# *IP Monitoring on z/OS*

## *Requirements and Techniques*

Gordon Webber
William Data Systems

Session 8195
February 2011

*Gordon.Webber@willdata.com*

# Topics

- **Why monitor IP ?**
- **IP monitoring Requirements**
  - What should be monitored
- **IP monitoring Issues**
  - Things to think about
- **IP monitoring Techniques**
  - How it can be achieved

# Why Monitor IP?

**Networks are *dynamic*, definitions change, and things *CAN* go wrong:**

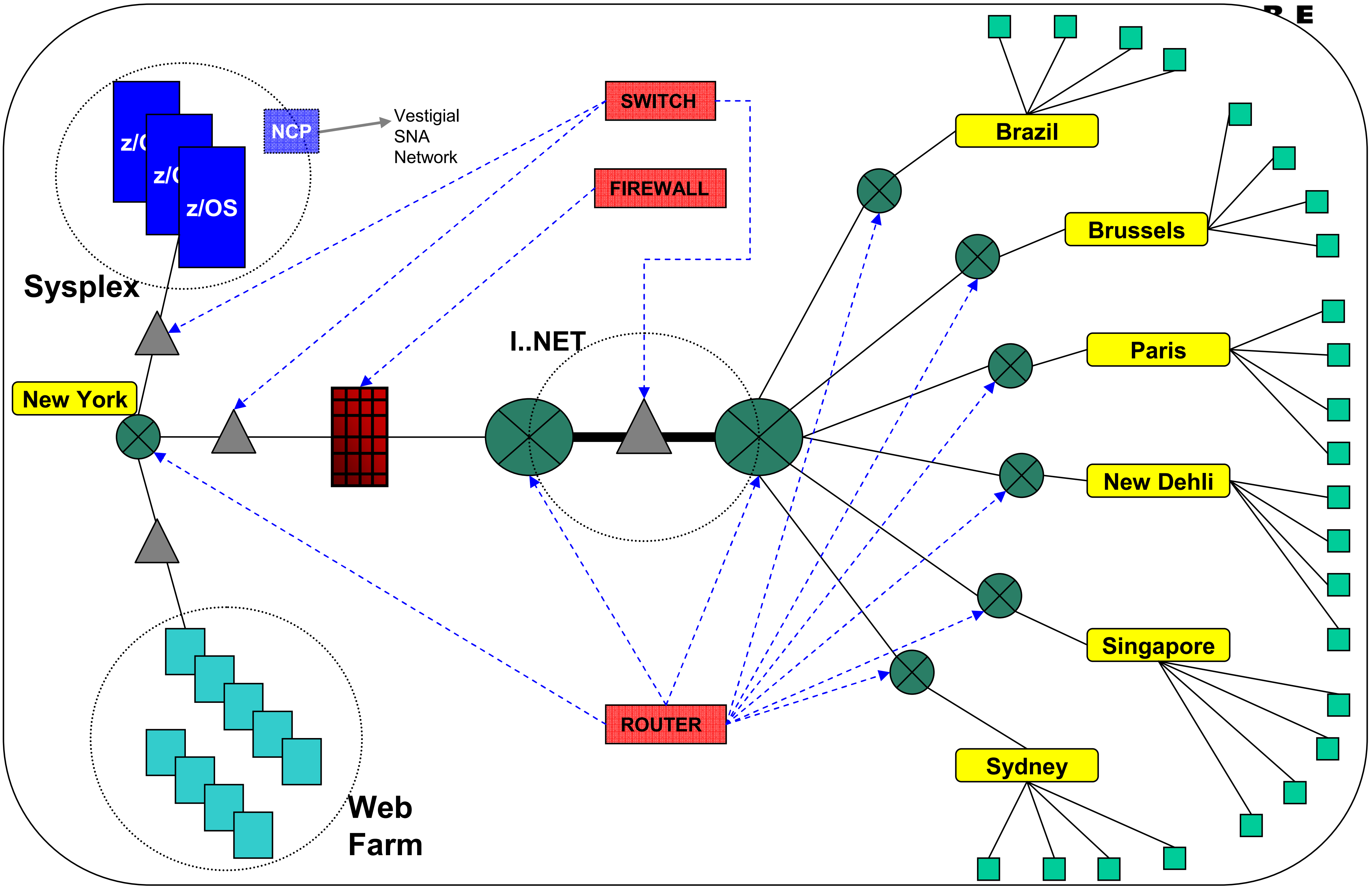
- Changes/Updates happen all the time!

- The "WAN" may be managed by another staff groups

- Synchronising changes is not always possible

**There are several areas in the network where these risks exist, all of which could affect z/OS services …**

# Network Risk Areas



# Network Risk Areas

**Possible cause of problems:**

- Hardware Failure
- Configuration Change (lost rights, paths, MTU)
- Firmware Change loses Configuration
- Traffic Rates Change – congestion
- New Application: port conflict, packet size (fragment)
- Cable Fault / Severed Cable
- WAN Switch Failure
- WAN DNS Failure
- Security Attack
- Lost  Secure Information

# Why Monitor IP?

## “It’s a Network Problem!”

- Access issues, poor response times, connection drops, and unexpected behaviour of network applications are often blamed on the network.

- The network administrator usually has to prove where the fault lies.

- This is not pro-active and wastes time… ***And money*!**

# Why Monitor IP?

- **IP encompasses :**
  - TCP , UDP , ICMP , OSPF , Others
- **Critical to providing service on z/OS**
  - TCP/IP services: Telnet, FTP, WebSphere, MQ …
  - SNA services: TN3270, Enterprise Extender
  - Perhaps even X.25 !

*(Are you meeting your Response Times?)*

- **Fault tolerance**
  - Protocols and features “hide” problems
  - System resources – too late when it runs out
- **Security**
  - IP networks are often “open”, therefore security is a serious issue; … externally ***and*** internally.

*(Just who **is** using your network ?)*

# Why Monitor Sysplex

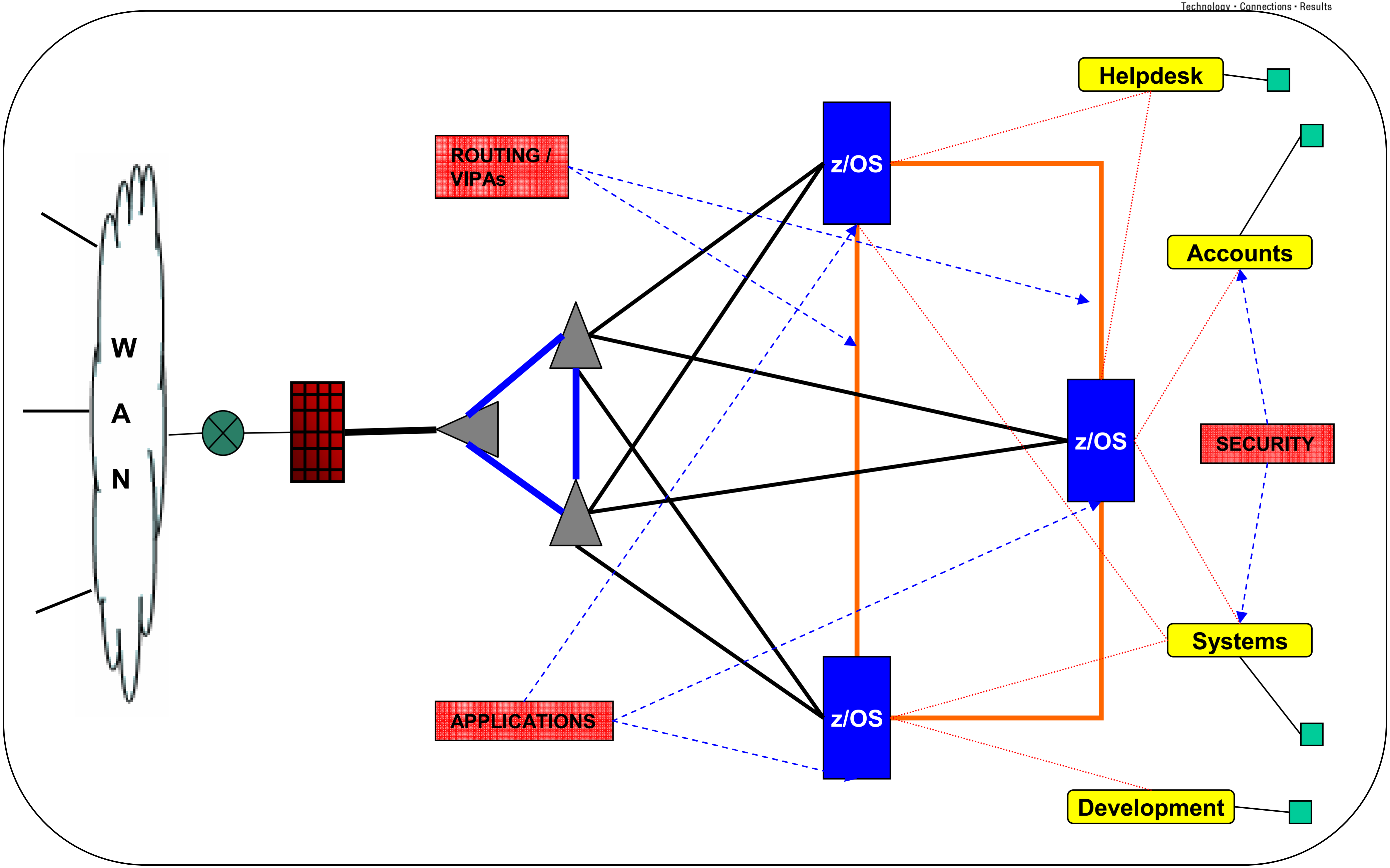
- To monitor routing: z/OS Systems are probably still a mixture of IP and SNA, using CTC, XCF, OSA & MPCIPA connections. **Routing *can* be dynamic**.

- To monitor Base Network elements may not be dynamic, ***but Applications may be***:
    - Application = Service
    - VIPA = Dynamic Application
    - workload management…

  *(Where are your services running?)*
- To monitor Application Performance

- To ensure Internal Security

# Sysplex Risk Areas



# Sysplex Risk Areas

**Possible cause of problems:**

- Hardware Failure
- Application Failure
- Routing / Path Changes
- Unwanted / Unexpected internal traffic (other protocols)
- Buffer Shortages
- IP Stack Resources Shortages
- Configuration Changes (switches)
- Spanning Tree Problems
- Duplicate (Important) IP Addressing
- Illegal Access to Resources (e.g. FTP)

# So, Why Monitor IP?

## Because –

- IP is a critical component of z/OS
- External IP monitoring *does not* understand z/OS

  (*but z/OS may understand other systems….?*)
- To Monitor Network Status
- To save downtime costs
- To Diagnose Problems (*and maintain SLAs*!)
- To Plan for the Future (Capacity Planning is essential to ensure that agreed levels of accessibility and performance can be maintained)

*Time-to-resolution is a major cost factor*

# *IP Monitoring Requirements*

*What a Monitor should do for YOU!*

# Requirements of a Monitor

**A Good Monitor Should Provide Information and Support in the Following Areas:-**

- To ensure continuous **AVAILABILITY**
- To ensure the best possible **PERFORMANCE**
- To enable effective **CAPACITY PLANNING**
- To enhance system **SECURITY**
- To assist with **PROBLEM DETERMINATION**

# Requirements: Availability

**Purpose :**

***To ensure critical resources are available…***

- **We Need to Monitor**
  - Current status (up/down)
  - Current usage (connections, packet rates)
  - Sysplex wide availability

- **Typical resources to be monitored**
  - TCPIP Stacks
  - Interfaces (**OSA**, Links, Devices, VIPA, XCF)
  - Services (Ports)
  - Gateways (Local routers)
  - Remote Hosts (Servers, remote routers, clients)
  - Unix System Services

# Requirements: Performance

**Purpose :**

***To maintain service delivery levels by…***

- **Service Delivery Monitoring**
  - Response Times (typically TN3270) (*not PING*!)
  - Network Transit Times (other TCP services)
  - Round-Trip Times (ping)
  - Connection counts
  - Packet/Byte Rates
- **System Resource Monitoring**
  - TCPIP resource consumption (CPU%, CSM, ECSA)
  - Unix System Services (Processes, Memory, Userids)
- **Protocol Monitoring**
  - TCP Events: Retransmissions, Fragmentation
  - Service specific Events: OSPF, Enterprise Extender
  - ICMP Events

# Requirements: Performance

## A Good Monitoring Process Should :-

- Highlight High CPU
- Highlight High Memory Usage
- Highlight (immediately) when any monitored link fails
- Highlight (immediately) when OSPF traffic exceeds limits.
- Know your "**baselines**" !

**e.g. OSPF –**

- Can be a high user of the processor
- Can maintain multiple copies of routing information causing high memory usage.
- Can, when faced with a "bouncing" link, cause updates to "flood" the network while informing all other routers of every link state change.

# Requirements: Capacity Planning

**Purpose :**

***To ensure continued service delivery levels…***

- **Same input data as performance monitoring**
  - Provided by IP monitor
  - Collected over a longer period of time

- **Analysis of archived data**
  - Looking for trends

- **“What if” Analysis**
  - Simulate additional load to judge impact

# Requirements: Security

**Purpose :**

***To ensure integrity of services and data…***

- **Not necessarily the responsibility of an IP monitor**
  - Refer to Security specific tools:
    - Security Server
    - RACF

*(But, of course, the Monitor itself must be secure!)*

- ***But*…IP Monitoring *can* provide added value**
  - Audit trails of activity
  - Detection of secure (SSL/TLS) connections
  - Highlighting new host systems
  - Detection of unusual activity …
    - Denial of service attacks
    - Port Scans
    - Unexpected connections

# Requirements: Problem Determination

**Purpose :**

***To maximize service levels…***

- **Fast detection of potential problems**
  - Background monitoring in real-time
  - Monitoring using both high and low thresholds
  - Highlight what is ***not*** working
- **Hierarchical Views** *(easy navigation)*
  - Drill down to locate failing component quickly
  - Historical information : Ended connections
- **Utilities**
  - To help isolate and fix the problem
- **Automation**
  - To raise additional alerts
  - To automatically fix common problems

# IP Monitoring Issues

*Things To Think About !*

# Issues:
# Real-Time Monitoring

***How quickly are monitored events detected ?***

- **What does "Real Time" mean ?**
  - IP events are detected ***as they occur***
  - Many tools claim real-time – not all deliver

- **Real-Time Monitoring**
  - Required to identify transient problems
  - Required to aid problem determination
    - See problems as they are happening
    - Perform additional diagnostic tests
  - Only approach for
    - **Response time** monitoring
    - Some protocol monitoring
    - Problem determination

# Issues:
# Response Times

## Response Time, NTT & RTT :

- **There is often confusion over what really constitutes Response Times -**
  - True Response Time is the sum of Network Delay + Application Delay
  - "Ping" (ICMP) times do *NOT* represent Application response times
  - Network "Round-trip" time is also insufficient for this protocol

# Issues: Response Times

**Response Time Requires a Request/Response Exchange:**

```
Tn3270                 IP Monitor/          Host
User                   TCPIP Stack          Application

Incoming data  ->          (1)          ->    Data received
                           (2)          <-    Response data
Response data  <-          (3)
TCP ack        ->          (4)
```

**Given this situation the monitor can calculate :-**

time(2) - time(1) = Application Response time
time(4) - time(3) = Network Response time
time(4) - time(1) = Total Response time

**rfc2562**

# Issues:
# Response Times

*NTT – "Network Transit Time":*

**For Applications that do not have a Request & Response exchange, the "best-effort" solution is "Network Transit Times".**

**This is the measurement of just the Network leg that we saw in the previous example:**

time(4) - time(3) in the previous example

# Issues:
# Response Times

*RTT – "Round Trip Time":*

- **Most monitors have this facility, and use "ping" (ICMP) as the tool.**
- **Valid when used to prove that a network connection exists.**
- **A valid indication as to the state of the network.**

# Issues:
# Response Times

## *RTT – "Round Trip Time" (cont):*

**However,** This is ***NOT*** an indication of **application** response because:

- ICMP may take a different network path (nb. "CoS")
- ICMP may ***not*** be permitted to flow past firewalls
- ICMP answered by lower levels ; "packet turn-around"
- ICMP packets are small and unrepresentative
- "Ping" must be repeated ………

***Consider - Accuracy ? Network load?***

# Issues:
# Polled or Event Driven

***How is monitoring data extracted from system ?:***

Dictates performance and scalability

- **Polled : Monitor asks system for data**
  - *Cannot* be real-time
  - User decides event frequency :-
    - **High : Close to real-time but high resource usage**
    - **Low : Loss of detail, but lower resource usage**
    - **On request : Good for display purposes only**
  - Size of network impacts resource usage
  - Security Policy – is the requestor port allowed?
- *However, there are cases where this can be justified:*
  - Gathering/monitoring information via SNMP (e.g. **OSA**, neighbourhood routers)
  - Under controlled circumstances (reduced workload)
  - For specific diagnostic purposes

# Issues:
# Polled or Event Driven

***How is monitoring data extracted from system ?:***

- **Event Driven : System supplies the monitor with data**
  - True “Real-time” monitoring
  - System decides event frequency
    - **High : Increased resource usage**
    - **Low : Reduced resource usage**
  - Size of network has less impact on resource usage
  - Where practical, always the preferred method

# Issues:
# Usability (1)

***How easy is the monitor to set up, maintain and use ?:***

- ***Does it . . .***
  - **Have "Plug and play" configuration ?**
    - Dynamic detection of network changes
  - **Display or Monitor ?**
  - **Have Sysplex wide monitoring** ?
    - Monitor multiple stacks / multiple LPARS
    - Resource availability ?
  - **Interface with other management tools ?**
  - **Have a Range of end user interfaces ?**
    - GUI and/or 3270 ?  NETVIEW ?

# Issues: Usability (2)

***How easy is the monitor to set up, maintain and use ?:***

- ***Does it . . .***
  - **Have Alert management**
    - Concentrate on what is important
    - Remove fixed problems from alert list
  - **Know When to Alert...?**
    - Must be a user decision
    - Based on local requirements and network specific thresholds
    - Thresholds setup can take a ***long*** time; *is this automated?*

# Issues: Scalability

***How much data can the monitor cope with ?:***

- **You may need to monitor :**
  - Growing number of new services
  - Potentially 10,000s concurrent connections
  - Very high TCP connection rates (WebSphere, DB2)
  - Very high UDP activity (Enterprise Extender)

- **You  may need to provide :**
  - High speed data collection
  - High speed data analysis
  - Powerful filtering of collected data for ease of reading

# Issues: Scalability

***How much data can the monitor cope with ?:***

- **You may be impacted by techniques employed :**
  - Can the collector keep up?
  - Loss of data? (buffer transfer)
  - Can you access the data during periods of network outage?
  - Does the act of data collection and reporting impact the network?

# IP Monitoring Techniques

*The Art of Monitoring!*

# Techniques

***In order to be Pro-Active, we need the right facilities :***

- **The best Methods of Data Collection** to make sure you have all the information
- **The best Presentation of the data** to make sure you see the important events
- **... and a timely Alerting system** to make sure you see problems in time!

# Techniques: Netstat Command

***The Standard TCP/IP Command Interface for Monitoring***

- **Good source of information on active resources**
- **High volumes of detailed information available**
- **Key Issues**
  - Have to poll for information
  - Limited to active connections
  - Limited information on non-TCP activity
  - Limited filtering capabilities
  - No application programming interface
    - Force to “screen scrape”
  - Scalability: **impact on performance**
    *(load increases with number of connections)*

# Techniques: Netstat Command

ALL CURRENT CONNECTIONS

```
netstat -b
MVS TCP/IP onetstat CS V2R10       TCPIP Name: TCPIP          04:08:37
09/20/2004          MVS TCP/IP Real Time Network Monitor
User Id  B Out      B In       L Port  Foreign Socket        State
-------- -----      ----       ------  --------------        -----
BPXOINIT 0000000000 0000000000 10007   0.0.0.0..0            Listen
EXIV40
FTPD1
IMPLEX
TCPIP
TCPIP
TCPIP
```

ALL DEVICES & LINKS

```
netstat -d
MVS TCP/IP onetstat CS V2R10       TCPIP Name: TCPIP          04:10:12
DevName: VIPA                 DevType: VIPA      DevNum: 0000
  DevStatus: Ready
  LnkName: VIPALINK             LnkType: VIPA        LnkStatus: Ready
NetNum: 0   QueSize: 0
BytesIn: 0000000000      BytesOut: 0000000000
BSD Ro
MTU S
DestA
Packe
Proto
SrcPo
IpAdd
Multi
Multi
```

ALL TELNET CONNECTIONS

```
netstat -t
MVS TCP/IP onetstat CS V2R10     TCPIP Name: TCPIP            04:11:22
Internal Telnet Server Status:
Conn     Foreign Socket      State    BytesIn  BytesOut ApplName LuName
----     --------------      -----    -------  -------- -------- ------
000067DB 192.168.21.13..1145 Establsh 0027629  3086794  A16TSO01 P16TCP01
000067DC 192.168.21.13..1146 Establsh 0000032  0001597           P16TCP02
000067DD 192.168.21.13..1147 Establsh 0000032  0001597           P16TCP03
000067DE 192.168.21.13..1148 Establsh 0000032  0001597           P16TCP04
000067DF 192.168.21.13..1149 Establsh 0000560  0028185  IPXP16   P16TCP05
00006834 192.168.5.234..1119 Establsh 0025980  0925471  A16TSO02 P16TCP06
000068CE 192.168.1.57..3098  Establsh 0002035  0104279  A16TSO03 P16TCP07
000068D7 192.168.1.57..3099  Establsh 0000467  0017284  IPXP16   P16TCP08
```

# Techniques:
# SMF Exits

## *The Development of Exit Routines to Intercept SMF Data*

- **Good source for resource and statistical data**
- **Event driven – no polling required**
- **Record Type 118**
  - Connection start/stop
  - Specific Telnet/FTP activities
  - TCP and IP statistics
- **Record Type 119**
  - Duplicates data in 118 records
  - Additional data for UDP, Ports, Interfaces
- **Issues**
  - Performance with event based records
  - May need multiple SMF exits
  - Keep or delete records? – more overhead!
  - **NOT real-time!** (“*close, but no cigar*”)

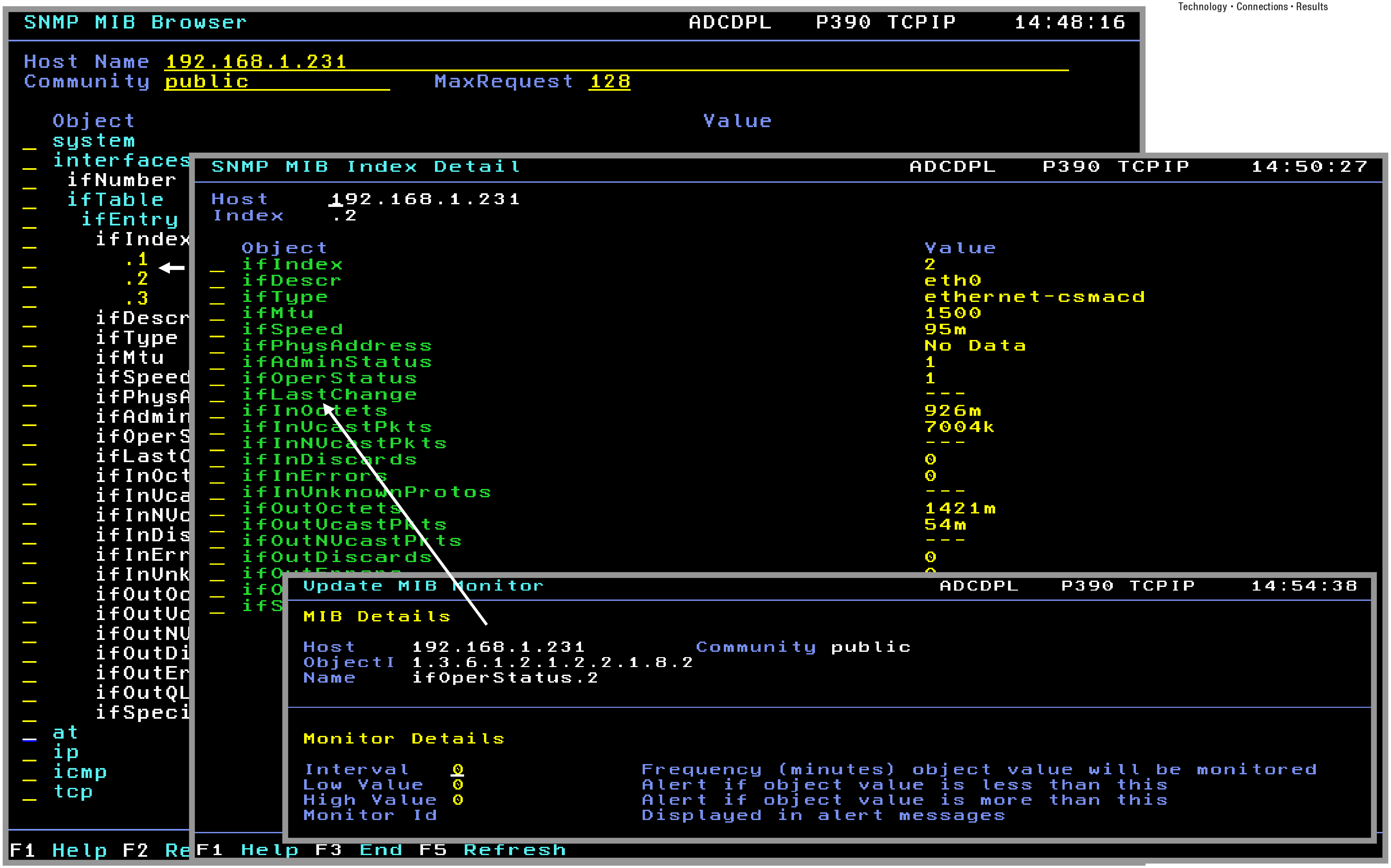
# Techniques:
# SNMP

## *Configure and Activate z/OS SNMP Components*

- **High volumes of useful data**
- **Industry standard MIBs available (RFCs)**
  - System, TCP, UDP, ICMP, SNMP statistics
- **z/OS specific MIBs available**
  - **OSA** **(***MIB Browsers can be very useful tools* *****)**
  - Additional connection information
- **Access to external data**
  - **OSA**, CIP, Servers, routers …
- **Distributed Protocol Interface (DPI) Support** (rfc 1592)
  - Used by zOS itself for TCPIP MIBs
  - Agent/Subagent structure (snmpGet, snmpConnect…)

*more …..*

# Techniques: SNMP



```
SNMP MIB Browser                                    ADCDPL   P390 TCPIP    14:48:16

Host Name 192.168.1.231
Community public                MaxRequest 128

    Object                                           Value
_   system
_   interfaces
_    ifNumber
_    ifTable
_     ifEntry
_      ifIndex
_        .1
_        .2
_        .3
_      ifDescr
_      ifType
_      ifMtu
_      ifSpeed
_      ifPhysA
_      ifAdmin
_      ifOperS
_      ifLastC
_      ifInOct
_      ifInUca
_      ifInNUc
_      ifInDis
_      ifInErr
_      ifInUnk
_      ifOutOc
_      ifOutUc
_      ifOutNU
_      ifOutDi
_      ifOutEr
_      ifOutQL
_      ifSpeci
_   at
_   ip
_   icmp
_   tcp

F1 Help F2 Re
```

```
SNMP MIB Index Detail                                ADCDPL   P390 TCPIP    14:50:27

Host     192.168.1.231
Index    .2

   Object                                            Value
_  ifIndex                                           2
_  ifDescr                                           eth0
_  ifType                                            ethernet-csmacd
_  ifMtu                                             1500
_  ifSpeed                                           95m
_  ifPhysAddress                                     No Data
_  ifAdminStatus                                     1
_  ifOperStatus                                      1
_  ifLastChange                                      ---
_  ifInOctets                                        926m
_  ifInUcastPkts                                     7004k
_  ifInNUcastPkts                                    ---
_  ifInDiscards                                      0
_  ifInErrors                                        0
_  ifInUnknownProtos                                 ---
_  ifOutOctets                                       1421m
_  ifOutUcastPkts                                    54m
_  ifOutNUcastPkts                                   ---
_  ifOutDiscards                                     0
_  ifO                                               0
_  ifO
_  ifS

F1 Help F3 End F5 Refresh
```

```
Update MIB Monitor                                   ADCDPL   P390 TCPIP    14:54:38

MIB Details

Host     192.168.1.231          Community public
ObjectI  1.3.6.1.2.1.2.2.1.8.2
Name     ifOperStatus.2

Monitor Details

Interval   0      Frequency (minutes) object value will be monitored
Low Value  0      Alert if object value is less than this
High Value 0      Alert if object value is more than this
Monitor Id        Displayed in alert messages
```

# Techniques:
# SNMP

## SNMP Issues:

- **Have to poll for information – *not* real time**
- **You need to know the Data Structure**
- **There is a UDP overhead to extract data**
  - Multiple "gets" can be required
  - DPI introduces additional overhead
- **Requires SNMP (server) to be active on z/OS**
- **Limited to active connections**
- **IP network must be available for it to work**
- **Security Policy - SNMP exposes the host, may *not* be allowed!**
- **Overhead – adds network traffic**

# Techniques:
# TCPIP/USS API Calls

## *Early Development of Code to Drive the Program Interfaces*

- **Direct calls to TCPIP/USS via APIs**
- **High speed**
- **USS based APIs are good for some performance data**
- **Good for supplementary monitoring information**

- **Issues**
  - Have to poll for information
  - Very limited functionality provided by TCPIP itself

- ***HOWEVER, From Comm. Server V1.5 (PTF on V1.4)***
  - *New APIs (APAR PQ7724) are much better*
  - *Event driven*

# Techniques: TCPIP/USS API Calls

***The New IBM (TCP/IP) APIs provide:***

- **Access to TCP/IP packet and data trace buffers in "Real-Time" (*), as trace data is collected** ***(collected records need formatting)***
- **Activation and Deactivation Events for TCP connections** ***(SMF 119 images)***
- **Event information for FTP and TN3270 clients and servers** ***(SMF 119 images)***
- **Enterprise Extender statistics**
- **Monitors activities for TCP connections & UDP endpoints**
- **TCP/IP storage usage**

** This is **may** only be Real-Time with regard to collection !*

*more .....*

# Techniques: TCPIP/USS API Calls

- **Event Driven APIs**
  - Data saved in 64K buffers
  - Monitor connects to TCPIP using AF_UNIX socket
  - TCPIP sends token when buffer full (or timer expires)
  - Close (enough?) to real-time (delay whilst buffering)
  - Monitor must call IBM routine to get copy of 64K buffer
  - Good for perf. & protocol monitoring and problem diags.

- **Things to consider**
  - High volume of Packet trace/connection data
  - Monitor must be able to copy data fast enough
  - More data available – powerful filtering needed
  - IBM can overwrite 64K buffers - loss of monitor data
  - CPU utilisation of monitor . . . ?
  - Monitor does not control packet tracing level

***This is still an operator command***

# Techniques: TCPIP/USS API Calls

***What is meant by "Real time" in this context? …***

- **Often defined as the ability to capture packets**
- **Often using the IBM Packet Trace buffers**
- **However, capturing and processing are different things:**
  - Failure to report Errors/Attacks/Changes in time can render the information useless
  - Using capture buffers may result in a data overrun / data loss!
- **True "Real-Time" processing means:**
  - The packets are processed as they traverses the IP stack
  - Buffering is not required
  - There is **NO** delay in processing the data, **NO** buffer overhead, **NO** storage overhead, and **NO** loss of data.

# Techniques: TCPIP/USS API Calls

***Why is "real time" important here? …***

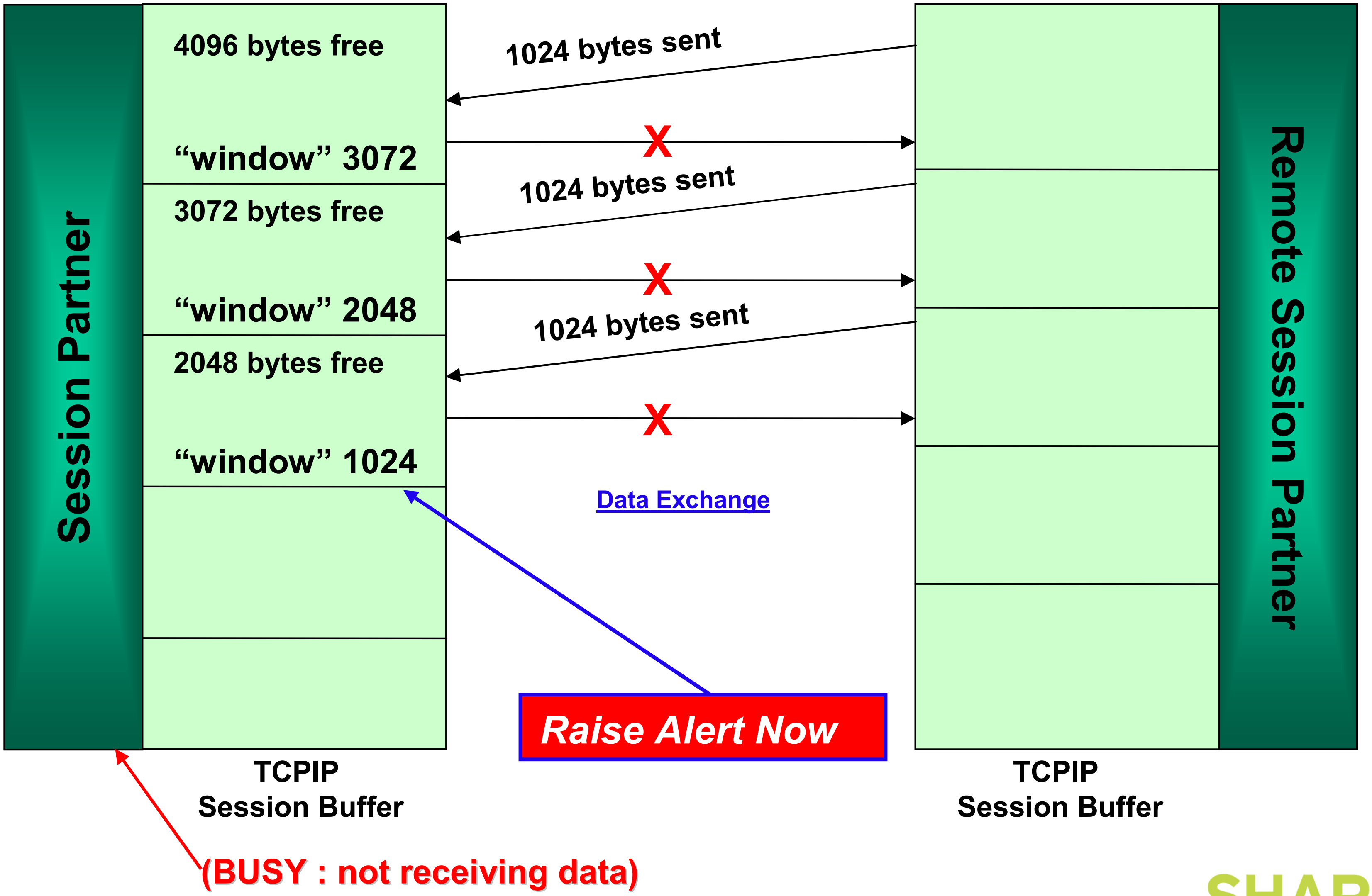
- **Required for all transient problems**
- **Required for problem diagnosis**
- **Required for true Response Time Monitoring**
- **Required for some protocol issues**
  (eg. Retransmissions, Fragmentation, Window Size*)
- **Required for Scalability**

*( * see following example…)*

WILLIAM DATA SYSTEMS

# Techniques: TCPIP/USS API Calls

SHARE Technology • Connections • Results



SHARE in Anaheim 2011

# IP Monitoring: Conclusions

- **IP Monitoring Requirements**
  - Easy to define and understand
  - Not so easy to achieve with standard tools
  - CS since V1.5 has addressed some of the issues
- **IP Monitoring Issues**
  - "Real-time" or not "Real-time"?
  - Polling vs Event driven data collection
  - Usability
  - Performance and Scalability
- **IP Monitoring Techniques**
  - No single (usable, scalable) source for all data
- **Effective Monitoring**
  - Can only be achieved using multiple techniques
  - "Real-time" is mandatory for some requirements
  - Performance and scalability must be considered
  - Usability must be considered

# *Thank you !*