

# A journey through the layers of EE

## How to translate VTAM messages into IP talk

Matthias Burkhard

IBM

[mburkhar@de.ibm.com](mailto:mburkhar@de.ibm.com)



Twitter

: [mreede](#)



SNA Wizards

[sna.wizards@groups.facebook.com](mailto:sna.wizards@groups.facebook.com)

Monday, August 8, 2011: 4:30 PM-5:45 PM

Session 9243 – Europe 10

# Session Content

When it is not enough to present a set of VTAM messages to your network provider to solve a HPR problem in the IP network, the time has come to learn a second language.

Join this session to learn how

```
IST1494I PATH SWITCH STARTED FOR RTP CNR00062 TO netid.cpname  
IST1818I PATH SWITCH REASON: SHORT REQUEST RETRY LIMIT EXHAUSTED
```

translates into the '[4 and a half UDP Firewall Filter Rule](#)' problem.

Come and understand why a [High](#) Performance Routing (HPR) pipe is sometimes performing [Low](#) and is still not using LPR protocol.

Get to know the underlying architecture of HPR: RTP, ANR, ARB

Walk with us through the layers of an EE packet and say Hello to all the bits and bytes that you better call a friend from now on. It will speed up problem resolution in heterogenous networks as most problems that result in nasty VTAM messages are not to be solved within z/OS!

# EE problem categories

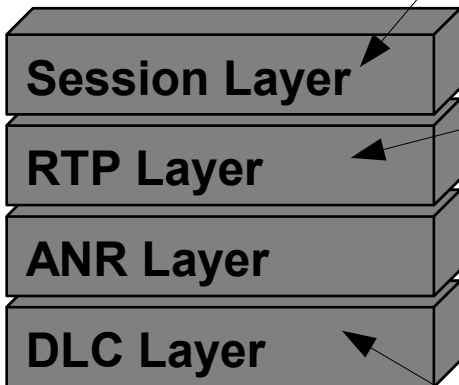
## The external symptoms of EE problems

- Connectivity issues
  - Links don't set up / Links INOP
- Session hang problems
  - Sessions don't setup (PSESST, PBIPLUBF)
  - Sessions hang X-Clock
  - Sessions don't terminate (PSESSEND)
- HPR PATHSWITCH
  - SRQ Retry Limit Exhausted
- Performance Problems
  - Slowdown and Retransmissions
  - GPU utilisation



# VTAM messages

## The layers that trigger them



```
IST1097I  CP-CP SESSION WITH APPN.AS400NN      TERMINATED
IST1280I  SESSION TYPE = CONLOSER  - SENSE = 80020000
IST314I   END
```

```
IST1494I  PATH SWITCH STARTED FOR RTP CNR0F621 TO APPN.AS40
IST1818I  PATH SWITCH REASON: SHORT REQUEST RETRY LIMIT EXH

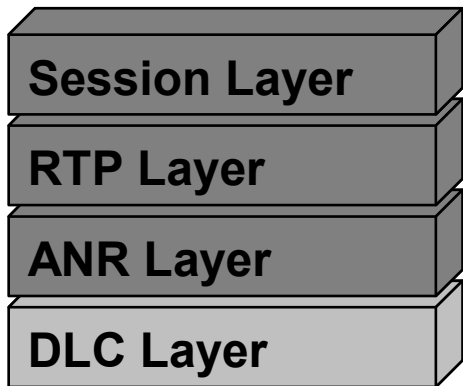
IST1494I  PATH SWITCH FAILED FOR RTP CNR0F621 TO APPN.AS400
IST1495I  NO ALTERNATE ROUTE AVAILABLE
```

```
IST1411I  INOP GENERATED FOR E000000B
IST1430I  REASON FOR INOP IS XID OR LDLC COMMAND TIMEOUT
IST314I   END
IST1141I  SOFT INOP FOR LCBCP1 OVERRIDDEN BY SOFT INOP
```

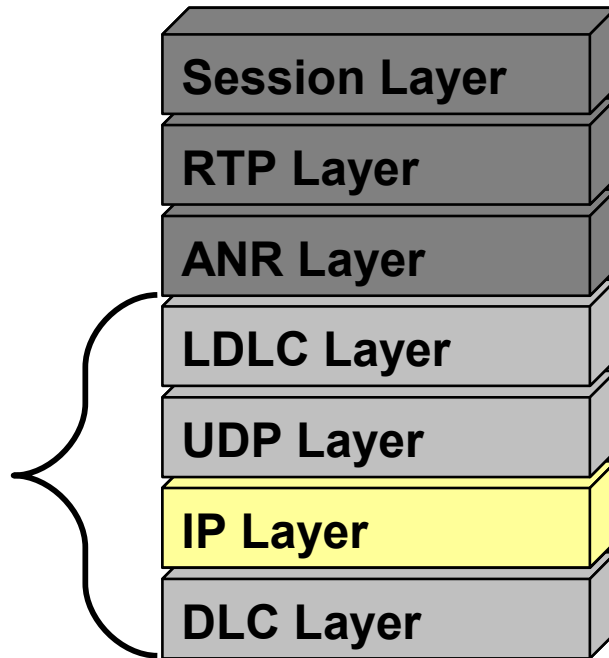
# Enterprise Extender

## Just another HPR-only APPN DLC type?

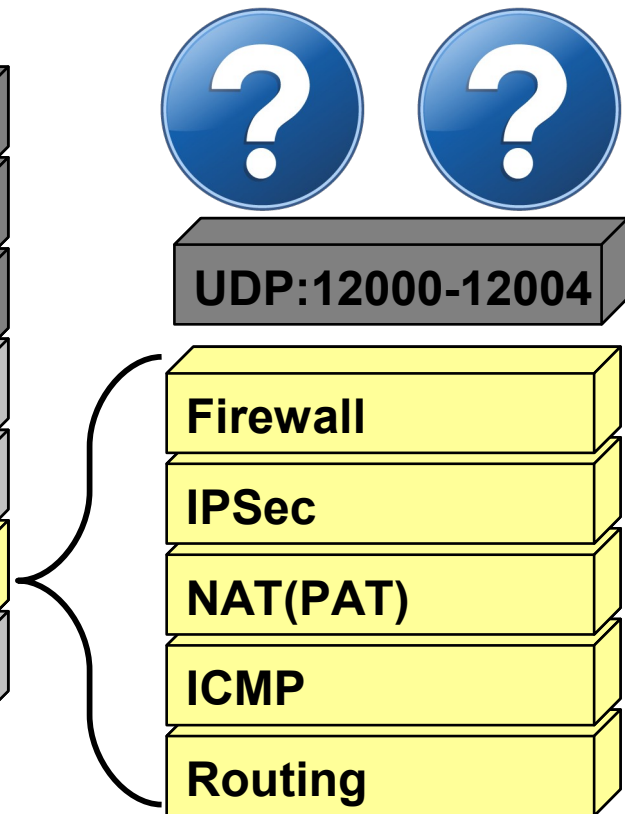
### SNA view



### EE view



### IP -view



# HPR Pipe Display – Lots of Information

```
IST2178I RPNCB ADDRESS 1CA30018
IST1963I APPNCOS = #INTER - PRIORITY = HIGH
IST1476I TCID X'269E31C3000100AC' - REMOTE TCID X'0C18BC6900010098'
IST1481I DESTINATION CP xxxx.MV019 - NCE X'D000000000000000'
IST1587I ORIGIN NCE X'D000000000000000'
IST1966I ACTIVATED AS ACTIVE ON 07/31/11 AT 10:55:19
IST1479I RTP CONNECTION STATE = CONNECTED, BACKPRESSURE - MNPS = NO
IST1959I DATA FLOW STATE = NORMAL
```

Session

RTP Layer

ANR Layer

DLC Layer

```
IST1968I ARB INFORMATION
IST1844I ARB MODE = YELLOW
IST1697I RTP PACING ALGORITHM = ARB RESPONSIVE MODE
IST1477I ALLOWED DATA FLOW RATE = 258 KBITS/SEC
IST1516I INITIAL DATA FLOW RATE = 500 KBITS/SEC
IST1841I ACTUAL DATA FLOW RATE = 266 KBITS/SEC
IST1969I MAXIMUM ACTUAL DATA FLOW RATE = 32 MBITS/SEC
IST1862I ARB MAXIMUM SEND RATE = 16 MBITS/SEC
IST1846I CURRENT RECEIVER THRESHOLD = 395975 MICROSECONDS
IST1846I MAXIMUM RECEIVER THRESHOLD = 417000 MICROSECONDS
IST1846I MINIMUM RECEIVER THRESHOLD = 185000 MICROSECONDS
IST1970I RATE REDUCTIONS DUE TO RETRANSMISSIONS = 11660
IST924I -----
```



# HPR Pipe Display – Lots of Information

```
IST1973I OUTBOUND TRANSMISSION INFORMATION:
IST1974I NUMBER OF NLPS SENT =                2354126 ( 2M )
IST1975I TOTAL BYTES SENT =                   1899005183 ( 1G )
IST1849I LARGEST NLP SENT =                   1319 BYTES
IST1980I SEQUENCE NUMBER = 1773745903 (X'69B936EF')
IST1842I NUMBER OF NLPS RETRANSMITTED =       35283
IST2249I NLP RETRANSMIT RATE = 1.4987%
IST2236I LAST NLP RETRANSMITTED ON 08/02/11 AT 11:14:18
IST1976I BYTES RETRANSMITTED =                43056398 ( 43M )
IST1478I NUMBER OF UNACKNOWLEDGED BUFFERS =    49
IST1958I NUMBER OF ORPHANED BUFFERS =          0
IST1843I NUMBER OF NLPS ON WAITING-TO-SEND QUEUE = 50
IST1847I NUMBER OF NLPS ON WAITING-FOR-ACKNOWLEDGEMENT QUEUE = 15
IST2268I NUMBER OF BYTES ON WAITING-FOR-ACK QUEUE = 37324
IST1977I MAXIMUM NUMBER OF NLPS ON WAITING-FOR-ACK QUEUE = 140
IST2269I MAXIMUM NUMBER OF BYTES ON WAITING-FOR-ACK QUEUE = 575540
IST1978I WAITING-FOR-ACK QUEUE MAX REACHED ON 08/02/11 AT 10:22:57
IST2085I NUMBER OF NLPS ON OUTBOUND WORK QUEUE = 1
IST2086I MAXIMUM NUMBER OF NLPS ON OUTBOUND WORK QUEUE = 66
```

Session

RTP Lay

ANR Lay

DLC Lay



**IBM**®

# EE problem categories

## The #1 root cause of EE problems

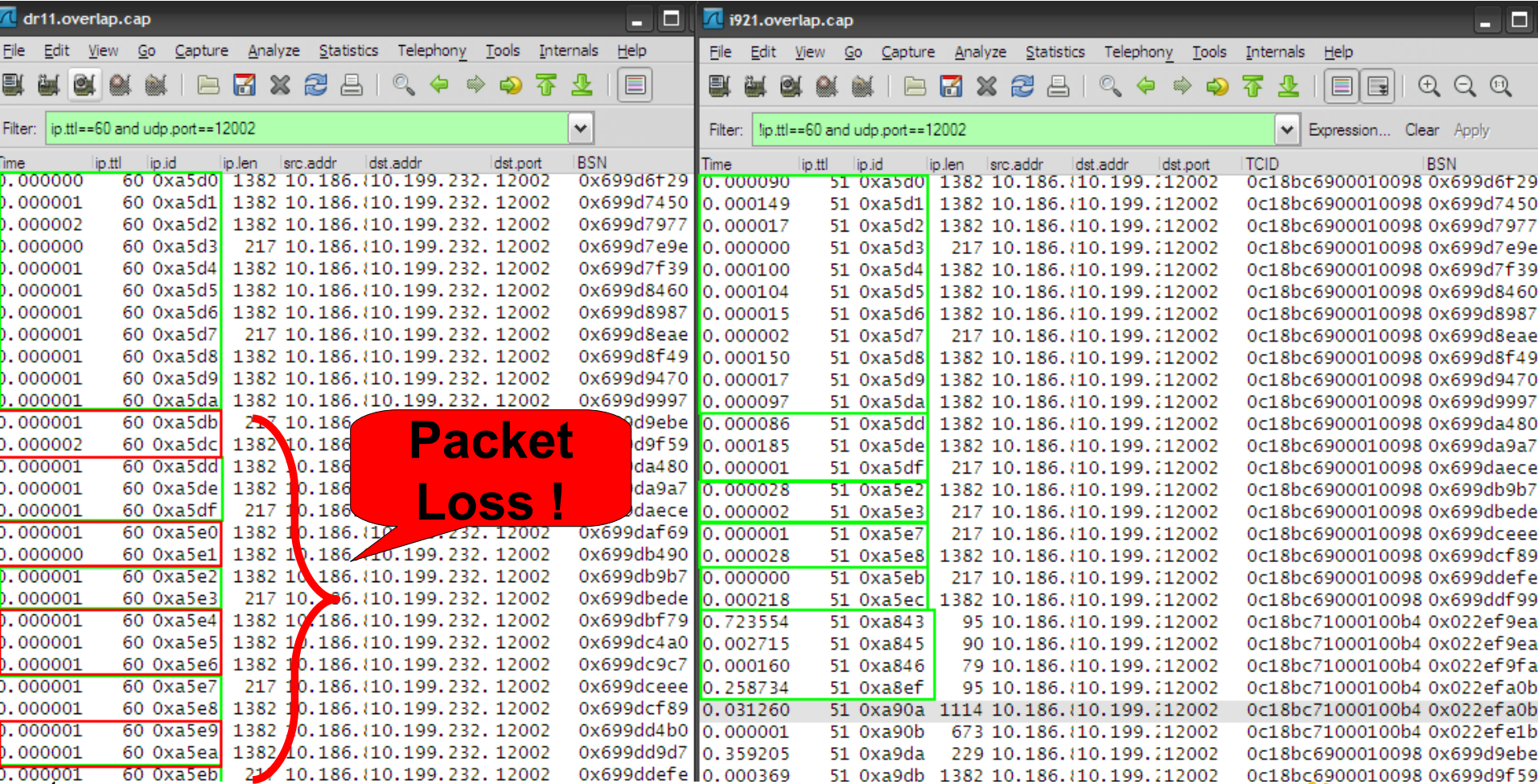
- Connectivity issues
  - IP routing issues
  - FW filter rules
- Session hang problems
  - IP Fragmentation
  - IPSec tunnels
- HPR PATHSWITCH
  - FW connection tables / FW filter rules
- Performance
  - Retransmissions
  - Slowdowns , delays, congestion



**Packet Loss !**



# Packet loss – Use ip.id to track a packet in traces at source and destination



Filter: ip.ttl==60 and udp.port==12002

Time	ip.ttl	ip.id	ip.len	src.addr	dst.addr	dst.port	BSN
0.000000	60	0xa5d0	1382	10.186.110.199	10.199.232.12002	12002	0x699d6f29
0.000001	60	0xa5d1	1382	10.186.110.199	10.199.232.12002	12002	0x699d7450
0.000002	60	0xa5d2	1382	10.186.110.199	10.199.232.12002	12002	0x699d7977
0.000000	60	0xa5d3	217	10.186.110.199	10.199.232.12002	12002	0x699d7e9e
0.000001	60	0xa5d4	1382	10.186.110.199	10.199.232.12002	12002	0x699d7f39
0.000001	60	0xa5d5	1382	10.186.110.199	10.199.232.12002	12002	0x699d8460
0.000001	60	0xa5d6	1382	10.186.110.199	10.199.232.12002	12002	0x699d8987
0.000001	60	0xa5d7	217	10.186.110.199	10.199.232.12002	12002	0x699d8eae
0.000001	60	0xa5d8	1382	10.186.110.199	10.199.232.12002	12002	0x699d8f49
0.000001	60	0xa5d9	1382	10.186.110.199	10.199.232.12002	12002	0x699d9470
0.000001	60	0xa5da	1382	10.186.110.199	10.199.232.12002	12002	0x699d9997
0.000001	60	0xa5db	217	10.186.110.199	10.199.232.12002	12002	0x699da9be
0.000002	60	0xa5dc	1382	10.186.110.199	10.199.232.12002	12002	0x699da959
0.000001	60	0xa5dd	1382	10.186.110.199	10.199.232.12002	12002	0x699da480
0.000001	60	0xa5de	1382	10.186.110.199	10.199.232.12002	12002	0x699da9a7
0.000001	60	0xa5df	217	10.186.110.199	10.199.232.12002	12002	0x699daece
0.000001	60	0xa5e0	1382	10.186.110.199	10.199.232.12002	12002	0x699daf69
0.000000	60	0xa5e1	1382	10.186.110.199	10.199.232.12002	12002	0x699db490
0.000001	60	0xa5e2	1382	10.186.110.199	10.199.232.12002	12002	0x699db9b7
0.000001	60	0xa5e3	217	10.186.110.199	10.199.232.12002	12002	0x699dbede
0.000001	60	0xa5e4	1382	10.186.110.199	10.199.232.12002	12002	0x699dbf79
0.000001	60	0xa5e5	1382	10.186.110.199	10.199.232.12002	12002	0x699dc4a0
0.000001	60	0xa5e6	1382	10.186.110.199	10.199.232.12002	12002	0x699dc9c7
0.000001	60	0xa5e7	217	10.186.110.199	10.199.232.12002	12002	0x699dcfee
0.000001	60	0xa5e8	1382	10.186.110.199	10.199.232.12002	12002	0x699dce89
0.000001	60	0xa5e9	1382	10.186.110.199	10.199.232.12002	12002	0x699dd4b0
0.000001	60	0xa5ea	1382	10.186.110.199	10.199.232.12002	12002	0x699dd9d7
0.000001	60	0xa5eb	217	10.186.110.199	10.199.232.12002	12002	0x699ddefe

Filter: ip.ttl==60 and udp.port==12002

Time	ip.ttl	ip.id	ip.len	src.addr	dst.addr	dst.port	TCID	BSN
0.000090	51	0xa5d0	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d6f29
0.000149	51	0xa5d1	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d7450
0.000017	51	0xa5d2	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d7977
0.000000	51	0xa5d3	217	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d7e9e
0.000100	51	0xa5d4	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d7f39
0.000104	51	0xa5d5	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d8460
0.000015	51	0xa5d6	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d8987
0.000002	51	0xa5d7	217	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d8eae
0.000150	51	0xa5d8	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d8f49
0.000017	51	0xa5d9	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d9470
0.000097	51	0xa5da	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d9997
0.000086	51	0xa5dd	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699da480
0.000185	51	0xa5de	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699da9a7
0.000001	51	0xa5df	217	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699daece
0.000028	51	0xa5e2	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699db9b7
0.000002	51	0xa5e3	217	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699dbede
0.000001	51	0xa5e7	217	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699dcfee
0.000028	51	0xa5e8	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699dce89
0.000000	51	0xa5eb	217	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699ddefe
0.000218	51	0xa5ec	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699ddf99
0.723554	51	0xa843	95	10.186.110.199	10.199.232.12002	12002	0c18bc71000100b4	0x022ef9ea
0.002715	51	0xa845	90	10.186.110.199	10.199.232.12002	12002	0c18bc71000100b4	0x022ef9ea
0.000160	51	0xa846	79	10.186.110.199	10.199.232.12002	12002	0c18bc71000100b4	0x022ef9fa
0.258734	51	0xa8ef	95	10.186.110.199	10.199.232.12002	12002	0c18bc71000100b4	0x022efa0b
0.031260	51	0xa90a	1114	10.186.110.199	10.199.232.12002	12002	0c18bc71000100b4	0x022efa0b
0.000001	51	0xa90b	673	10.186.110.199	10.199.232.12002	12002	0c18bc71000100b4	0x022efe1b
0.359205	51	0xa9da	229	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d9e9e
0.000369	51	0xa9db	1382	10.186.110.199	10.199.232.12002	12002	0c18bc6900010098	0x699d9f59

Packet Loss!

# EE problems

## The top 3 questions

- Where did my packet get lost ?
  - Locally ( close to the source )
  - Remotely (close to the destination)
  - Somewhere in between ( network, firewalls ...)
- Why was it discarded?
  - Incorrect routing ?
  - Congestion ?
  - Security Policies ?

What exactly *is* the IP packet that you are missing?  
What does it look like?  
Are you sure you really sent it?

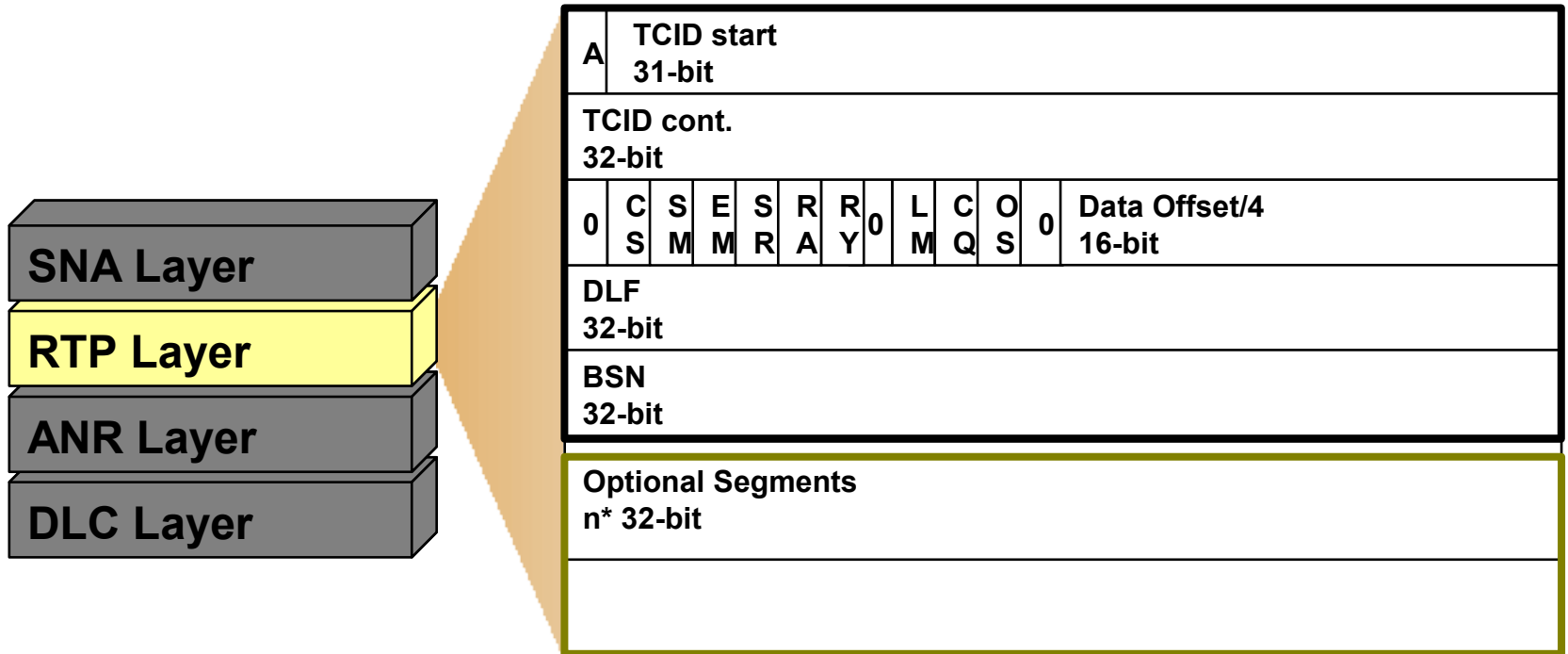
- How can I prove it?
  - VTAM messages?
  - VTAM Traces?



# RTP layer

## Transport Header – Optional Segments

- Where HPR vocabulary lives



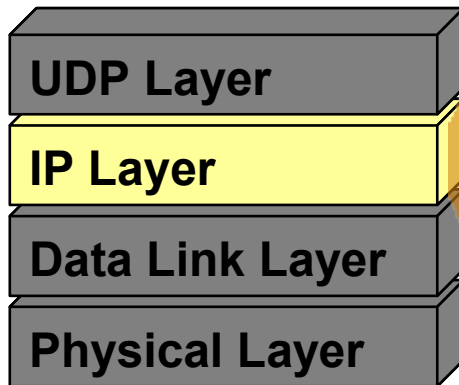
All fields and segments are always a multiple of 4 bytes in length

SETUP: 0D	CIDEX: 10	SWINFO: 14	ARB: 22	STATUS: 0E	COOB: 0F	CFAULT: 12
-----------	-----------	------------	---------	------------	----------	------------

# The IP header

## IP Header and additional protocols

- Where IP vocabulary lives



version 4-bit	header length	3-bit TOS field 5-bit reserved	total length 16-bit		
identification 16-bit			0	D F	M F
time to live TTL 8-bit		protocol 8-bit	fragment offset 13-bit		
header checksum 16-bit			source IP address 32-bit		
destination IP address 32-bit			options  e.g. record route, timestamp, padding, etc. maximum 40 bytes		

- and



# A typical EE packet

## What this session will focus on

- **IP Header**

- Length
- **Identifier**
- **Flags/FragOffs**
- **TTL**
- **IP addresses**

- **ICMP**

- **UDP Header**

- Ports
- Length

- **Checksum**

- **LDLC**

- SAPs
- Control
  - **XID/TEST/UI/DISC**

- **HPR**

- **NHDR**
- **THDR**
- **Optional Segments**

- **SNA PIU (TH,RH,RU)**

- Sense Codes
- **FMH7s**



# IP Header

## IP\_ID and addresses

- **IP Header**
  - TOS precedence
  - Length
  - **Identifier**
  - Flags/FragOffs
  - TTL
  - IP addresses

<b>4</b>	<b>5</b>	3-bit TOS field 5-bit reserved	total length 16-bit		
identification 16-bit		0	D F	M F	fragment offset 13-bit
time to live TTL 8-bit		protocol 8-bit		header checksum 16-bit	
source IP address 32-bit					
destination IP address 32-bit					

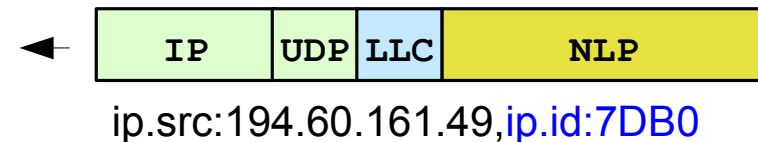
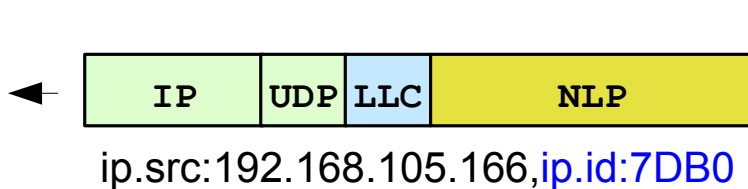
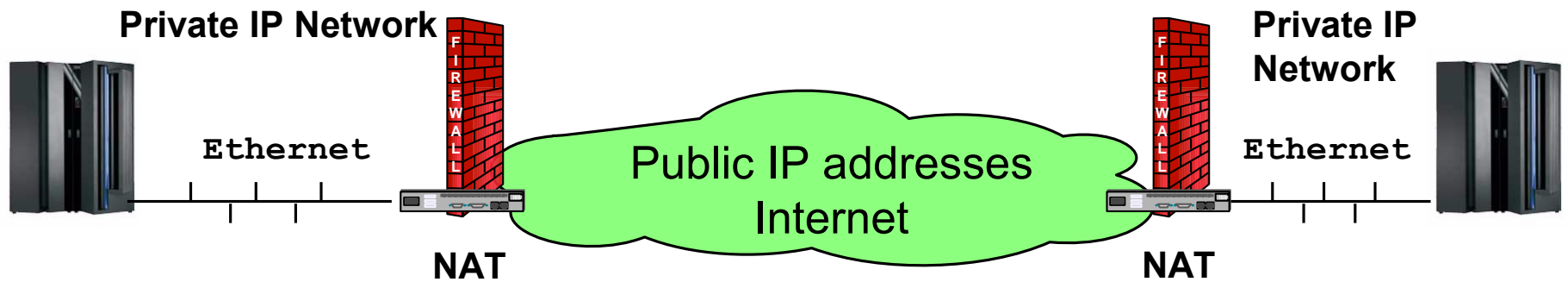
- **IP\_ID is a 2 byte field incremented with every new packet generated at the sending stack (unique until it wraps)**
- **IP addresses: in IPV4 (header 45) 4 byte long**
- **Can be changed by NAT devices in flight**



# IP Identifier

## Following a packet through the network

- NAT will change the addresses – the IP\_ID will stay the same



Here it is: IP.ID  
7DB0!

Where is my  
IP.ID 7DB0!





# IP Header

## IP Identifier

- Required to track down a packet
- Will not change in flight

<b>4</b>	<b>5</b>	3-bit TOS field 5-bit reserved	total length 16-bit			
identification 16-bit			0	D F	M F	fragment offset 13-bit
time to live TTL 8-bit		protocol 8-bit	header checksum 16-bit			
source IP address 32-bit						
destination IP address 32-bit						

```

                IPID          10.186.86.243 --> 10.217.18.47
IP      45C00058 9C730000 3C1162AD 0ABA56F3 0AD9122F
UDP     2EE12EE1 0044D9DF
LLC     040403
NLP     C608 80 FF00
RTP     000000000100257B0004000D00000000000000E2CD
        03225904000000000000000000000000
        050E0000000400020000CF2F00000000000000000000000000
    
```

ARB REPLY  
STATUS





# IP Header

## IP length and Fragmentation

- **IP Header**
  - TOS precedence
  - **Length**
  - Identifier
  - **Flags/Frag\_Offs**
  - TTL
  - IP addresses

<b>4</b>	<b>5</b>	3-bit TOS field 5-bit reserved	total length 16-bit	
identification 16-bit		0	D F	M F
time to live TTL 8-bit		protocol 8-bit	fragment offset 13-bit	
header checksum 16-bit		source IP address 32-bit		
destination IP address 32-bit				

- The length indicates how large the IP datagram is
- If the datagram exceeds the MTU size of the weakest link an intermediate router will fragment the packet .
- Reassembly is then done at the destination IP.



# IP Fragmentation

## Two fragments arriving

- IPID is the same
- MF flag is set in 1<sup>st</sup> fragment
- Fragment Offset is > 0
- No protocol header in 2<sup>nd</sup> fragment

<b>4</b>	<b>5</b>	3-bit TOS field 5-bit reserved	total length 16-bit	
identification 16-bit		0	D F	M F
time to live TTL 8-bit		protocol 8-bit		header checksum 16-bit
source IP address 32-bit				
destination IP address 32-bit				

### Fragment

```

IP_V4 Header
  lgth ID , -MF
4500 0304 5A48 2000 2F11BA0B 0AC7E821
0ABA56F3
  UDP Header
2EE22EE2 059C8EF5
  Logical Link Control Header
080403
  
```

```

IP_V4 Header
  lgth ID , -- fgmt_offs
4500 02C0 5A48 005E 2F11D9F1 0AC7E821
0ABA56F3
  IP FRAGMENT
40404040 40404040 40404040 40404040
40404040 40404040 40404040 40404040
40404040 40404040 40404040 40404040
  
```

# IP Fragmentation Issues and Path MTU discovery

- Fragmentation increases CPU at receiver
- Fragmentation causes an additional delay
- Fragmentation across Firewall infrastructure often not allowed
  - FW filter rules check on IP@ ,protocol and port numbers
    - 2<sup>nd</sup> fragment does not have port numbers
- Path MTU Discovery (PMTUD) available in VTAM V1R10
  - DF bit is set causing ICMP message from router if fragmentation is required
  - Contains MTU size of next hop

```

IP_V4 Header
      , -DF
45C00058 275D4000 40110000 0A03000C
0A00021C
UDP Header
2EE12EE1 00441680
Logical Link Control Header
C80403
  
```

# IP Fragmentation

## Out of order arrival

- 2nd Fragment of 5001 arriving after 5002
- IP layer will reassemble the 2 fragments (if both arrive...)

Description	TTL	IP Address	<>	IP Address (+ PortN	Iden	Lengt
EE_HIG FID5	48	10.186.86.24	<-	10.199.232.33 (12002	4FFE	300
EE_HIG <b>First Frag</b>	47	10.186.86.24	<-	10.199.232.33 (12002	5001	836
EE_HIG continued	48	10.186.86.24	<-	10.199.232.33 (12002	5002	845
IP/ <b>FRAGMENT Last</b>	47	10.186.86.24	<-	10.199.232.33	5001	768
EE_HIG continued	48	10.186.86.24	<-	10.199.232.33 (12002	5003	1,488
EE_HIG continued	48	10.186.86.24	<-	10.199.232.33 (12002	5004	830
EE_HIG FID5	48	10.186.86.24	<-	10.199.232.33 (12002	5005	1,488

- The upper layer protocol needs to reorder the data
  - Reordering is done in TCP protocol, not in UDP
  - For Enterprise Extender , HPR RTP will perform this function

# IP Header

## TTL - Time To Live

- IP Header**

- TOS precedence
- Length
- Identifier
- Flags/Frag\_Offs
- TTL**
- IP addresses

<b>4</b>	<b>5</b>	3-bit TOS field 5-bit reserved	total length 16-bit		
identification 16-bit		0	D F	M F	fragment offset 13-bit
time to live TTL 8-bit		protocol 8-bit		header checksum 16-bit	
source IP address 32-bit					
destination IP address 32-bit					

- 1 byte field that controls how far an IP packet can travel
- It gets decremented by every router on the path
- When it reaches 1, the router will discard the packet

# IP TTL

## Initial TTL values

- The initial TTL value is configurable in every IP stack
  - Most IP stacks use the default though
    - Why not?

OS Protocol	AIX	z/OS	i5OS	Linux Unix	Win	Routers
TCP	x3C	x40	x40	x40	x80	xFF
UDP	x1E	x40	x40	x40	x80	xFF

- Knowledge of the initial TTL at the source can be used to determine (guess) the operating system and the distance of a remote host (in # of hops)
- An ICMP error message will be sent when a router discards a packet because of an inbound TTL of 1

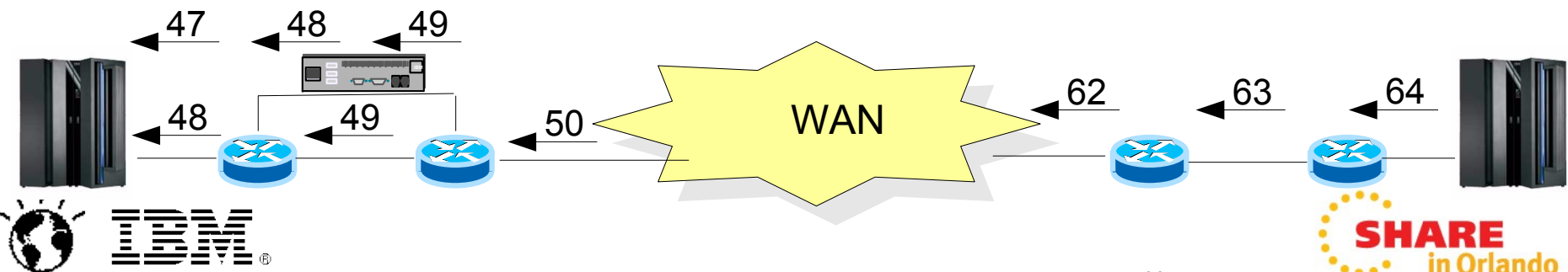
# IP TTL

## Guessing the topology

- Packets arriving with a TTL of 48
  - The sending IP stack is 16 hops away (Initial TTL=64)

Description	TTL	IP Address	<>	IP Address (+ PortN	Iden	Length
EE_HIG FID5	48	10.186.86.24	<-	10.199.232.33 (12002	4FFE	300
EE_HIG <b>First Frag</b>	47	10.186.86.24	<-	10.199.232.33 (12002	<b>5001</b>	836
EE_HIG continued	48	10.186.86.24	<-	10.199.232.33 (12002	5002	845
IP/ <b>FRAGMENT Last</b>	47	10.186.86.24	<-	10.199.232.33	<b>5001</b>	768
EE_HIG continued	48	10.186.86.24	<-	10.199.232.33 (12002	5003	1,488
EE_HIG continued	48	10.186.86.24	<-	10.199.232.33 (12002	5004	830
EE_HIG FID5	48	10.186.86.24	<-	10.199.232.33 (12002	5005	1,488

- However, fragmented packets arrive with a TTL of 47



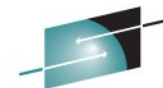
# IP TTL D NET,EEDIAG,TEST=YES

- Sets TTL purposely too short to learn the IP route's RTT

```
D NET,EEDIAG,TEST=YES,LIST=ALL,IPADDR=(10.999.232.65,10.888.86.241)
IST350I  DISPLAY TYPE = EEDIAG
IST2130I  ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
IST2119I  ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE00000B
IST2131I  EEDIAG DISPLAY COMPLETED ON 03/08/11 AT 12:12:58
IST2132I  LDLC PROBE VERSIONS: VTAM = V1                PARTNER = V1
IST1680I  LOCAL  IP ADDRESS 10.999.232.65
IST1680I  REMOTE IP ADDRESS 10.888.86.241
IST924I  -----
IST2133I  INTFNAME: OSAGES5L                INTFTYPE: IPAQENET
IST2134I  CONNECTIVITY SUCCESSFUL                PORT: 12000
IST2137I  1  10.999.999.193                RTT: 1
IST2137I  2  10.88.99.66                    RTT: 0
IST2137I  3  10.333.33.1                    RTT: 0
IST2137I  4  10.333.39.21                    RTT: 12
```



# EEDIAG TEST=YES,LIST=DETAIL



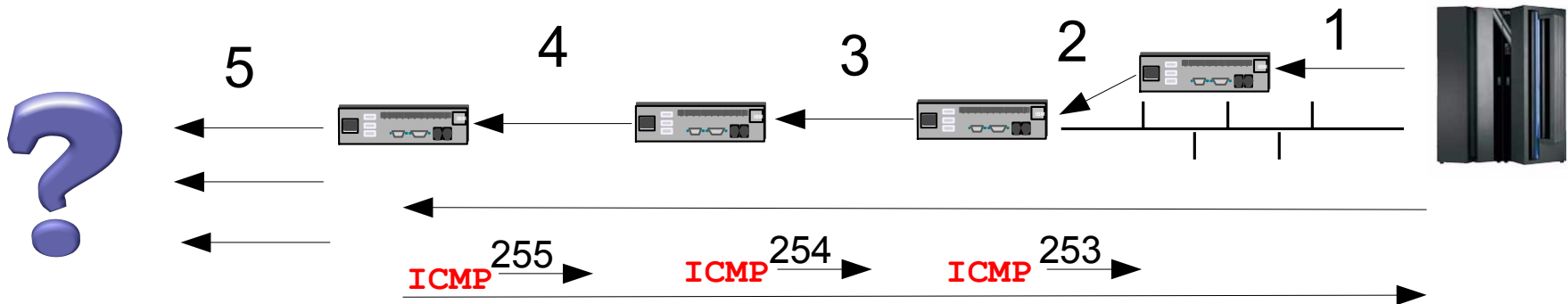
```
D NET ,EEDIAG ,TEST=YES ,LIST=DETAIL ,IPADDR=(10.999.232.65,10.888.86.241)
IST350I  DISPLAY TYPE = EEDIAG
IST2130I  ENTERPRISE EXTENDER CONNECTIVITY TEST INFORMATION
IST2119I  ENTERPRISE EXTENDER DISPLAY CORRELATOR: EE00000B
IST2131I  EEDIAG DISPLAY COMPLETED ON 03/08/11 AT 12:12:58
IST2132I  LDLC PROBE VERSIONS: VTAM = V1          PARTNER = V1
IST1680I  LOCAL  IP ADDRESS 10.999.232.65
IST1680I  REMOTE IP ADDRESS 10.888.86.241
IST924I  -----
IST2133I  INTFNAME: OSAGES5L          INTFTYPE: IPAQENET
IST2134I  CONNECTIVITY SUCCESSFUL          PORT: 12000
IST2137I  1  10.999.999.193          RTT: 1
IST2137I  2  10.88.99.66            RTT: 0
IST2137I  3  10.333.33.1             RTT: 0
IST2137I  4  10.333.39.21           RTT: 12
IST2137I  5  10.333.39.3            RTT: 17
IST2137I  6  10.444.130.1           RTT: 15
IST2137I  7  10.555.1.206           RTT: 18
IST2137I  8  192.666.250.162        RTT: 17
IST2137I  9  10.777.127.186         RTT: 15
IST2137I  10 10.888.86.241          RTT: 13
IST2134I  CONNECTIVITY SUCCESSFUL          PORT: 12001
```



# IP TTL D NET,EEDIAG,TEST=YES Trace

- ICMP TTL Timeout messages required to come back

Delta	Description (short)	TTL	IP Address	Iden <>	IP Address (+ PortNu)
4.668.333	EE_CMD	1	198.238.232.3	<-	198.239.65.137 (12000)
0.000.169	<b>ICMP/TIMEOUT (TTL)</b>	255	147.55.217.3	B7C0 ->	198.239.65.137
0.000.328	EE_CMD	2	198.238.232.3	<-	198.239.65.137 (12000)
0.000.212	<b>ICMP/TIMEOUT (TTL)</b>	255	147.55.215.2	5906 ->	198.239.65.137
0.000.897	EE_CMD	3	198.238.232.3	<-	198.239.65.137 (12000)
0.000.712	<b>ICMP/TIMEOUT (TTL)</b>	254	147.55.215.6	430C ->	198.239.65.137
0.000.017	EE_CMD	4	198.238.232.3	<-	198.239.65.137 (12000)
0.000.878	<b>ICMP/TIMEOUT (TTL)</b>	253	10.2.3.3	3B31 ->	198.239.65.137
0.000.010	EE_CMD	5	198.238.232.3	<-	198.239.65.137 (12000)
<b>3.301.891</b>	EE_CMD	5	198.238.232.3	<-	198.239.65.137 (12000)



# ICMP

## Listen to the network music

- ICMP protocol is used in IP to
  - Test connectivity (PING)
  - Report errors
    - Destination unreachable
    - Time-out conditions
  - Propagate information
    - MTU size of next hop with PMTU Discovery
- ICMP packets are very important in diagnosis
  - Often not allowed through secured infrastructure
    - Firewall rules block ICMP in general
  - Often not traced because of trace filters
    - Source IP address of ICMP packets not predictable

PMTUD (V1R10) depends on receipt of ICMP messages



# ICMP

## Error message

- ICMP protocol type 1
- Contains Error information in the ICMP header
  - 0301 – cannot route any further, host unreachable
- Contains the original IP header that caused this error
  - As seen at the ICMP sender (IPID:B31A, TTL=3A,

```

IN  149.83.5.17 <- 172.17.60.1 ICMP/DESTUNR(03) :Host Unreachable (01)
IP_V4          IP_V4 Header
0000 45000038 B5210000 F9018A2C AC113C01
0010 95530511

ICMP          ICMP Internet Control and Messaging
0000 030156FF 00000000

IP_V4          IP_V4 Header
0000 45C000A0 B31A0000 3A111952 95530511
0010 AC1B6CA1

ICMP_IPFRAG_DATA  ICMP IPFRAG
0000 2EE02EE0 008C47B3
  
```

# A typical EE packet

## What's left ...

- **IP Header**
  - Length
  - **Identifier**
  - **Flags/FragOffs**
  - **TTL**
  - **IP addresses**
- **ICMP**
- **UDP Header**
- **Ports**
- **Length/ Checksum**
- **LDLC**
  - **SAPs**
  - **Control**
    - **XID/TEST/UI/DISC**
- **HPR**
  - **NHDR**
  - **THDR**
  - **Optional Segments**
- **SNA PIU (TH,RH,RU)**
  - **Sense Codes**
  - **FMH7s**

# VTAM

## HPR Pipes – RTP PUs - TCIDs

- VTAM knows a pipe by a PU name in ISTRTPMN
  - Typically starts with CNRxxxxx
- The names are different, depending on the RTP node
  - If the remote RTP is also a VTAM, it will have another CNRxxxxx name
  - If it is a distributed SNA stack, the name will be
    - @Rnnnnnn
- The TCIDs must be used to correlate the display output
  - The 'local TCID' here is the 'remote TCID' at the other end
- Both TCIDs must be used to follow a pipe's traffic in a trace
  - All NLPs carry the receiver's local TCID



# RTP - TCID

## Different Names, same TCIDs

```

IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = RTPS
IST1695I PU NAME      CP NAME      COSNAME SWITCH CONGEST STALL SESS
IST1960I CNR000C4 BROWN.KEN  SNASVCMG NO      NO      NO      2
IST1960I CNR000C3 BROWN.KEN  RSETUP   NO      NO      NO      0
IST1960I CNR000C2 BROWN.KEN  CPSVCMG  NO      NO      NO      2

D NET, ID=CNR000C4, E
IST097I DISPLAY ACCEPTED
IST075I NAME = CNR000C4, TYPE = PU_T2.1 667
IST1043I CP NAME = KEN - CP NETID = BROWN - DYNAMIC LU = YES
IST1962I APPNCOS = SNASVCMG - PRIORITY = NETWORK
IST1476I TCID X'1AAFE05000010119' - REMOTE TCID X'0000000002002F1E'
IST1481I DESTINATION CP BROWN.KEN - NCE X'80'
    
```

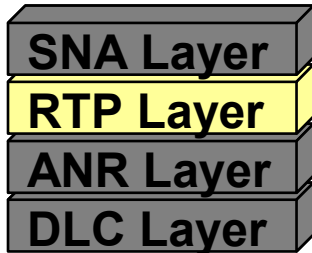
Communications Server-Knotenoperationen - Einzelknotensicht eines lokalen Systems

Operationen Aufrufen Darstellung Fenster Hilfe

Einzelknotensicht eines lokalen Systems

Name	Zieladresse	Ak...	Name der Se...	G...	Emp...	Send...	Maximal...	Lokale TCID	Ferne TCID
@R000001	DEIBMTA...	2	CPSVCMG	8...	828	( 7864	1469	0000000001002F1E	1AAFE04E0001010B
@R000003	DEIBMTA...	2	SNASVCMG	4...	2933	( 200	1469	0000000002002F1E	1AAFE05000010119
@R000002	DEIBMTA...	0	RSETUP	2...	273	( 7864	1469	0000000003002F1E	1AAFE04F0001010C

# RTP THDR TCID



A	TCID start 31-bit											29B2A158											
TCID cont. 32-bit												000100D6											
0	C	S	E	S	R	R	0	L	C	O	0	Data Offset/4 16-bit											
	S	M	M	R	A	Y		M	Q	S													
DLF 32-bit																							
BSN 32-bit																							

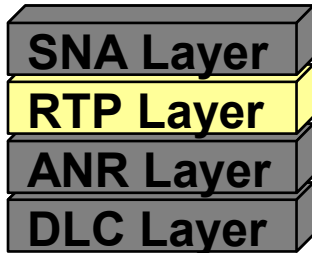
- The TCID is the first 8 bytes in the THDR
  - It uniquely identifies the pipe at the receiving RTP node
  - To follow traffic in both directions, both TCIDs must be used when filtering a trace.

```

IP      45000063 41B10000 72118824 0AD9122F 0ABA56F3
UDP    2EE12EE1 004F3669
LLC    040403
ANR    C600 80 D9001401000000 8002000C01000000 D000000000000000 FF00
RTP    29B2A158000100D60004000A0000000000000001E1
STATUS 050E00000000000020000010600000000000000000
  
```



# RTP Transport Header THDR BSN



A	TCID start 31-bit														
TCID cont. 32-bit															
0	C	S	E	S	R	R	0	L	C	O	0	Data Offset/4 16-bit			
	S	M	M	R	A	Y		M	Q	S					
DLF 32-bit											00000000				
BSN 32-bit											000001E1				

- The Byte Sequence Number keeps track of the data sent
  - Increments with every byte of payload (DLF field )
  - Also increments if End\_of\_Message bit is set

```

IP      45000063 41B10000 72118824 0AD9122F 0ABA56F3
UDP     2EE12EE1 004F3669
LLC     040403
ANR     C600 80 D9001401000000 8002000C01000000 D000000000000000 FF00
RTP     29B2A158000100D60004000A00000000000001E1
STATUS 050E00000000000020000010600000000000000000
  
```

# Translating from VTAM to IP

## Where is my lost packet?

- How can I identify an HPR packet in an IP Packet Trace?
- At the sender
  - Note the unique BSN/DLF on a given pipe (TCID)
  - Remember the IPID in the IP header
- In the network/ at the receiver
  - Look for the IP identifier of the sent IP packet
    - Remember: the IP addresses may be NAT'ed
    - The IP datagram may have been fragmented
    - The trace may not show the full packet
  - Verify the BSN and TCID
- Network Support people will be happy to track down a lost



# RTP Retransmission STATUS Segment reporting a GAP

```
----- BSN 00000074 DLF 00000012 sent out -----
IP      45C0004A 43500000 3C11BBDE 0ABA56F3 0AD9122F
UDP     2EE12EE1 0036B528
```

- A lost packet is retransmitted

```
NLP     C600 80 FF00
RTP     00000000100530330000000500000012 00000074 SOM EOM
FID5TH 5D0000000000020000000000
```

```
----- GAP report coming in asking for 00000074
```

```
IP      4500005B 43AB0000 72118632 0AD9122F 0ABA56F3
UDP     2EE12EE1 0047BDA2
```

```
LLC     040403
```

```
NLP     C600 D400000000000000 FF00
```

```
RTP     0F160F7C0001011B0004000C00000000000000EE
```

```
STATUS 070E800100000002 00000074 000000000000000000000000087000000A3
```

```
----- BSN 00000074 DLF 00000012 sent out again
```

```
IP      45C0005E 43620000 3C11BBB8 0ABA56F3 0AD9122F
UDP     2EE12EE1 004A9AF8
```

```
LLC     040403
```

```
NLP     C608 80 FF00
```

```
RTP     0000000010053033C04000A00000012 00000074 SOM EOM
```

```
STATUS 05E0000000300000000000EE0000000000000000
```

```
FID5TH 5D0000000000020000000000
```

Where is my  
IPID 4350!



# D NET,EEDIAG,REXMIT=

## Finding retransmitting connections

```
D NET,EEDIAG,REXMIT=1
```

```
IST097I  DISPLAY  ACCEPTED
IST350I  DISPLAY TYPE = EEDIAG
IST2065I  ENTERPRISE EXTENDER CONNECTION REXMIT INFORMATION
IST2067I  EEDIAG DISPLAY ISSUED ON 08/25/09 AT 16:08:59
IST924I  -----
IST1680I  LOCAL  IP ADDRESS 10.232.72.11
IST1680I  REMOTE IP ADDRESS 129.35.231.237
IST2024I  CONNECTED TO SWITCHED PU PUSA01
IST924I  -----
IST2033I  PORT PRIORITY = MEDIUM
IST2036I    NLPS SENT           =           95 ( 000K )
IST2038I    NLPS RETRANSMITTED =           7 ( 000K )
IST2068I    NLP RETRANSMIT RATE =           7%
IST924I  -----
IST2035I  TOTALS FOR ALL PORT PRIORI
IST2036I    NLPS SENT           =           95 ( 000K )
IST2038I    NLPS RETRANSMITTED =           7 ( 000K )
IST2068I    NLP RETRANSMIT RATE =           7%
IST2069I  Fnsa.nlp.thdr.optional.0e.gap == 1 AT 13:24
IST314I  ENL
```

We're losing way too many packets!

Wireshark Filter to find GAP reports:

Fnsa.nlp.thdr.optional.0e.gap == 1 AT 13:24



# RTP - Data Flow Control

## ARB algorithm

- Adaptive **Rate Based** algorithm
  - Operates on Send Rates (bits/s)
  - Initial Sendrate ( 5% of TG's capacity in APPN Topology)
  - Allowed Sendrate
    - Controlled by the receiving RTP
- The goal is to avoid congestion *before* packets get lost
  - BASE ARB was too polite to compete with greedy TCP/IP
  - Responsive Mode ARB (ARB2) is standard these days
  - Progressive Mode ARB available in V1R11 and CS V6R4
- ARB Segments (22) are used to adjust the allowed sendrate
  - Based on network delay changes

# RTP Data Flow Control

## ARB in action

```
07:20:00.56 ARB RPNCB:1157C800 SR(kb/s) : 468 SZ:1401
07:20:00.64 ODPK EE_LOW 25BBF0C300010296 BSN:0025D375 continued me OUT
07:20:00.64 ODPK EE_LOW 25BBF0C300010296 BSN:0025D8BE continued me OUT
07:20:00.64 ODPK EE_LOW 25BBF0C300010296 BSN:0025DE07 continued me OUT
07:20:00.64 ODPK EE_LOW 25BBF0C300010296 BSN:0025E350 continued me OUT
07:20:00.64 ODPK EE_LOW 25BBF0C300010296 BSN:0025E899 continued me OUT
07:20:00.64 ODPK EE_LOW 25BBF0C300010296 BSN:0025EDE2 continued me OUT
07:20:00.64 ODPK EE_LOW 25BBF0C300010296 BSN:0025F32B continued me OUT
07:20:00.64 ODPK EE_LOW 25BBF0C300010296 BSN:0025F874 continued me OUT
07:20:00.67 ODPK EE_LOW 253ABFFE000100BF BSN:00000467 ARB REP -25 IN
07:20:00.68 ARB RPNCB:1157C800 SR(kb/s) : 387 SZ:1401
07:20:00.75 ODPK EE_LOW 25BBF0C300010296 BSN:0025FDBD ARB REQ cont OUT
07:20:00.76 ODPK EE_LOW 25BBF0C300010296 BSN:00260306 continued me OUT
07:20:00.76 ODPK EE_LOW 25BBF0C300010296 BSN:0026084E continued me OUT
07:20:00.76 ODPK EE_LOW 25BBF0C300010296 BSN:00261419 continued me OUT
07:20:00.77 ODPK EE_LOW 25BBF0C300010296 BSN:00261962 continued me OUT
07:20:00.77 ODPK EE_LOW 25BBF0C300010296 BSN:00261962 continued me OUT
07:20:00.77 ODPK EE_LOW 25BBF0C300010296 BSN:00000467 ARB REQ cont OUT
```

- RTP sending at 468 kb/s
- Slowdown2 coming in
- Sendrate reduced to 387 kb/s

Here are the slowdown segments:  
Network delays building up!

**Wireshark Filter to find ARB SLOWDOWN:**  
`sna.nlp.thdr.optional.22.raa >1`



# RTP - Data Flow Control

## ARB algorithm – VERY time sensitive

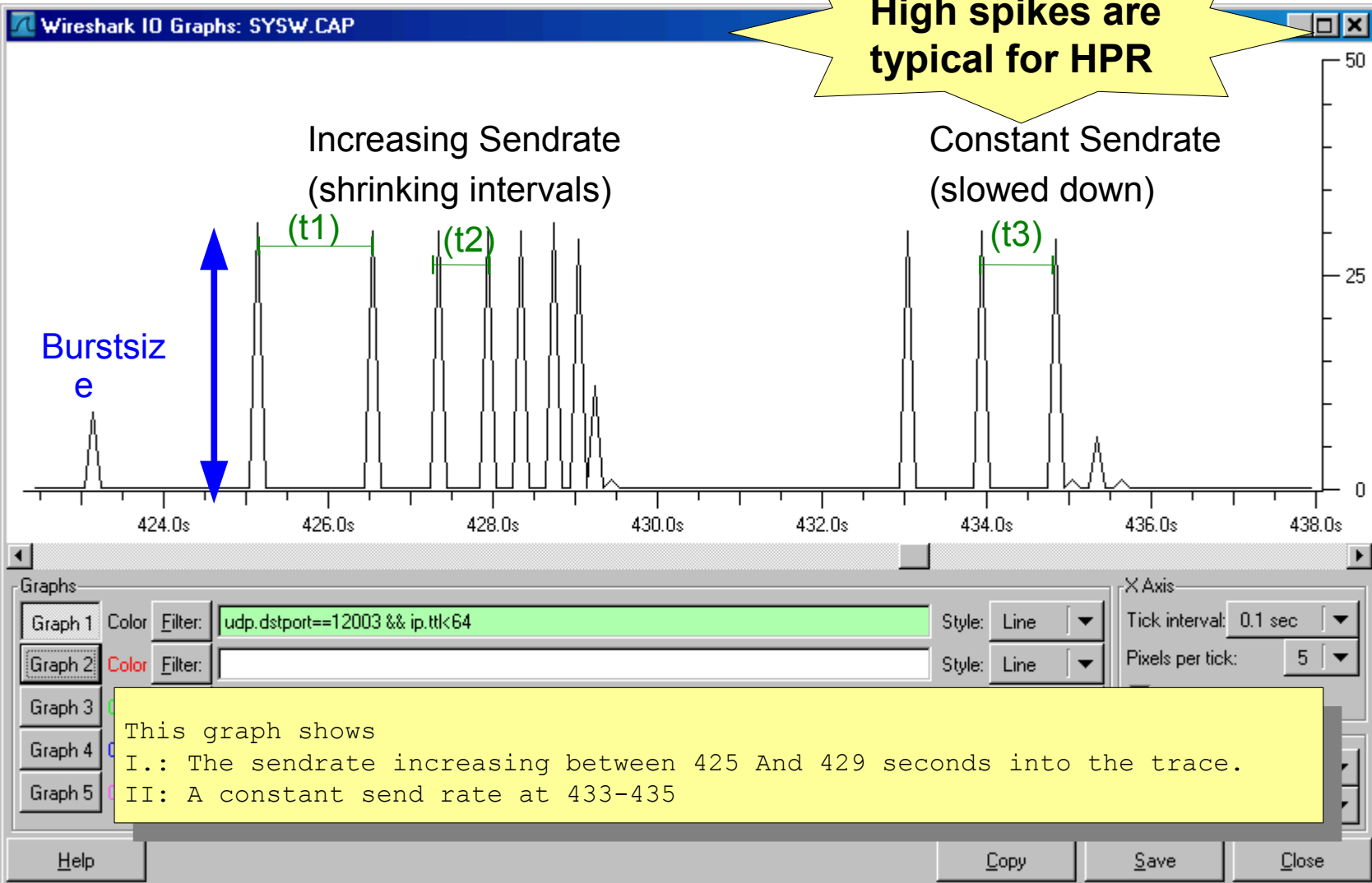
- The receiving RTP is measuring the network delays
- Increasing network delays are treated as an indication of congestion building up in the network
  - Queues in bottleneck routers are building up
- To avoid a queue overflow in the network, ARB reduces the sendrate before packets are dropped allowing the network to recover sooner.
- Sometimes the delays are caused *within* the RTP nodes
  - CPU constrained systems (zSeries with few real CPUs)
  - z/OS and Linux under z/VM
  - Windows/Linux under VMWARE/Citrix environments





# ARB algorithm - Burstsize/Burstinterval

High spikes are typical for HPR

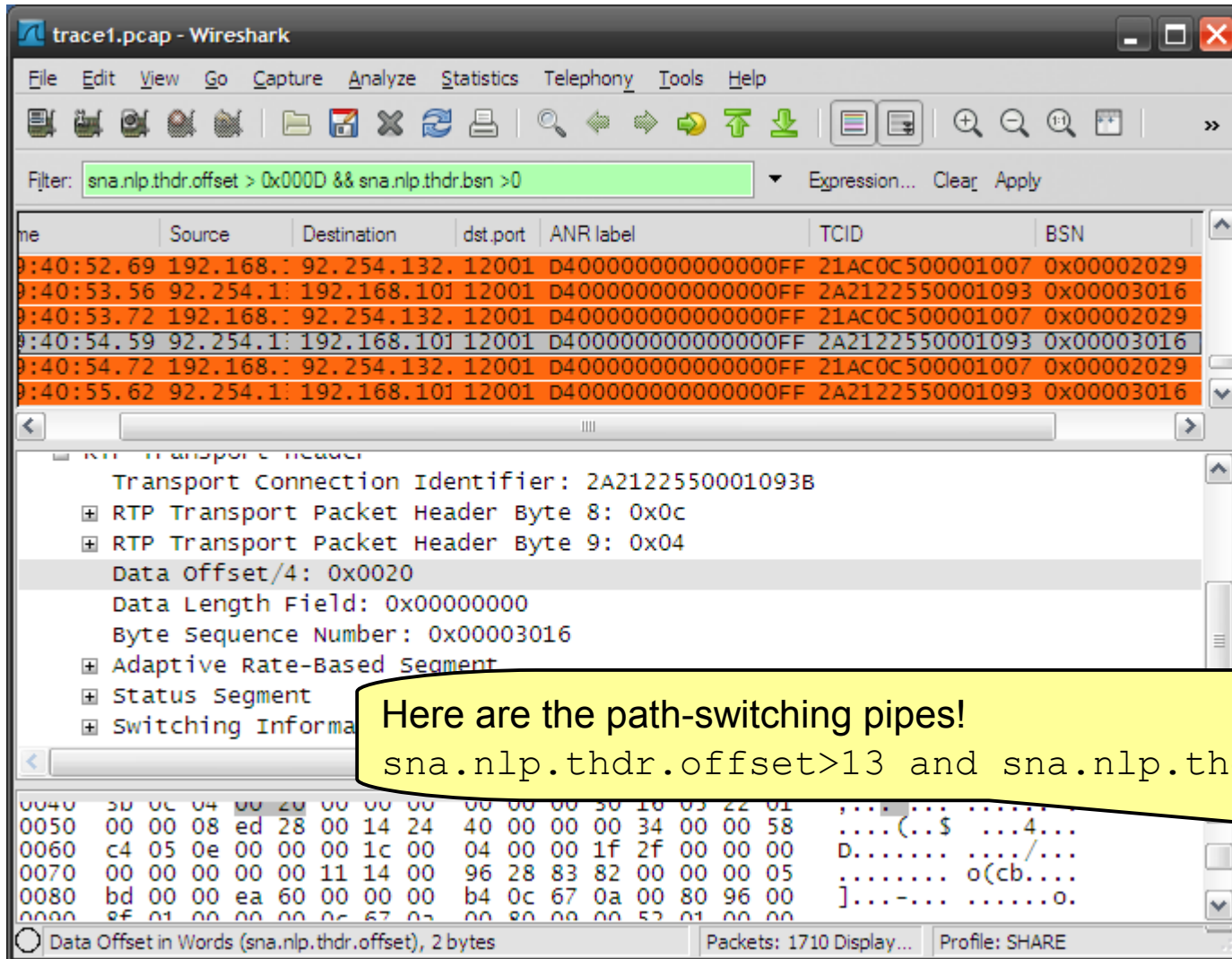


# RTP – PATHSWITCH

## Identifying switching pipes in a trace

- Most HPR problems show up as path-switching pipes
- NLPs of an existing path-switching pipe will contain following information
  - The BSN will higher than zero
  - ARB SETUP is present to re-initialize the ARB algorithm
    - Length is 5 words
  - SWINFO is present to describe the characteristics of the new path
    - Length is variable but typically larger than 8 words
- In abbreviated traces the segments might not be traced

# Wireshark filter - PATHSWITCH



trace1.pcap - Wireshark

Filter: `sna.nlp.thdr.offset > 0x000D && sna.nlp.thdr.bsn > 0`

Time	Source	Destination	dst.port	ANR label	TCID	BSN
0:40:52.69	192.168.1.1	92.254.132.12001	12001	D400000000000000FF	21AC0C500001007	0x00002029
0:40:53.56	92.254.132.12001	192.168.101	12001	D400000000000000FF	2A2122550001093	0x00003016
0:40:53.72	192.168.1.1	92.254.132.12001	12001	D400000000000000FF	21AC0C500001007	0x00002029
0:40:54.59	92.254.132.12001	192.168.101	12001	D400000000000000FF	2A2122550001093	0x00003016
0:40:54.72	192.168.1.1	92.254.132.12001	12001	D400000000000000FF	21AC0C500001007	0x00002029
0:40:55.62	92.254.132.12001	192.168.101	12001	D400000000000000FF	2A2122550001093	0x00003016

Transport Connection Identifier: 2A2122550001093B

- RTP Transport Packet Header Byte 8: 0x0c
- RTP Transport Packet Header Byte 9: 0x04
- Data offset/4: 0x0020
- Data Length Field: 0x00000000
- Byte Sequence Number: 0x00003016
- Adaptive Rate-Based Segment
- Status Segment
- Switching Information

Data Offset in Words (sna.nlp.thdr.offset), 2 bytes

Packets: 1710 Display... Profile: SHARE

Here are the path-switching pipes!  
`sna.nlp.thdr.offset>13 and sna.nlp.thdr.bsn>0`



# IPCS Formatter

## Export SYSTCPDA to sniffer

- Convert your SYSTCPDA/SYSTCPOT traces in sniffer

```
//SYSTSIN DD *  
PROFILE MSGID  
  IPCS NOPARM  
  SETD PRINT NOTERM LENGTH(160000) NOCONFIRM FILE(INDMP)  
  DROPD  
CTTRACE COMP(SYSTCPDA) -  
  OPTIONS((SNIFFER(9000,TCPDUMP))) SUB((TCPIP))  
END
```

- Download the resulting file in binary
- Run a trace tool against the trace to filter the down to the problematic packets
- Talk to your network personnel using *their language!*

Here's the problem:

Fix the firewall rule 5 hops away!



**IBM**®

# wireshark

## Display Filters

- Stored as dfilters in
  - C:\Documents and Settings\Administrator\Application Data\Wireshark

```
"IP Fragmentation" ip.frag_offset>0 or ip.flags.mf == 1 or icmp.code == 4
"Low TTL - Routing Loop" ip.ttl < 8 and !ip.ttl==1 and !udp.srcport==33434
"ICMP Errors" icmp and !icmp.type==8 and !icmp.type==0
"PING" icmp.type==8 or icmp.type==0
"Time Delta" frame.time_delta > 0.2
"Warning" expert.severity>=1024
"EEonly " (udp.dstport > 11999 && udp.dstport<12005) or ICMP
"SNA BIND/UNBIND" (data.data[0] == 31 or data.data[0] == 32) and sna.rh.ru_category == 0x03
"UNBIND Cleanup" data.data[0] == 32 and data.data[1] >02 and sna.rh.ru_category == 0x03
"CPSVCMG Pipe " sna.nlp.nhdr.anr == d4:00:00:00:00:00:00:ff
"Pipe SETUP " sna.nlp.thdr.bsn == 0 or sna.gds.type == 0x12ce
"PATHSWITCH" sna.nlp.thdr.offset > 0x000D && sna.nlp.thdr.bsn >0 or sna.gds.type == 0x12ce
"Pipe Termination" sna.nlp.thdr.optional.type == 0x12
"GAP " sna.nlp.thdr.optional.0e.gap == 1
"SLOWDOWN" sna.nlp.thdr.optional.22.raa >1
```

# Questions?

- Come to Tuesday's session on wireshark and EE problems



- Thank You for attending this session!