modify PATHIN MAXMSG to maintain number of buffers*
- PATHIN MAXMSG too small?
  - *May need more buffers if they tend to dwell in member message delivery*

The z/OS Health Checker looks for these problems

Transport Class Must Have Paths



Big msg sent via paths in small class

Pathin buffer pool resized for larger msgs

So has fewer buffers for given MAXMSG

So pathin may get nobuff condition

So both big and small may get delays

Signal Transfer Considerations

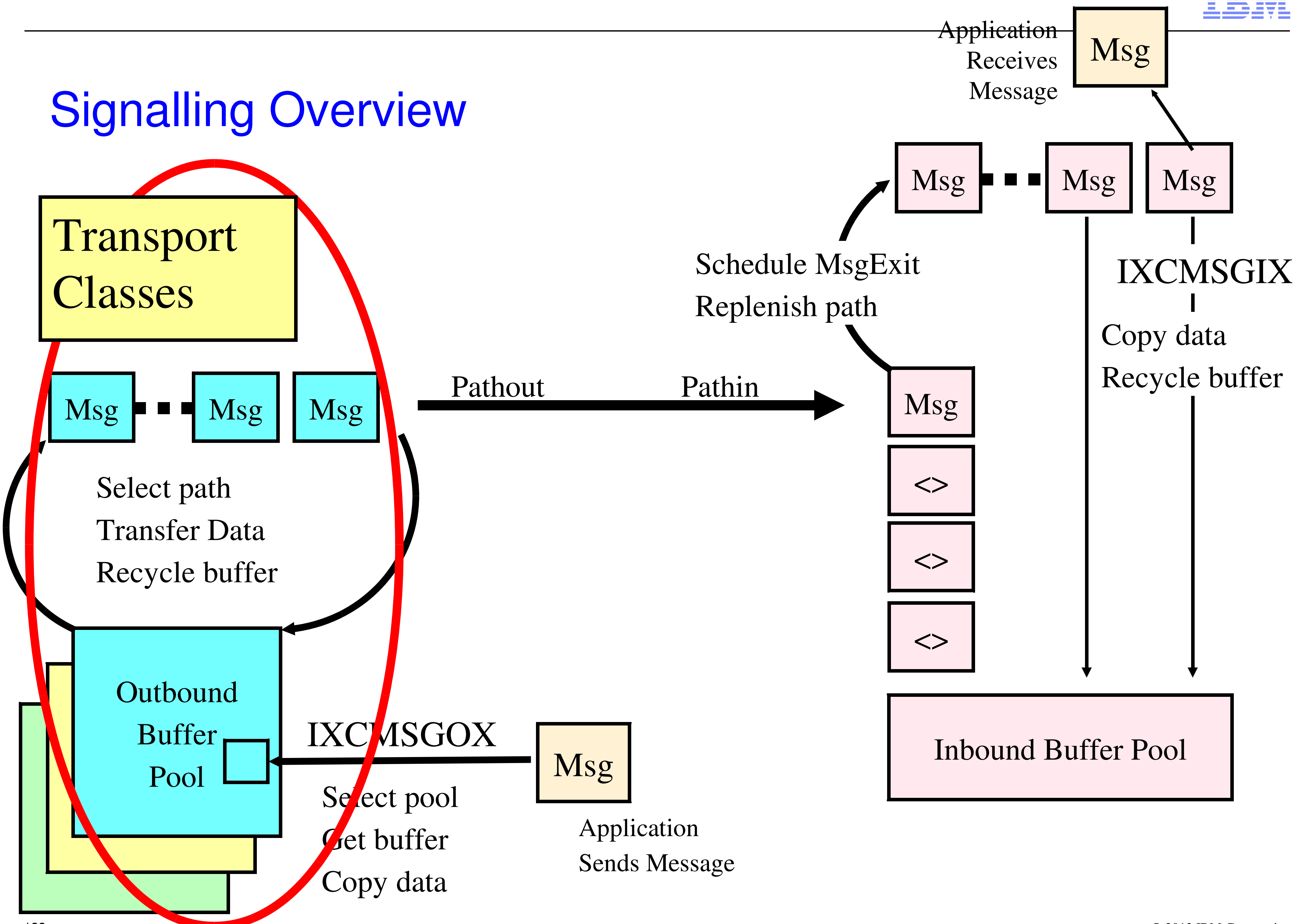
- Queue Time
- Transfer Time
- CTC vs CF Structure

Signal transfers times are interesting in that they may indicate delays.

But signal transfers are generally always good unless you have other problems (stalled members, no buffer conditions, message flooding, CF service time issues, etc). If you resolve those problems, you won't have transfer time issues.

So the handout has plenty of details to explain the technical details. But generally there is nothing to see here. Move along..

Signalling Overview



Transport Class Concerns

- Appropriate segregation
  - –Class Length
  - –Group (?)
- Signalling Paths
  - –**How many (must be > 0)**
  - –Type
- Signal Buffers

Make one class for 956 byte messages. Perhaps one, or two, maybe three classes for bigger message sizes.

Put at least two paths in each class for each target system.

Make sure the PATHIN MAXMSG values are reasonable for the chosen transport class length.

Problem Taxonomy

- Dead System
- Sick System
- Sysplex Fabric
- Sysplex Componentry
  - Coupling Facility
  - Signalling Service
  - **Couple Data Sets**
  - External Time Reference
- Configuration / Capacity
- Software Issues

Couple Data Set Concerns

- Connectivity
  - Access (see “Sysplex Fabric: Couple Data Sets”)
  - Implications for partial connectivity
- Capacity
  - Be sure all formatted for same number of systems
  - Otherwise needs to be formatted to meet needs of exploiter
    - Up to them to indicate when too small
- Performance
  - Response time
  - Contention
- Regression

Couple Data Set Performance

- Various CDS generally have relatively low request rates
  - Though may depend on application design/behavior
  - Sysplex monitors run amok?
    - Be careful how you run RMF and similar tools
- But good performance is needed to avoid
  - Removal of CDS
  - Application delays
- First resolve access issues if any
  - See “Sick System: DASD I/O Issues”
  - See “Sysplex Fabric: Couple Data Sets”

Typical sources of CDS performance issues

- DASD
  - Old and slow?
  - Device caching enabled?
  - Synchronous mirroring?
- Record Size
  - CDS Formatted with too much white space?
- Contention
  - Using MSGBASED processing for CFRM?
  - Reserves?
  - Other data sets with high request rates on same volume?
    - Increased request rates due to workload changes

Couple Data Set Regression

- When IPLing the sysplex, your couple data sets should either be:
  - –The ones most recently used by the sysplex (typical and best), OR
  - –Freshly formatted, never before used (atypical, usually DR site)
- Couple data sets often contain status and configuration data
- Regressing to an older CDS is risky because the data in that CDS may not be consistent with:
  - –The current configuration
  - –Data recorded in other (non-regressed) couple data sets
  - –Related application data recorded in other data sets

You might get some messages prompting operator "Should we use this CDS?" when IPLing. Once the sysplex is up, I don't know of any reliable way to detect that regressed CDS was used.

Couple Data Set Regression ...

- Regression typically occurs when you use:
  - A backup copy of the CDS
  - A former primary/alternate CDS no longer actively in use
  - A CDS that was previously in use by some other sysplex

- NOTE: Bringing an old CDS into use via ACOUPLE is safe

- Resetting the TOD clock can also wreak havoc since time stamps recorded in the CDS may suddenly seem to be:
  - In the distant past
  - From the future

Problem Taxonomy

- Dead System
- Sick System
- Sysplex Fabric
- Sysplex Componentry
  - Coupling Facility
  - Signalling Service
  - Couple Data Sets
  - **External Time Reference**
- Configuration / Capacity
- Software Issues

External Time Reference

- System is in a wait-state if not time synchronized with sysplex
  - See “Sysplex Fabric: External Time Reference”
- Server Time Protocol (STP)
  - Must provide and maintain a resilient configuration as there is potential for sysplex-wide outage !
  - Understand Primary, Backup, and Arbiter roles
- Operational issues
  - Reassigning roles after failure (or planned outage) of any server that has a special role
  - Dealing with STP events
  - Dealing with loss of time synchronization

Operator Alerts for STP Events

- Operator alerts sent to z/OS console as well as HMC for STP related hardware & timing events:
  - Dial-out time service outside allowable tracking range
  - Dial-out access failure
  - NTP server failure
  - NTP servers not synchronized

- IEA031I STP ALERT RECEIVED. ALERT CODE = *nn*

- Available in z/OS 1.11
- Aslo available at z/OS 1.9 and 1.10 with OA28323
- Supported by z10 and z9 servers with MCL driver 76

Problem Taxonomy

- Dead System
- Sick System
- Sysplex Fabric
- Sysplex Componentry
  - Coupling Facility
  - Signalling Service
  - Couple Data Sets
  - External Time Reference
- **Configuration / Capacity**
- Software Issues

Configuration and Capacity Concerns

- Resiliency
- Workload changes
  - Growth
  - Reconfiguration
- That might lead to more:
  - XCF signals
  - CF requests
  - CDS accesses

Preventing Configuration Issues

- Adhere to best practices for availability
  - No single points of failure
  - Ensuring redundancy and fail-over capability often enough to mitigate configuration mistakes (perhaps until failures occur)
- Use IBM Health Checker for z/OS
  - My anecdotal experience leads me to believe that the number of multi-system outages has dropped significantly since the introduction of the health checker

White Paper: "Mission: Available"
www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101966

Preventing Capacity Issues

- Good capacity planning
- Resize structures
- White space and head room for failover

Typical causes of increased activity

- Signals
  - Merge of systems
  - Increase work
  - Application changes
  - GRS contention
  - Migration / Maintenance
- CF requests
  - Merge of systems
  - Increase in work
  - Change in type of work
  - Application changes
  - Processor upgrade or MCL application
- CDS access
  - Systems joining / leaving sysplex
  - Recovery
  - XCF groups joining / leaving

Loss of capacity

- Expect failures. Plan for them.
  - Provide spare capacity and head room for fail over
  - Reduce MTTR by automating restart of failed applications and systems

- See “z/OS MVS: Planning Operations” for information on using Auto-IPL to have failed system automatically IPLed back into the sysplex
  http://publib.boulder.ibm.com/infocenter/zos/v1r12/index.jsp?topic=%2Fcom.ibm.zos.r12.ieag300%2Fwsat.htm

Problem Taxonomy

- Dead System
- Sick System
- Sysplex Fabric
- Sysplex Componentry
  - Coupling Facility
  - Signalling Service
  - Couple Data Sets
  - External Time Reference
- Configuration / Capacity
- **Software Issues**

Methodology Recap

- At this point:
  - We have neither dead systems nor sick systems
  - Sysplex componentry is accessible, operational, and performing
  - Configuration is sufficient to support the workload
- If still have a problem, likely have an application specific issue

Design decisions for sysplex applications

- Communication
  - XCF Signalling Service
  - CF List Structure
- Data sharing
  - Distributed
  - Centralized
- Serialization
  - ENQ
  - XES Locking Services
  - “Owner”
- Topology
  - Peer to peer
  - Master (fixed or dynamic)
  - Ring
  - Star
- Coordination
  - Signals
  - XES Sync points
  - ENQ, locks, latches

Those design decisions have consequences

- Scalability
- Performance characteristics
- Failure modes

- All of which impact problem diagnosis
  – Hard to make progress if don't understand how its put together
  – Will likely need application specific expertise

Preventing software issues

- Stay current with maintenance
- Stay current with releases

- IBM Best Practice: Apply z/OS maintenance RSU regularly and HIPERs more often
  - http://www-03.ibm.com/support/techdocs/atsmastr.nsf/PubAllNum/Flash10106
- IBM Best Practice: Subscribe to Red Alerts
  - http://www-947.ibm.com/support/entry/portal/Overview/Software/Software_support_(general)?pgel=wspace

Summary

- Sysplex problems can be difficult to diagnose
  - Complex inter-dependencies
  - Symptoms may not be directly related to the root cause
- Prevention is highly recommended
  - **Enable SFM with BCPii** to resolve dead system issues
  - Exploit SFM policy to resolve sick but not dead issues
  - Use z/OS Health Checker to verify configuration
- Use the "Problem Taxonomy" to guide your diagnosis efforts. I believe it provides a disciplined approach that can help identify the likely area for root cause, even in the face of imperfect knowledge of the system

#SHAREorg

Questions?

Diagnosing Sysplex Problems
Session 13402

Please fill out the online session evaluation at either:

- SHARE.org/SanFranciscoEval, or
- Aim your smartphone at this QR code below:

Appendix

Non-disruptive CF dump

SETXCF DUMPCF,

{CFNAME=cfname[,TYPE=NONDISRUPTIVE|DISRUPTIVE][,UNCOND=NO|YES]}

{STRNAME=strname}

CFNAME=cfname

Allows the operator to specify the CF to be dumped.

TYPE=DISRUPTIVE|NONDISRUPTIVE

Allows the operator to optionally request a disruptive CF dump.

Default: Nondisruptive

UNCOND=YES|NO

Allows the operator to bypass a confirmation if a disruptive CF dump is requested.

Default: No

STRNAME=strname

Allows the operator to request a CF dump by specifying a structure name. The CF(s) in which the structure resides will be dumped and the dump requested will be non-disruptive.

OA35342 – Introduced the z/OS operator command to collect a non-disruptive serialized CF dump. CF dumps are reviewed by IBM hardware support.

Getting sysplex dumps

- If CF issue, activate SYSXES Ctrace
  - TRACE CT,16M,COMP=SYSXES
  - R XX,OPTIONS=(HWLAYER,REQUEST,CONFIG,CONNECT,RECOVERY,LOCKMGR),END

- If XCF issue, activate SYSXCF ctrace
  - TRACE CT,ON,COMP=SYSXCF
  - R xx,OPTIONS=(GRPNAME=(GEOPLEX),SIGNAL,GROUP,SFM),END

- Place dump command in IEADMCxx parmlib member:
- JOBNAME=(XCFAS),DSPNAME=('XCFAS'.*),
- SDATA=(ALLNUC,CSA,PSA,LPA,LSQA,NUC,RGN,SQA,SUM,SWA,TRT,XESDATA,COUPLE),
- REMOTE=(SYSLIST=*('XCFAS'),DSPNAME,SDATA)),END
- Initiate the dump on one system in the sysplex:
- DUMP COMM=(meaningful dump title),PARMLIB=xx