Putting a lid on XCF

Peter Enrico

Email: Peter.Enrico@EPStrategies.com

Enterprise Performance Strategies, Inc.
3457-53rd Avenue North, #145
Bradenton, FL 34210
http://www.epstrategies.com
http://www.pivotor.com

Voice: 813-435-2297
Mobile: 941-685-6789

z/OS Performance
Education, Software, and
Managed Service Providers

Creators of Pivotor®

Contact, Copyright, and Trademarks

Questions?

Send email to performance.questions@EPStrategies.com, or visit our website at https://www.epstrategies.com or http://www.pivotor.com.

Copyright Notice:

Trademarks:

Abstract

Putting a lid on XCF

- This webinar will walk through how XCF management has been greatly simplified with automatic transaction class management. Significant changes were made to z/OS 2.4 so that transport classes are no longer needed to segregate messages by size or application.

  During this webinar Peter Enrico will discuss all these changes and what needs to be done to take advantage of these XCF enhancements.

EPS: We do z/OS performance…

- We are z/OS performance!

- Pivotor
  - Performance reporting and analysis of your z/OS measurements
  - Example: SMF, DCOLLECT, other, etc.
  - Not just reporting, but cost-effective analysis-based reporting based on our expertise

- Performance Educational Workshops (while analyzing your own data)
  - Essential z/OS Performance Tuning
  - Parallel Sysplex and z/OS Performance Tuning
  - WLM Performance and Re-evaluating Goals

- Performance War Rooms
  - Concentrated, highly productive group discussions and analysis

- MSU reductions
  - Application and MSU reduction

z/OS Performance workshops available

During these workshops you will be analyzing your own data!

- Essential z/OS Performance Tuning
  - March 20-24, 2023
- Parallel Sysplex and z/OS Performance Tuning
  - May 2-3, 2023
- WLM Performance and Re-evaluating Goals
  - October 2-6, 2023
- Also… please make sure you are signed up for our free monthly z/OS educational webinars! (email contact@epstrategies.com)

Like what you see?

- Free z/OS Performance Educational webinars!
  - The titles for our Fall 2022-2023 webinars are as follows:
    - ✓ *Key Reports to Evaluate z16 Processor Caches*
    - ✓ *Understanding System Recovery Boost's Impact on Performance and Performance Reporting*
    - ✓ *WLM Management of DDF Work: What can you do and what has changed?*
    - ✓ *Intensity! Understanding the Concepts and Usage of Intensity Measurements*
    - ✓ *High, Medium, Low: Understanding how HiperDispatch influences performance in z/OS*
    - ✓ *How and why Pivotor is different than other performance management reporters*
    - *Putting a lid on XCF*
    - *Key Reports to Evaluate Usage of Parallel Access Volumes*
    - *Key Reports to Evaluate Coupling Facility CPU Utilization*
    - *Understanding how memory management has evolved in z/OS*
  - Let me know if you want to be on our mailing list for these webinars

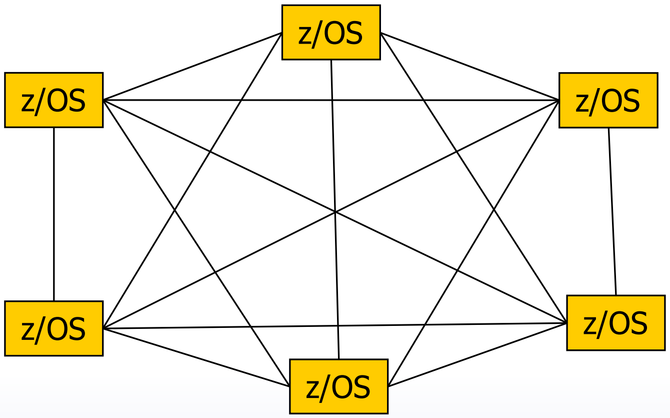
- If you want a free cursory review of your environment, let us know!
  - We're always happy to process a day's worth of data and show you the results
  - See also: http://pivotor.com/cursoryReview.html

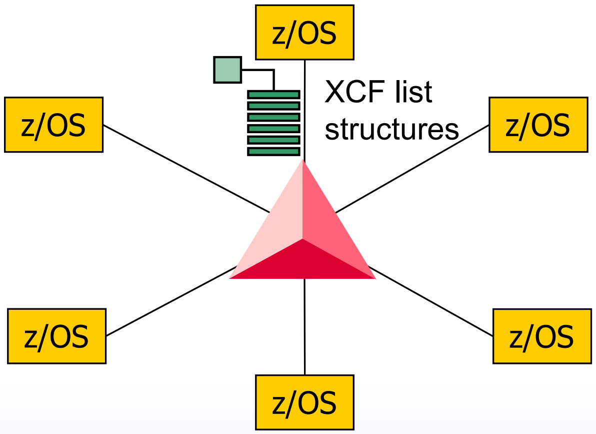
www.epstrategies.com

XCF Refresher

XCF Signaling in a Parallel Sysplex

- All systems need to be able to communicate with every other system
- Think of XCF as the postal delivery system of the z environment
  - Point-to-point messaging via CTCS
  - Messaging via the CF
  - Combination of both





XCF Review (Cross System Coupling Facility)

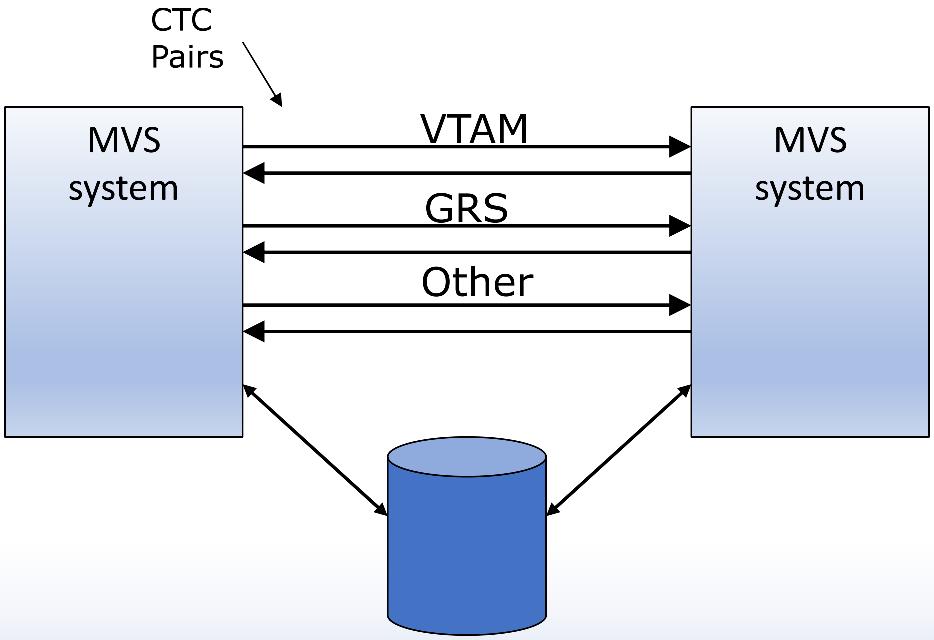
- An z/OS component that provides a means for applications/programs to communicate with applications on
  - same system (a.k.a. local communication)
  - other system (multi-system communication)

- The sending application system does not need to know where the receiving application is

- Objective of XCF
  - Allow applications to communicate to enable them to ‘act as one’
  - Provide the ability to send and receive messages
  - Allow systems to monitor each other’s activity
  - Notify other systems, or applications on other systems, that a failure has taken place. Communication is then used in recovery

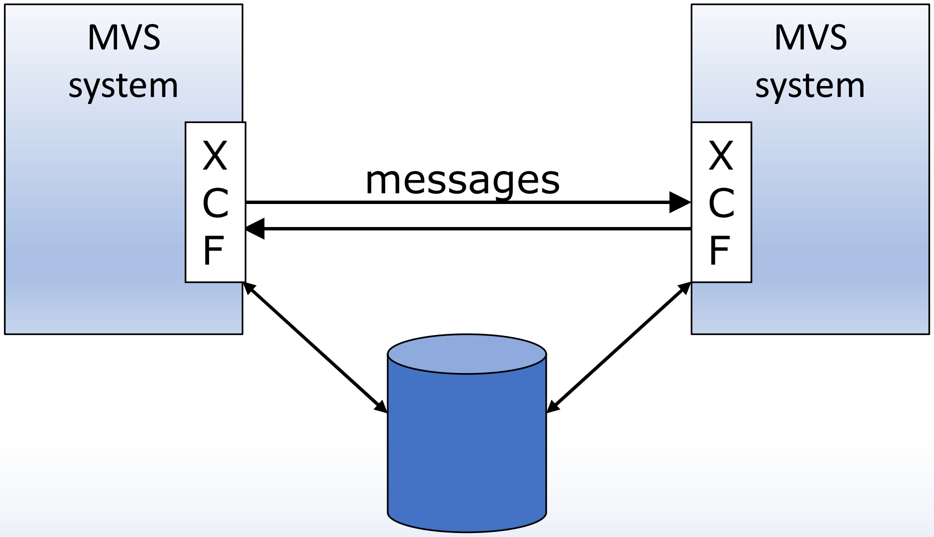
Sysplex Before XCF

- Each subsystem was required to provide its own transport mechanism for intersystem communication
  - Thus, each had to provide its own recovery mechanisms, signaling protocols, and tuning methodologies
  - Resulted in significant hardware and software planning efforts



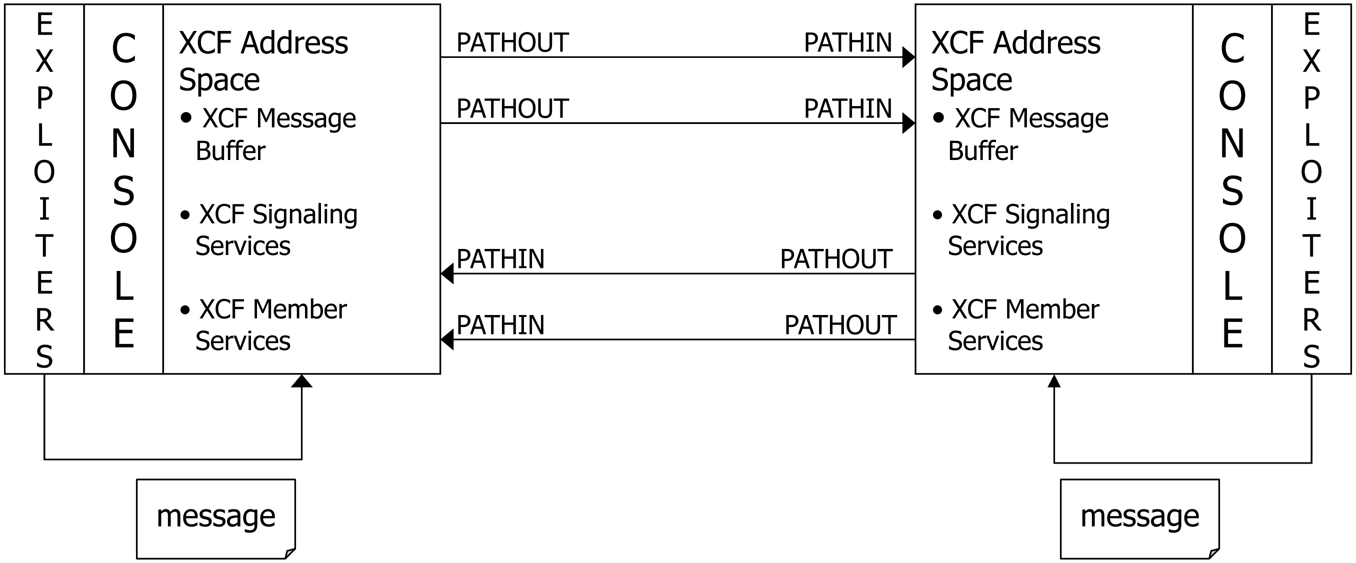
Sysplex After XCF

- XCF is a single message transportation protocol that any application could use
  - Consistent recovery, restart, tuning, and management for all exploiters
  - Once set up, any number of applications/programs can use it with no need for additional hardware or programming investment
- XCF owns the connections between systems, and multi-system applications connect to other systems by exploiting the XCF services
  - Each application does not need to have it own signaling path or private signaling service



Overview of XCF Environment

- XCF Exploiters are any z/OS facility that wants to send an inter-system message



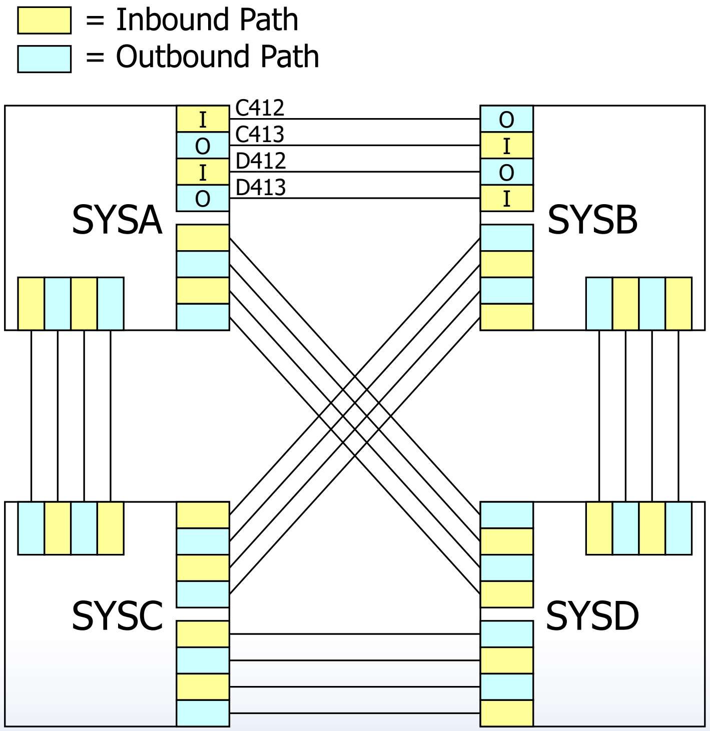
XCF Terminology

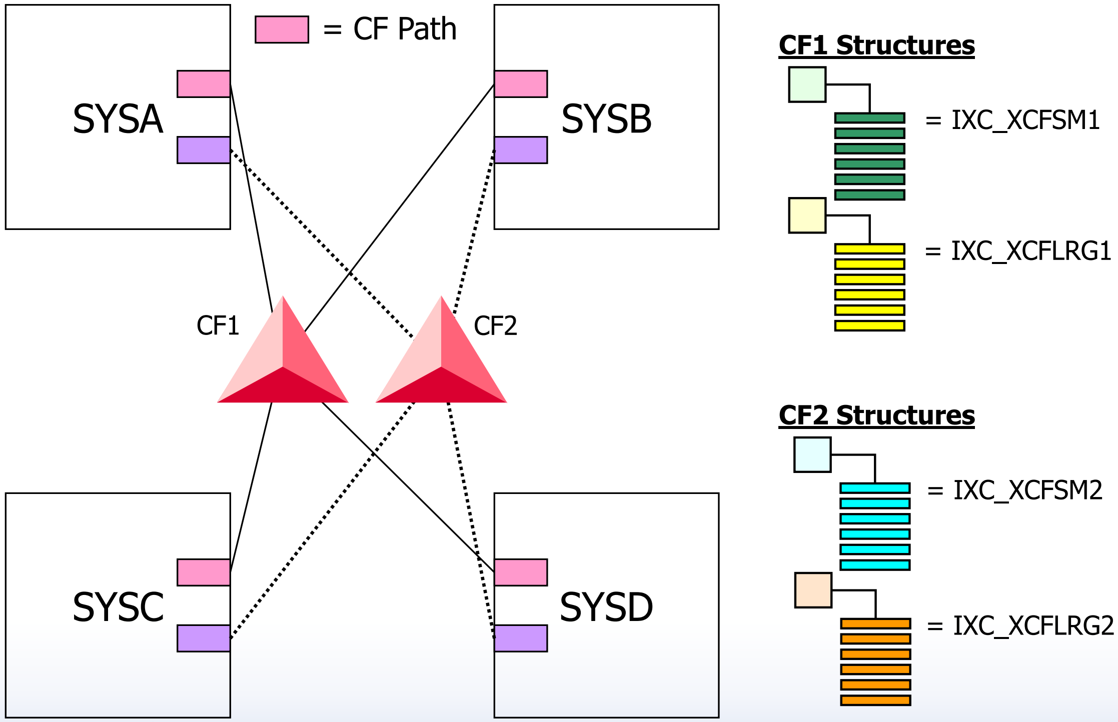
- In a basic Sysplex XCF signaling services are used for system communications
  - Physical basic Sysplex signaling resources include:
    - Signaling paths
    - Message buffers
  - Access to these physical signaling resources is controlled by:
    - Transport classes
  - Transport classes are made up of like message traffic defined in:
    - XCF groups
    - XCF group member

- To measure and tune the XCF environment we need to understand:
  - XCF signaling paths
  - Message buffers
  - Transport classes
  - XCF groups
  - Members of XCF groups

Example of XCF Paths

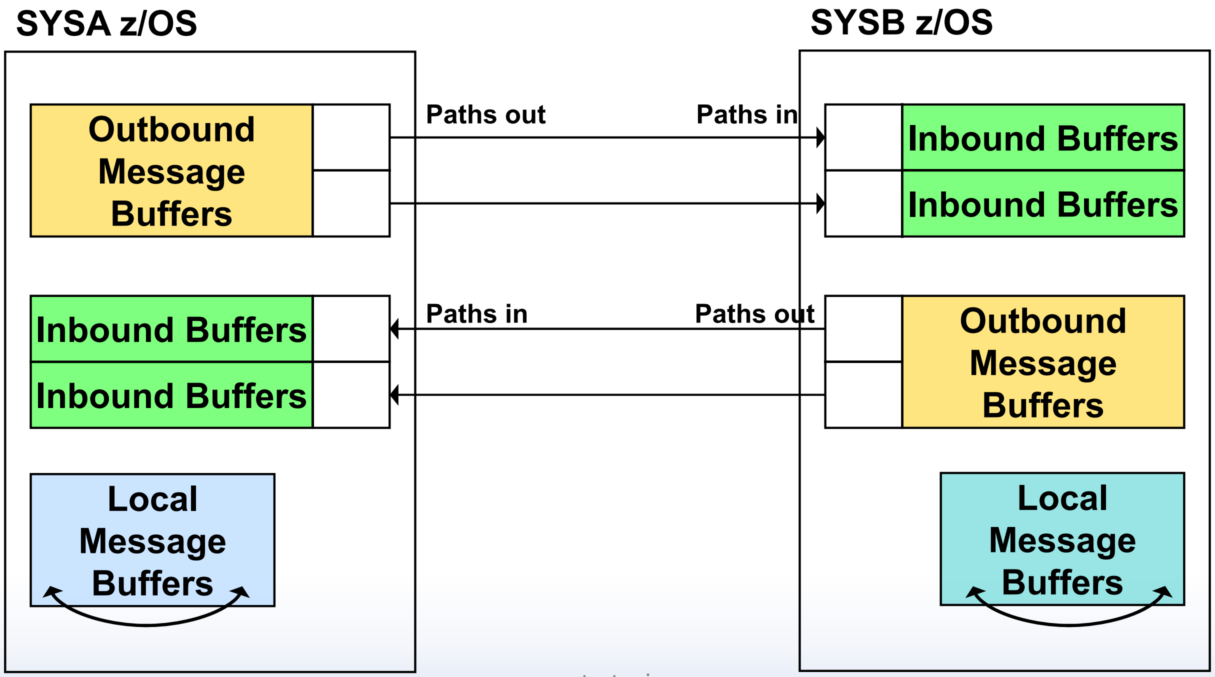
- A signaling path is unidirectional.
  - This means that a path is either designated as ‘inbound’ or ‘outbound’, but it cannot be both
  - Many installations have a combination of both types of paths





XCF Message Buffers

- Three types of message buffers (Inbound, Outbound, Local)
  - There is a supply of buffer space for each system being communicated with
  - CLASSDEF CLASSLEN parameter determines the size of the XCF message buffers used to hold the message of a transport class



XCF Buffer Sizes

- XCF has internal buffers of fixed sizes:
  - 1K, 4K, 8K, 12K, 16K, 20K, 24K, 28K, 32K, …, 64K

- But XCF uses 68 bytes for internal control blocks
  - Thus, the 1K buffer is really for messages length 1024-68 or 956 or less

- Example of max message size for a buffer
  - 956, 4028, 8124, 12220, etc..

- So, when setting up your buffers, if you specify a length that does not fit one of these sizes, XCF will round up to next largest size
  - Thus, a message size of 1024 not fit into a 1K buffer so a 4K buffer will be needed

XCF Groups and Members

- XCF signaling is provided so applications can communicate without having to create their own signaling protocols or require their own unique hardware

- XCF Groups
  - A set of related members that a multi-system application defines to XCF
  - Communication between members on different systems occurs over signaling paths
  - To prevent multi-system applications from interfering with one another, each XCF group has a unique name within the Sysplex

- Member
  - A specific function or part of a multi-system application
  - Defined to XCF and assigned to an XCF Group
  - Members on different systems can communicate via sending and receiving messages with other members in the same XCF Group

- The relationship of XCF groups and members is usually dictated by the multi- system application

Examples of Exploiters of XCF
(Not a complete list…)

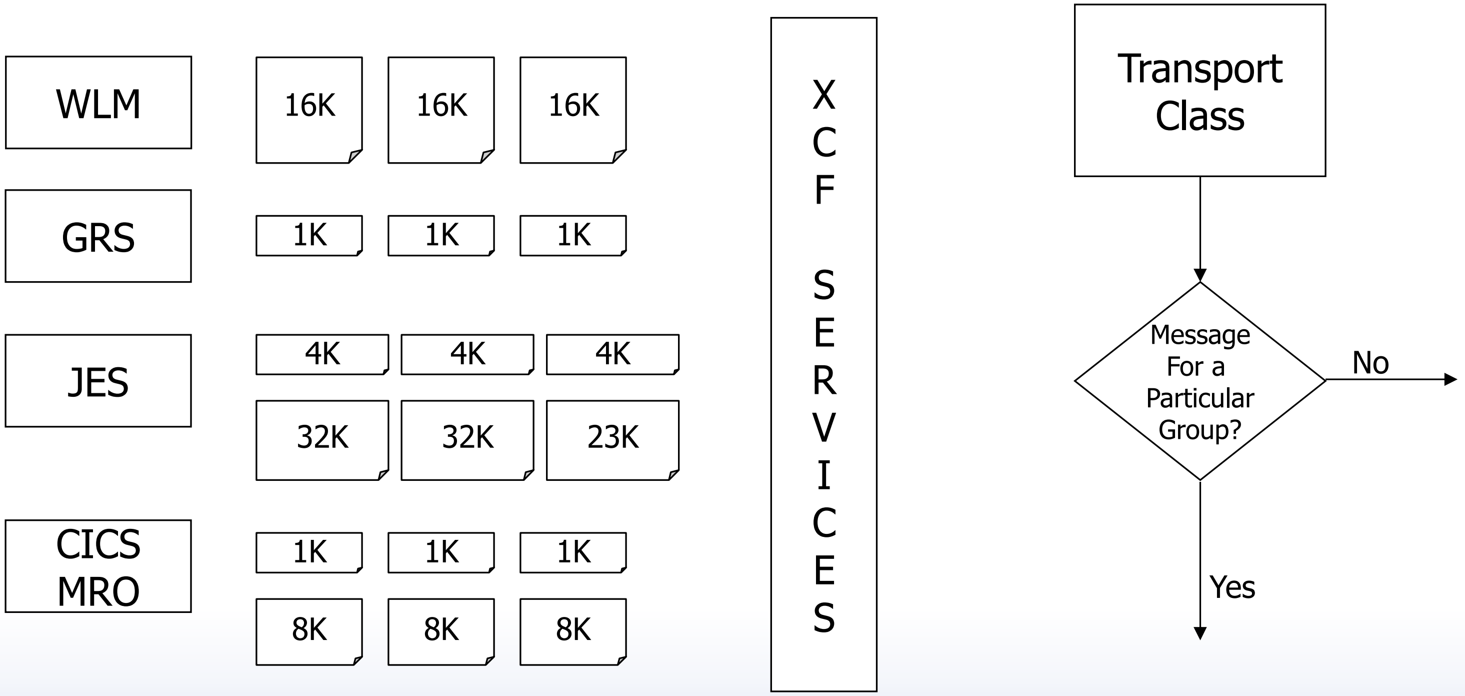
Group	Owner
AOFSMGRP	AOC
ASFBGRP1	AOC
ATRRRS	* RRS
BBGROUP	CPSM
COFVLFNO	* VLF
DFHIR000	CICS
DSNDB1G	DB2
DXRDBZG	DB2
EJESEJES	EJES
ESCM	ESCOM MGR

Group	Owner
EZBTCPCS	BatchPipes
IDAVQUIO	VSAM
IGWXSGIS	VSAM RLS
IRLMGRP1	IRLM
IRRXCF00	RACF
ISTCFS01	VTAM
ISTXCF	VTAM
IXCLOxxx	*# XES
JES2xx (local name)	$JES2 MAS
JES3xx Node name on NJERMT init statement	@JES3 Cmplx

Group	Owner
NJEx	NJE-JES2
SYSATBxx	APPC
SYSDAE	* DAE
SYSENF	* ENF
SYSGRS	* GRS
SYSIGW00	DFSMS PDSE
SYSMCS	* CONSOLES
SYSMCS2	* CONSOLES
SYSRMF	RMF
SYSWLM	* WLM

XCF Exploiters Send and Receive Messages

- Messages are sent and received by exploiters of XCF signaling services
  - Characteristic of message is dependent upon the exploiter and the contents of the message
  - Exploiter issues a signal and XCF will then transport the message based on XCF setup



XCF Transport Classes

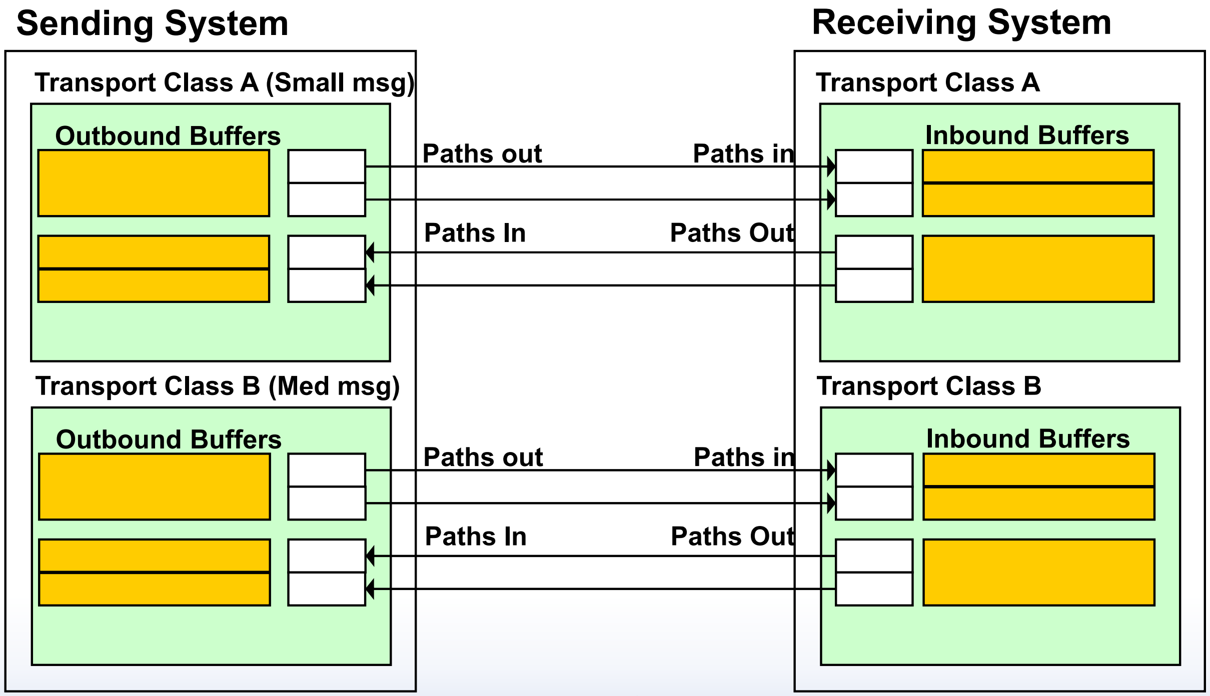
- A transport class is what is used to associate one or more XCF groups (with similar signaling requirements)

- A transport class is assigned to signaling resources that can meet the signaling requirements.
  - Signaling paths (inbound and outbound)
  - Message buffers

- Transport classes allow you to segregate message traffic based on
  - Length of messages
    - To help ensure that an optimal message buffer size or pool of buffers is available
  - Needs of the application's group
    - You can dedicate signaling resources to a particular application to decrease the competition for signaling resources, thus helping to ensure better performance
  - Both

Typical Transport Class Setup

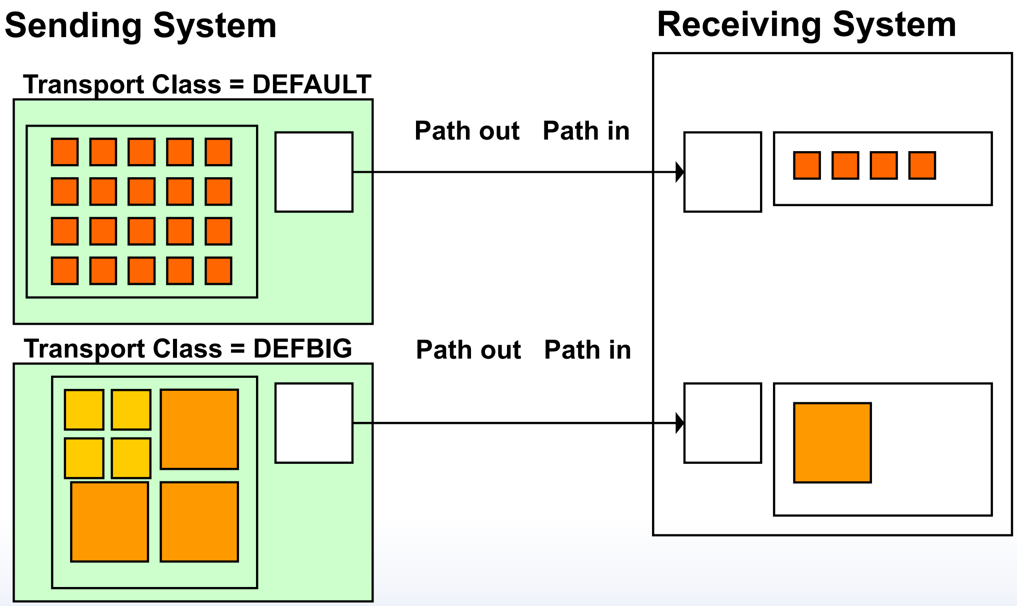
- Prior to z/OS 2.4, installations typically had multiple transport classes
  - Each assigned a different set of signaling resources (buffers and paths)
  - Usually designated for a particular size message



Flow of an XCF Message - Based on Length

Sending System
- If message length < TC DEFAULT CLASSLEN then use TC DEFAULT else use TC DEFBIG
- Copy message into buffer
- Select path on which to send message
- Initiate transfer of message over path to target system

Receiving System
- Receives message on inbound path
- Places message into inbound buffer
- Coordinates message to be delivered to target member (i.e. the application)



Example of Segregation by Size

- Assume the following COUPLExx statements
  - For this example, assume the DEFAULT transport class does not exist)

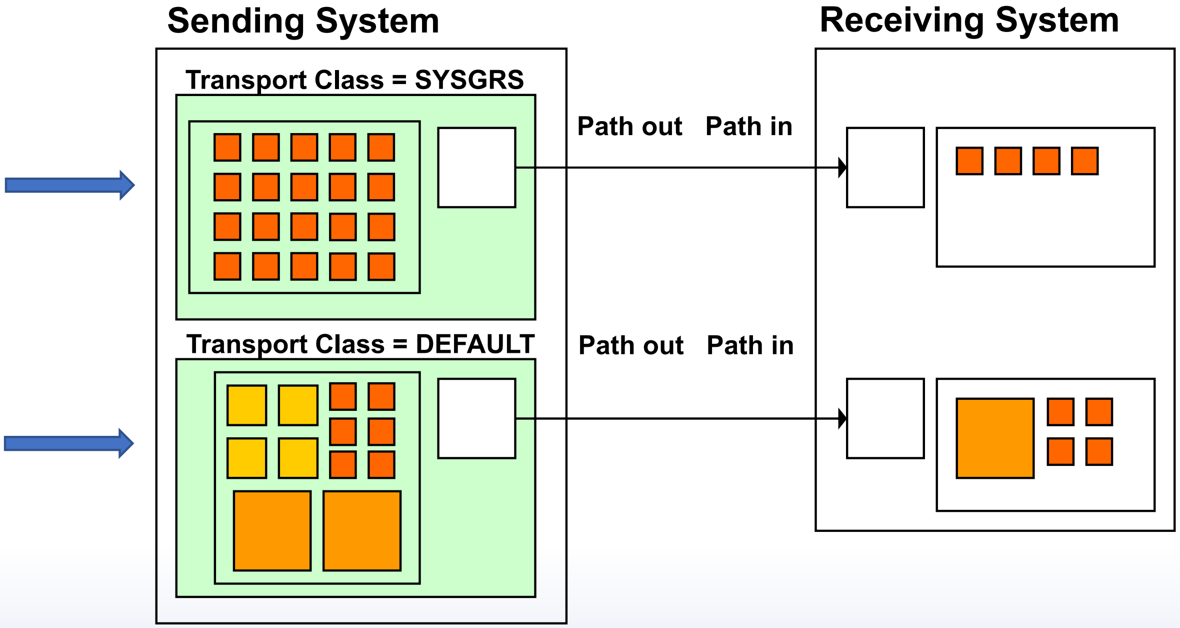
```
CLASSDEF CLASS(DEFAULT) CLASSLEN(956)   MAXMSG(1000)
CLASSDEF CLASS(DEFMED)   CLASSLEN(8124) MAXMSG(1000)
CLASSDEF CLASS(DEFBIG)   CLASSLEN(32000) MAXMSG(1000)

PATHOUT DEVICE(C410)        CLASS(DEFAULT)
PATHOUT STRNAME(IXC_SMALL) CLASS(DEFAULT)
PATHOUT DEVICE(C412)        CLASS(DEFMED)
PATHOUT STRNAME(IXC_MED)    CLASS(DEFMED)
PATHOUT DEVICE(C414)        CLASS(DEFBIG)
PATHOUT STRNAME(IXC_BIG)    CLASS(DEFBIG)
```

- The message will be grouped as follows:
  - DEFAULT - messages 1 to 956 in length
  - DEFMED - messages 957 to 8124
  - DEFBIG - messages 8125 to 62000

Flow of an XCF Message - Needs of Appl

Sending System
- If message is from GRS, use TC=SYSGRS
- Else use TC=DEFAULT
- If no buffer of correct size in selected TC, make one and copy message into buffer
- Select path on which to send message

Receiving System
- Receives message on inbound path
- Places message into inbound buffer
- Coordinates message to be delivered to target member (i.e. the application)



Note: Transport class and dedicated resources setup just for GRS

Note: Transport setup for all other messages

```
CLASSDEF CLASS(TCGRS)
          CLASSLEN(956)    MAXMSG(1500)
          GROUP(SYSGRS)

CLASSDEF CLASS(DEFAULT)
          CLASSLEN(62464)    MAXMSG(1500)
          GROUP(UNDESIG)
```

z/OS 2.4 Changes to XCF

Simplification!

z/OS v2.4 XCF Transport Class Simplification

- z/OS 2.4. Eliminates the need to define size only transport classes
  - Segregation of messages purely by size
  - XCF transport classes more self-managing and self-tuning
    - No longer need to tune and optimize XCF transport classes message sizes to match the signaling workload characteristics
    - Also results in decreased number of path definitions, etc.
  - No longer static definition for assignment of resources
    - System automatically applies resources where needed
  - Avoid performance and resiliency impacts from poorly-tuned transport class sizes
    - Also, improve resiliency by avoiding monopolization of message buffer space
  - New/improved statistics for reporting message path utilization, signal counts, and no-buffer conditions

- Later planned support will address group segregation
  - Isolation of ill-behaved members to avoid sympathy sickness
  - One member will not negatively impact signal delivery of other members

New _XCFMGD (pseudo) transport class

New control XTCSIZE to enable/disable new support

- Basically, a chicken switch
- When set to DISABLED, XCF signaling resources are managed as they were prior to z/OS 2.4
- When set to ENABLED, _XCFMGD transport class used

- Can disable or enable the XTCSIZE switch dynamically with the SETXCF FUNCTIONS operator command

```
SETXCF FUNCTIONS,DISABLE=XTCSIZE
SETXCF FUNCTIONS,ENABLE=XTCSIZE
```

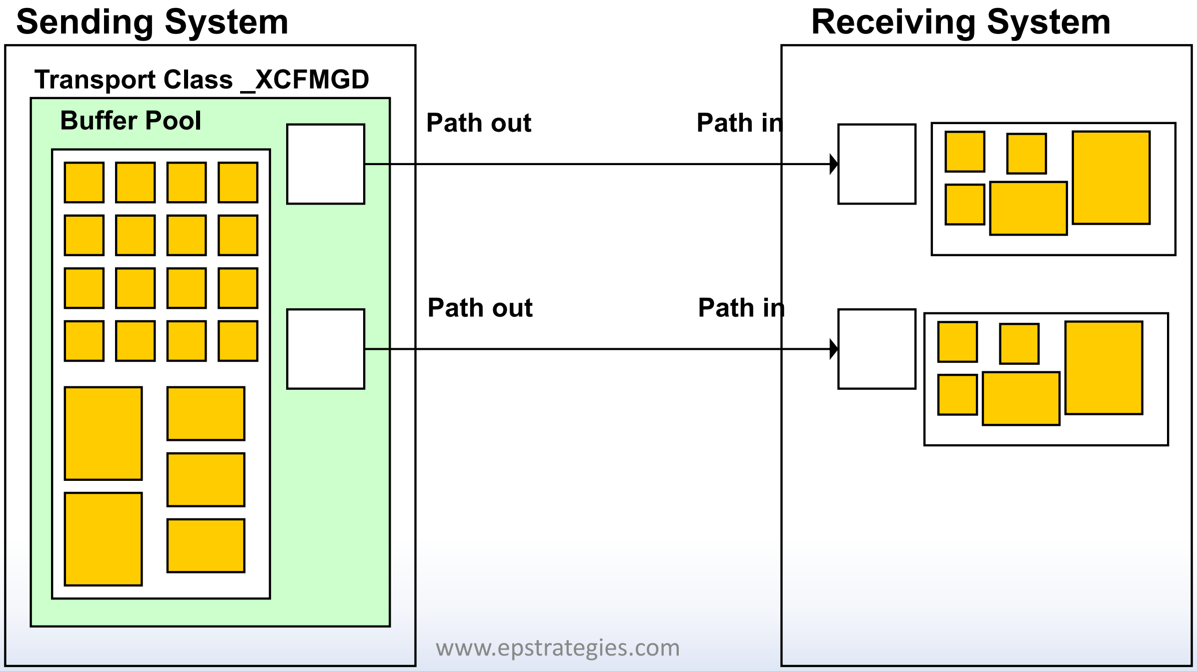
New _XCFMGD (pseudo) transport class in COUPLExx member of parmlib

- Implicitly defined by XCF (thus, it always exists)
  - Will not be used if XTCSIZE is DISABLED or if target system is pre-z/OS V2R4
  - Installation cannot directly control its attributes (classlen=0, XCF determines MAXMSG)
  - When XTCSIZE is ENABLED, all paths in the “XCF Managed” classes are logically reassigned to the _XCFMGD transport class
- Algorithm uses the “best fit” buffers on the send side
  - Maximizes number of signals that can be accepted for a given MAXMSG limit to better handle bursts of activity and delays
  - As a reminder, traditional classes generally use the “defined size” which might not be best fit
- Paths run at the maximum signal size
  - Thus, any message can be transmitted without any additional overhead
    - Never need to re-negotiate signal size (or tune) the signal paths

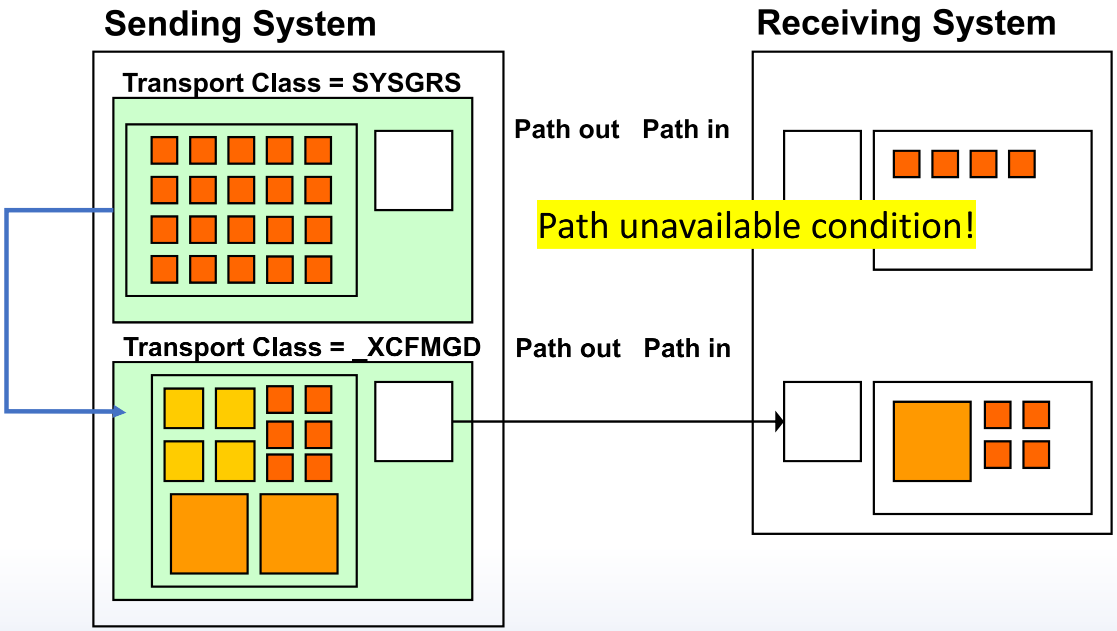
Example of using _XCFMGD

Sending System
- Obtain an outbound message buffer
- Select path on which to send message
- Initiate transfer of message over path to target system

Receiving System
- Receives message on inbound path
- Places message into inbound buffer
- Coordinates message to be delivered to target member (i.e. the application)



Flow of an XCF Message - Needs of Appl



Path unavailable condition!

```
CLASSDEF CLASS(TCGRS)
          CLASSLEN(956)    MAXMSG(1500)
          GROUP(SYSGRS)

CLASSDEF CLASS(DEFAULT)
          CLASSLEN(62464)    MAXMSG(1500)
          GROUP(UNDESIG)
```

Migration notes:

- May need to maintain your old COUPLExx XCF definitions
  - It is very likely that when migrating to z/OS 2.4, not all systems in the Sysplex will be migrated at the same time
  - Thus, it is very possible that during migration to z/OS 2.4 that some systems in the Sysplex will be back=level
- The traditional transport class definitions intended to manage size segregation should be maintained until all systems in the Sysplex are running z/OS V2R4
- Lesson, do not remove all setup from COUPLExx member until all systems are migrated
  - In addition, keeping the old definitions will allow for selective XTCSIZE disablement

COUPLExx member changes

Pre-z/OS 24

```
CLASSDEF CLASS(DEFAULT) CLASSLEN(956)
CLASSDEF CLASS(MSG08K)  CLASSLEN(8124)
CLASSDEF CLASS(MSG16K)  CLASSLEN(16316)
CLASSDEF CLASS(MSG24K)  CLASSLEN(24508)
CLASSDEF CLASS(MSG32K)  CLASSLEN(32700)

PATHIN  STRNAME(IXCSIG1,IXCSIG2) MAXMSG(2000)
PATHIN  STRNAME(IXCSIG3,IXCSIG3B,IXCSIG4,
                IXCSIG5,IXCSIG6)

PATHOUT STRNAME(IXCSIG1,IXCSIG2) CLASS(DEFAULT)
PATHOUT STRNAME(IXCSIG3)         CLASS(MSG08K)
PATHOUT STRNAME(IXCSIG3B)        CLASS(MSG08K)
PATHOUT STRNAME(IXCSIG4)         CLASS(MSG16K)
PATHOUT STRNAME(IXCSIG5)         CLASS(MSG24K)
PATHOUT STRNAME(IXCSIG6)         CLASS(MSG32K)
```

z/OS 2.4 +

```
PATHIN  STRNAME(IXCSIG1,IXCSIG2) MAXMSG(2000)
PATHIN  STRNAME(IXCSIG3,IXCSIG3B,IXCSIG4,
                IXCSIG5,IXCSIG6)

PATHOUT STRNAME(IXCSIG1,IXCSIG2)
PATHOUT STRNAME(IXCSIG3)
PATHOUT STRNAME(IXCSIG3B)
PATHOUT STRNAME(IXCSIG4)
PATHOUT STRNAME(IXCSIG5)
PATHOUT STRNAME(IXCSIG6)
```

(Or could just leave the definitions alone, and XCF will ignore if XTCSIZE is enabled)

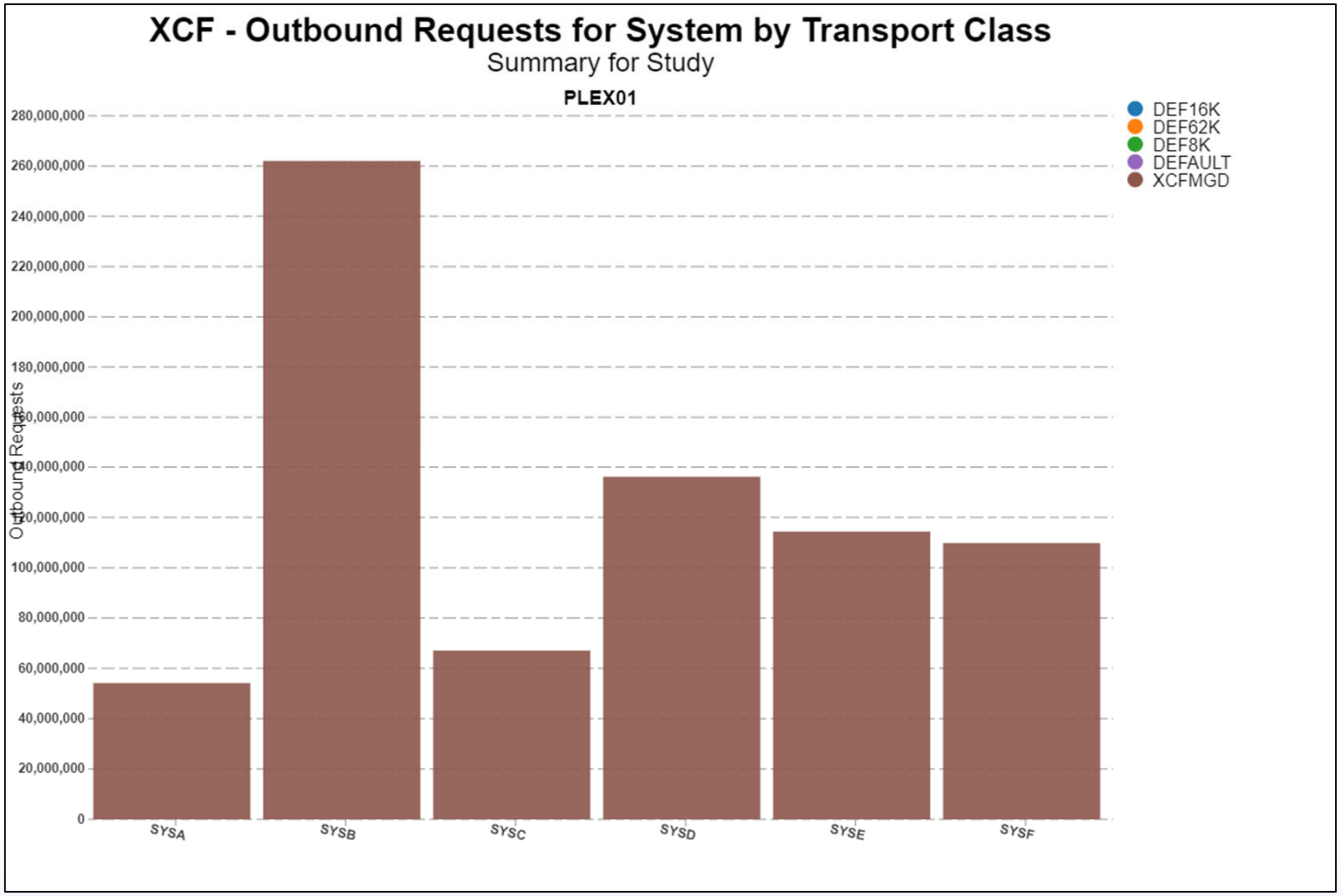
z/OS 2.4 XCF Measurement and Tuning

Approach to Analyzing and Tuning XCF

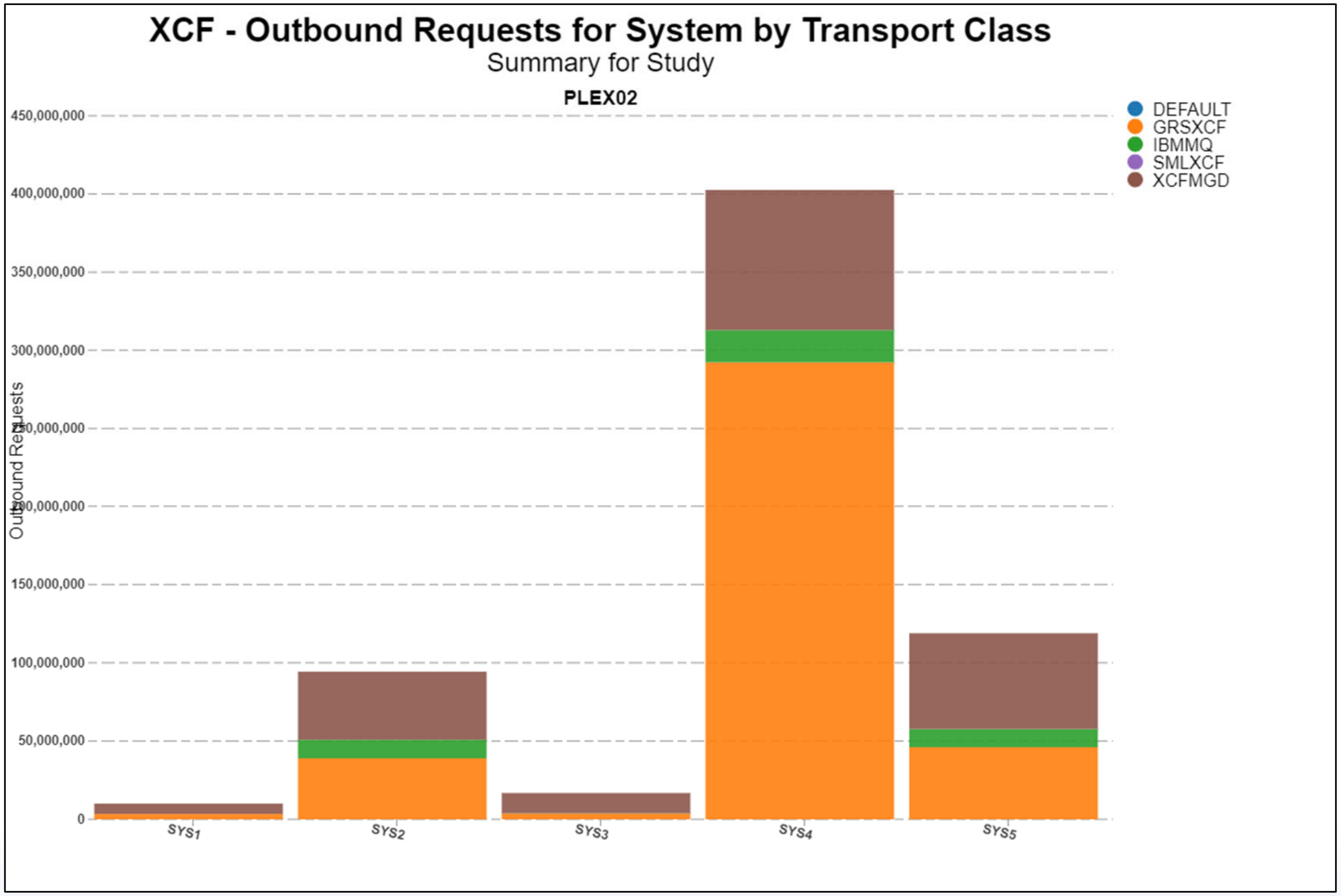
- Remember that XCF analysis has to be done for each separate system in the Sysplex

- For each subject system of this analysis:
  - Understand the physical XCF signaling environment
    - Analyze XCF parameters in COUPLExx
    - Analyze XCF Path Statistics (SMF 74.2) and RMF/CMF XCF Path Statistics report
  - Analyze the path statistics to determine
    - System to system load and performance
    - Usage of physical signaling resources
  - Analyze transport classes
    - Utilization of transport classes
      - _XCFMGD, and if any other transport classes are used for segregation by application
    - Buffer usage
  - Analyze XCF groups and members
    - Analyze which applications are driving the signaling
    - Analyze message sizes and loads by application

Example XCF MSG Segregation by Size



Example XCF MSG Segregation by Size and Application



Comments from Jamie… and then Q & A

Questions about content of webinar?

Of maybe general performance questions?