#SHAREorg

SHARE
Technology · Connections · Results

Towards the OSA and beyond - Using wireshark for z/OS Packet Trace Analysis

Matthias Burkhard
IBM Germany
mburkhar@de.ibm.com

Session 10827

March 14, 2012: 03:00 PM - 04:00 PM, OMNI, Hickory

Session Contents

The days are over when connectivity problems in the System z could be solved by z/OS personnel only.
In today's modern multi-tier multi-platform application designs a new approach in network diagnosis is required.

While the z/OS packet trace is always a good start on the quest to the real root cause of a problem, unfortunately outside the zSeries the SYSTCPDA packet trace is not known well enough to serve as a trusted evidence.
This session will demonstrate how the use of wireshark helped to speed up problem resolution for problems that surfaced on z/OS but had their root cause outside the mainframe.

This session is a preparation for the wireshark hands-on lab session 10828: Taming the (wire)shark

WebSphere MQ-FTE Performance Problem

- Understand Problem
  - What is the concern?
  - What is the impact?
  - What is the root cause?

Environment: WMQFTE V7.0.4.1
We have a performance problem with WMQFTE,
Our support person asked me to open this …'

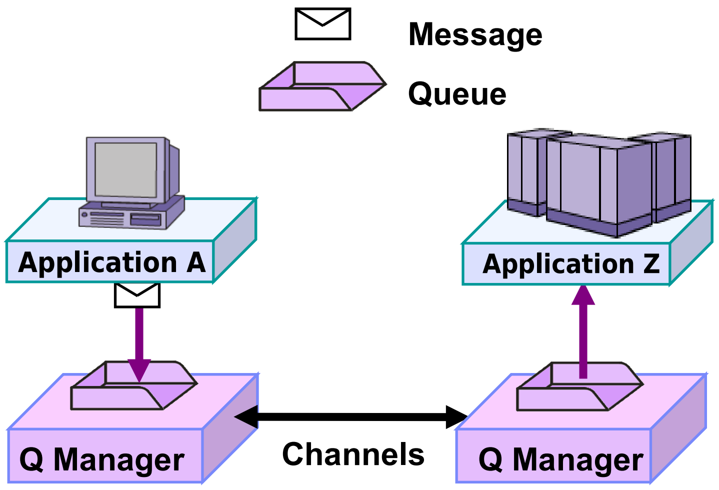
- Understand the Topology
  - What platforms are involved?
  - What does the network infrastructure look like?
  - What TCP/IP parameters are configured?

- Evaluate possible solutions/circumventions
  - Ease of implementation
  - Scope of responsibility

What is WebSphere MQ ?

- **Reliability**
  - Assured message delivery
  - Performance
- **Ubiquitous**
  - Breadth of support for platforms, programming languages and API
- **Loose application coupling**
  - Location transparency
  - Time independence
  - Real time / Near Real time
  - Data transparency (with WebSphere Message Broker)
  - Platform independence
- **Scalability**
  - Incremental growth

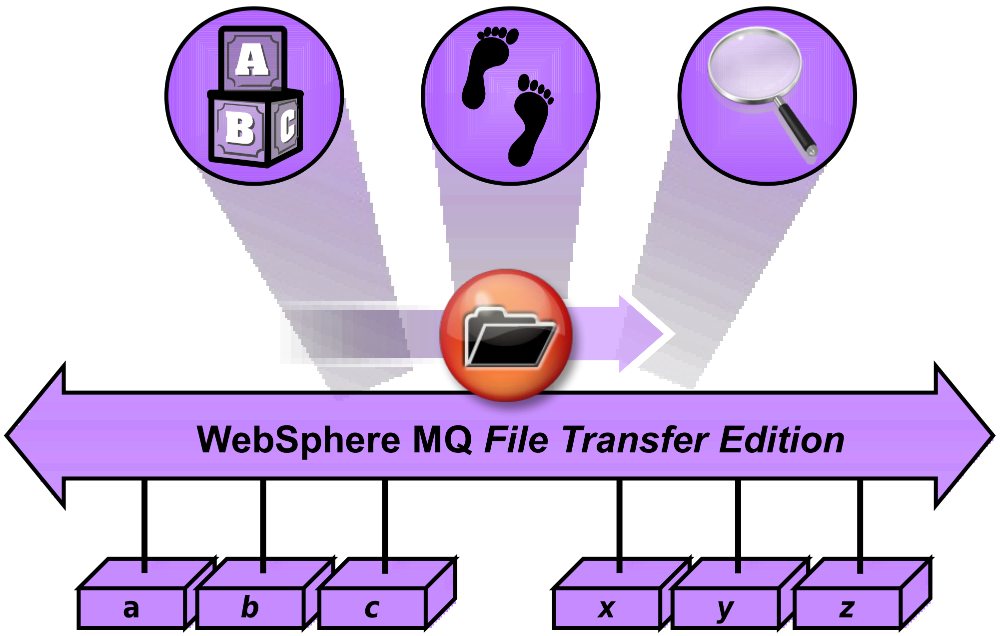
- **Rapid development**
  - Reduced Complexity
  - Ease of use



What is MQ-FTE File Transfer Edition?

- Adds file transfer services to WebSphere MQ to enable managed file movement
- Multi-purpose solution combining file transfer and messaging

- ☑ **Reliable** – leveraging the WebSphere MQ transport
- ☑ **Multi-purpose** – transfers both messages and files
- ☑ **Auditable** – logging subsystem tracks transfer at source and destination for audit purposes
- ☑ **Centralized** – monitoring, control and configuration
- ☑ **No need to program** – no MQ skills required
- ☑ **Graphical tooling** – visual configuration and status
- ☑ **Moves Massive files** – no size limit, Kb, Mb, Gb…
- ☑ **Command line interface** – for advanced users
- ☑ **Scripting support** – enables scripting of complex multi-step transfers in XML using Apache Ant
- ☑ **Flexible backbone** – moves files from anywhere to everywhere in network
- ☑ **Integration** with MQ-enabled apps and ESBs
- ☑ **Automatic** file character conversion
- ☑ **Security** of file payload using SSL
- ☑ **Support** for wide range of platforms



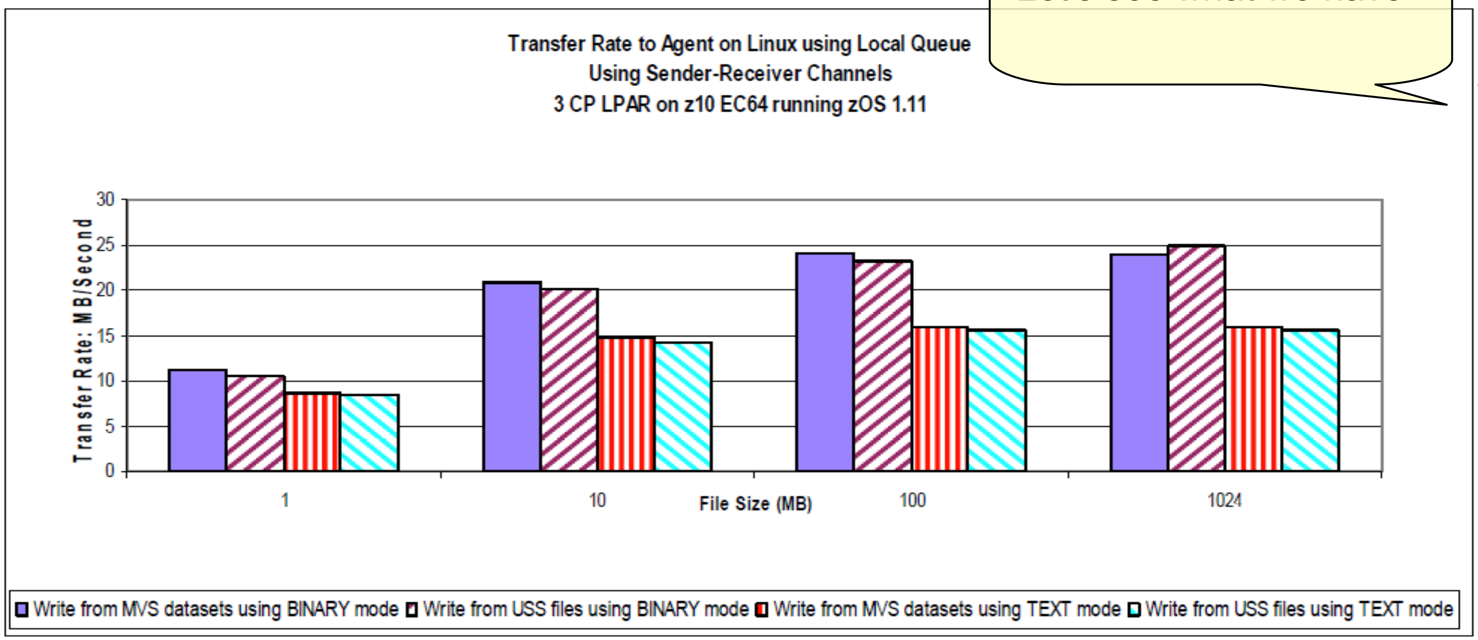
What Performance can be expected?

- Performance Reports
  - http://tinyurl.com/7kh6urn

Category 2 - Performance Reports				
No.	Name	Initial Release	Last Release	New/ Updated
FPL1	WebSphere MQ File Transfer Edition Performance Report - Linux V7.0.1	05Nov09	05Nov09	
FP11	WebSphere MQ File Transfer Edition Performance Report - System Z V7.0.1	15Dec09	15Dec09	
FP61	WebSphere MQ File Transfer Edition Performance Report - Solaris V7.0.1	05Nov09	05Nov09	

- FP11 System Z

Let's see what we have



SYSTCPDA Packet Trace – Session Report

IP CTRACE COMP(SYSTCPDA) SUB((TCPIP1)) OPTIONS((SESSION))

```
28160 packets summarized

 Local Ip Address:
 Remote Ip Address:


Host:                                    Local,           Remote
 Client or Server:                     Unknown,          Unknown
 Port:                                    1921,             1414
 Application:                                ,
 Link speed (parm):                         10,               10 Megabits/s

Connection:
 First timestamp:                   2012/02/23 16:56:24.824066
 Last timestamp:                    2012/02/23 16:58:52.543131
 Duration:                                     00:02:27.719065
 Average Round-Trip-Time:                                0.01[illegible] sec
 Final Round-Trip-Time:                                  0.067 sec
 Final state:                                      ESTABLISHED
 Out-of-order timestamps:                                    0

Data Quantity & Throughput:            Inbound,         Outbound
 Application data bytes:                   140,        129312168
 Sequence number delta:                    140,        129312168
 Total bytes Sent:                         140,        129312168
 Bytes retransmitted:                        0,                0
 Throughput:                                 0,          854.875 Kilobytes/s
 Bandwidth utilization:                  0.00%,           70.03%
 Delay ACK Threshold:                      200,              200 ms
```

...the folks involved with this project wanted the diagnostics to help determine why rate of throughput is not comparable to Connect Direct.

854Kb/s, 70% of what?
Which bandwidth is available ?
How fast is C:D?

OPTIONS(SESSION SPEED(10,10))

SYSTCPDA Packet Trace – Conversion

IP CTRACE COMP(SYSTCPDA) SUB((TCPIP1))

OPTIONS((SNIFFER(1500 TCPDUMP)))

```
//BURKSNIF JOB (7904,NCS),BURKHAR,MSGLEVEL=(1,1),MSGCLASS=K,CLASS=A,
//      NOTIFY=&SYSUID.,REGION=0M,TIME=150
//      SET INDUMP='ONTOP.GS100.P81535.C838.PKTRACE.$DUP0001'
//       SET SNIFF='ONTOP.GS100.P81535.C838.D0227.IEASYSXA.PCAP'
//      SET MIGLIB='TOP.ZOSR1C.MIGLIB'
//*               OTHERWISE THE SNIFFER FILE WILL BE EMPTY!!!
//*  THIS JOB CONVERTS A PACKET TRACE TO SNIFFER
//*  ATTENTION: PLEASE VERIFY THE TCPIP JOBNAME IS CORRECT
//*               OTHERWISE THE SNIFFER FILE WILL BE EMPTY!!!
//IPCSBTCH EXEC PGM=IKJEFT01,DYNAMNBR=30
//STEPLIB  DD DISP=SHR,DSN=&MIGLIB.
//IPCSDDIR DD DISP=SHR,DSN=&SYSUID..ZOS1C.DIRECTRY
//IPCSDUMP DD *
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//INDMP    DD DISP=SHR,DSN=&INDUMP.
//SNIFFER  DD DSN=&SNIFF.,
//    DISP=(NEW,CATLG),LRECL=1560,SPACE=(CYL,(550,50)),RECFM=VB,DSORG=PS
//* DISP=SHR
//IPCSPRNT DD SYSOUT=*
//IPCSTOC  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSTSIN  DD *
 PROFILE MSGID
  IPCS NOPARM
  SETD PRINT NOTERM LENGTH(160000) NOCONFIRM FILE(INDMP)
  DROPD
 CTRACE COMP(SYSTCPDA) SUB((TCPIPA)) -
     OPTIONS((SNIFFER(1500 TCPDUMP) -
     NOREASSEMBLY))  ENTIDLIST(4)
 END
```

Convert the SYSTCPDA to tcpdump format in BATCH

MQ-FTE Performance Problem

Statistics → Conversations

WebSphere MQ FTE uses 3 'channels', 2 for control flows - 1 for transfer,

Conversations: d0223.#1-2000.pcap

Ethernet: 1 | Fibre Channel | FDDI | IPv4: 1 | IPv6 | IPX | JXTA | NCP | RSVP | SCTP | TCP: 3 | Token Ring | UDP | USB | WLAN

TCP Conversations

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B
10.83.123.104	1921	10.199.105.195	1414	1 980	9 632 568	903	9 505 482	1 077
10.199.105.195	42496	10.83.123.104	3101	10	2 500	6	1 972	4
10.83.123.104	1908	10.199.105.195	1415	10	3 324	6	2 796	4

☐ Name resolution ☐ Limit to display filter

Help | Copy | Follow Stream | Close

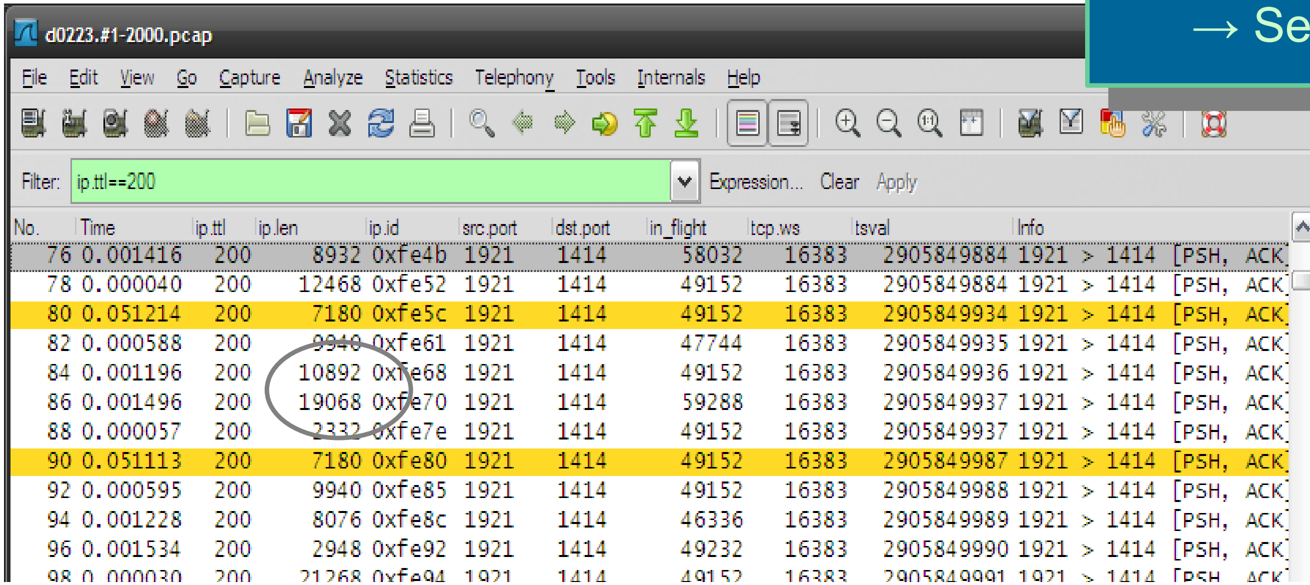
10.83.123.104
1908
1921
3101

10.199.105.195
1415: first
1414: second
42496: third

MQ-FTE Performance Problem

Filter on outbound packets using ip.ttl

ip.len greater than the standard
→ Segmentation Offload



No.	Time	ip.ttl	ip.len	ip.id	src.port	dst.port	in_flight	tcp.ws	tsval	Info
76	0.001416	200	8932	0xfe4b	1921	1414	58032	16383	2905849884	1921 > 1414 [PSH, ACK
78	0.000040	200	12468	0xfe52	1921	1414	49152	16383	2905849884	1921 > 1414 [PSH, ACK]
80	0.051214	200	7180	0xfe5c	1921	1414	49152	16383	2905849934	1921 > 1414 [PSH, ACK]
82	0.000588	200	9940	0xfe61	1921	1414	47744	16383	2905849935	1921 > 1414 [PSH, ACK]
84	0.001196	200	10892	0xfe68	1921	1414	49152	16383	2905849936	1921 > 1414 [PSH, ACK]
86	0.001496	200	19068	0xfe70	1921	1414	59288	16383	2905849937	1921 > 1414 [PSH, ACK]
88	0.000057	200	2332	0xfe7e	1921	1414	49152	16383	2905849937	1921 > 1414 [PSH, ACK]
90	0.051113	200	7180	0xfe80	1921	1414	49152	16383	2905849987	1921 > 1414 [PSH, ACK]
92	0.000595	200	9940	0xfe85	1921	1414	49152	16383	2905849988	1921 > 1414 [PSH, ACK]
94	0.001228	200	8076	0xfe8c	1921	1414	46336	16383	2905849989	1921 > 1414 [PSH, ACK]
96	0.001534	200	2948	0xfe92	1921	1414	49232	16383	2905849990	1921 > 1414 [PSH, ACK]
98	0.000030	200	21268	0xfe94	1921	1414	49152	16383	2905849991	1921 > 1414 [PSH, ACK]

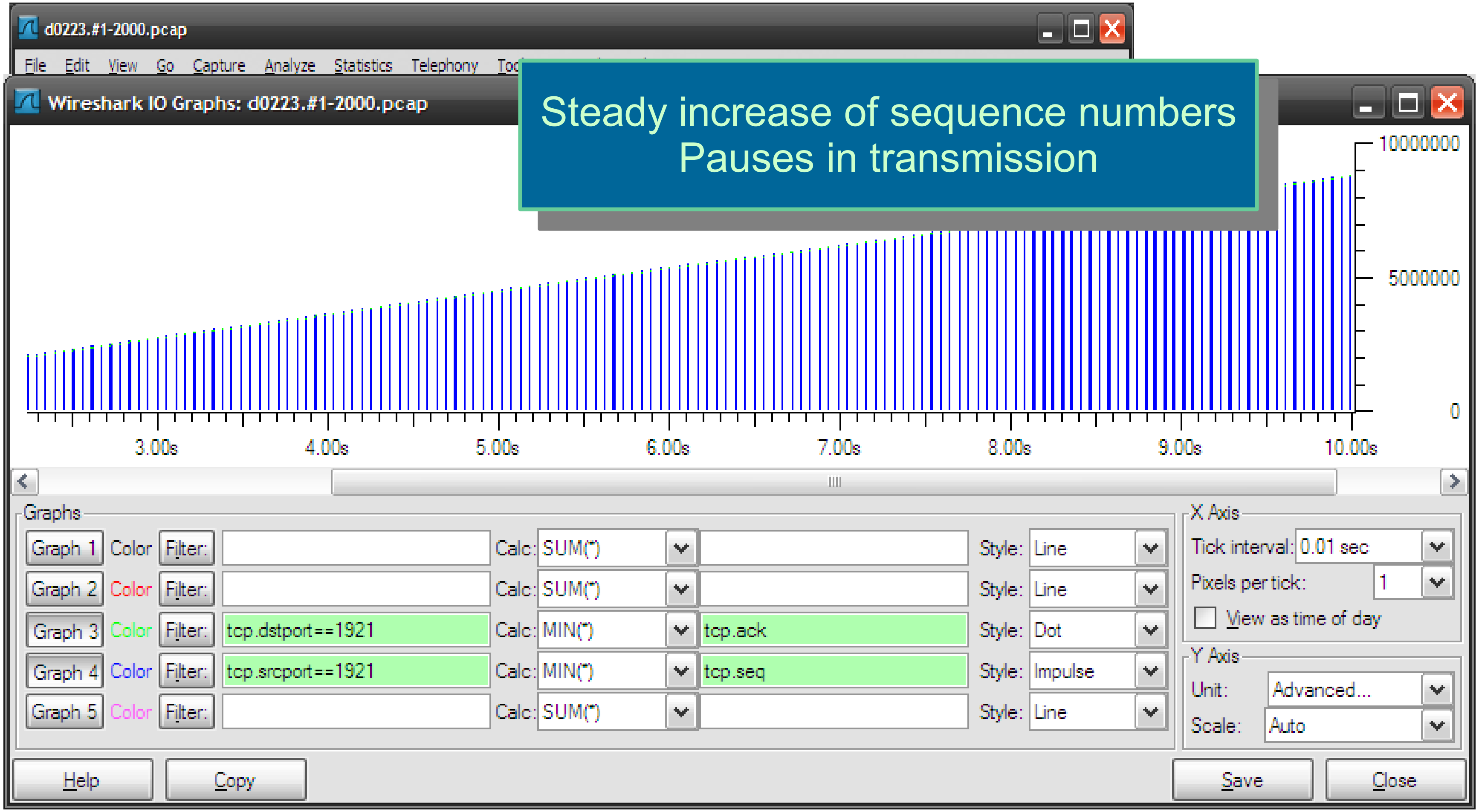
MQ Client
delays: 51 ms
SEGOFFLOAD

10.83.123.104
1908 →
1921 →
3101 ←

10.199.105.195
1415: first
1414: second
42496: third

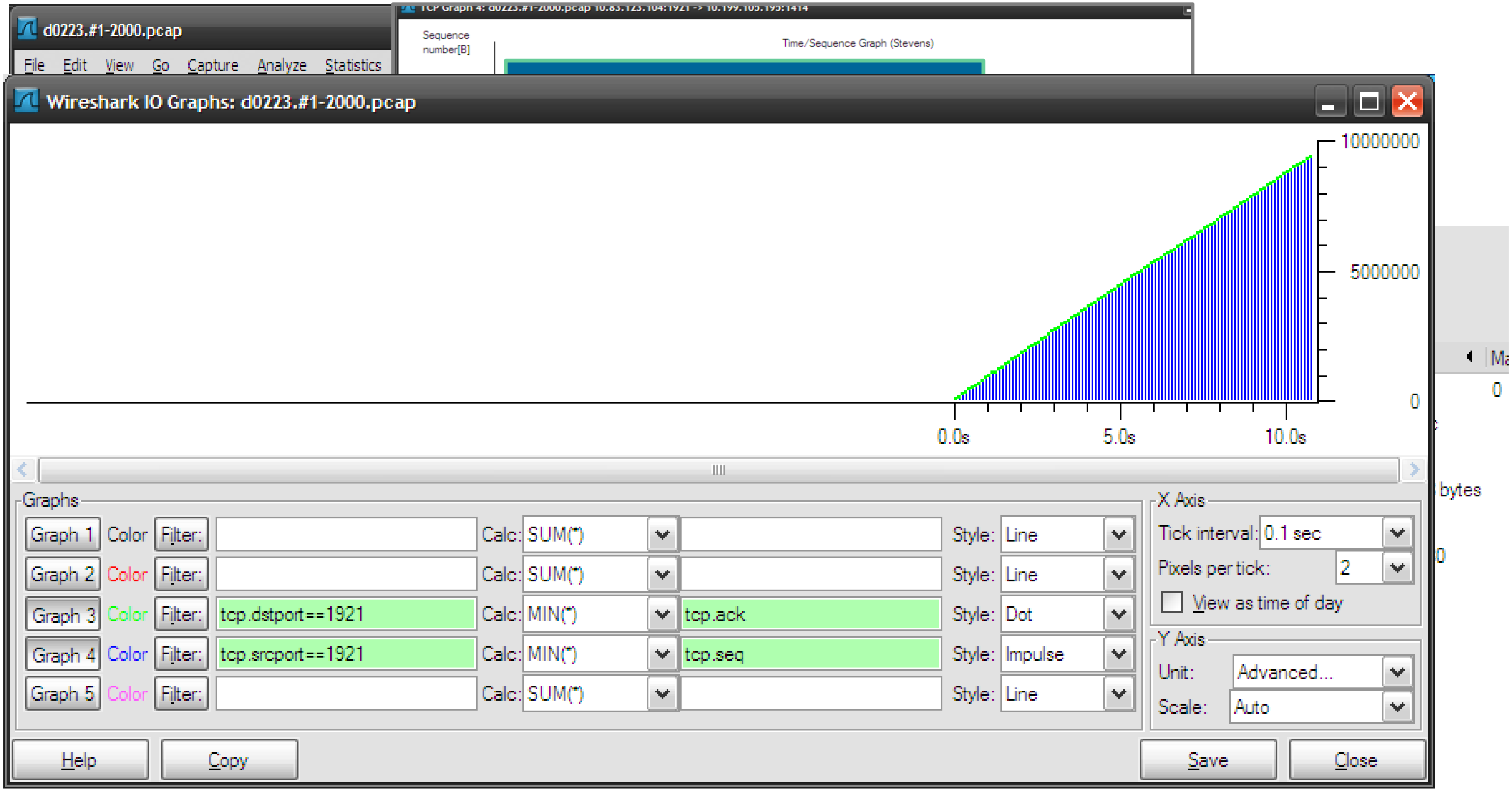
IO Graph – Throughput of a TCP connection

Print the TCP Sequence Graph over time – IO Graph



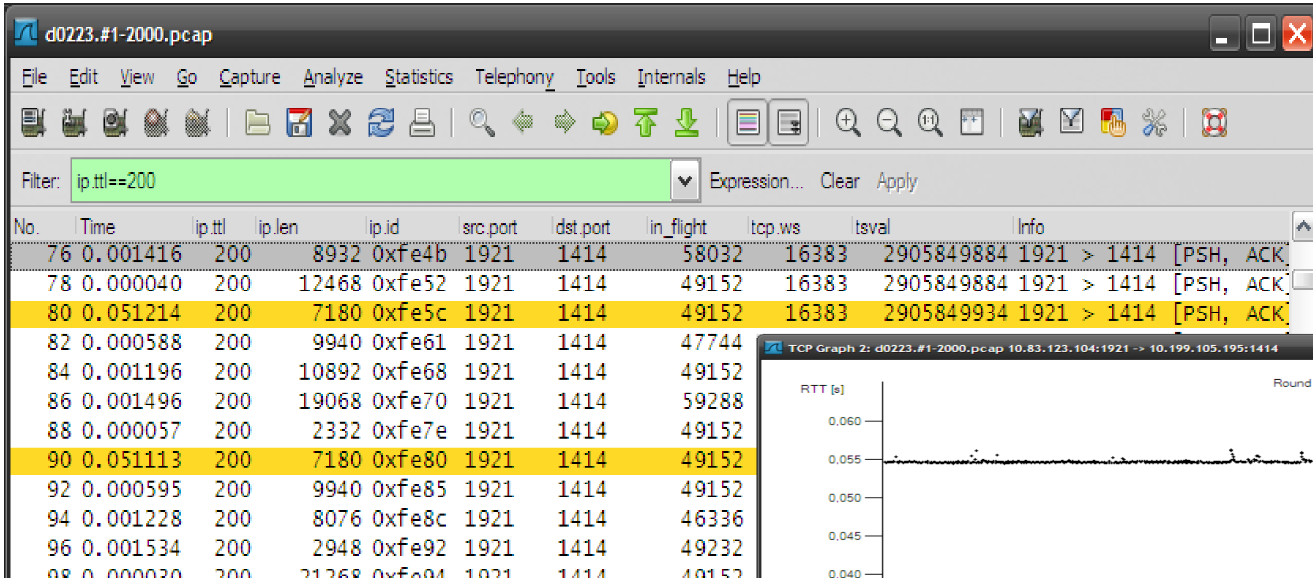
MQ-FTE Performance Problem - Throughput

Print the TCP Sequence Graph over time

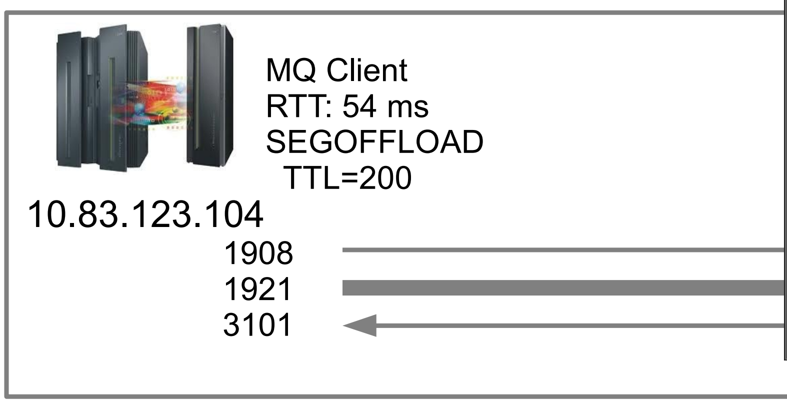


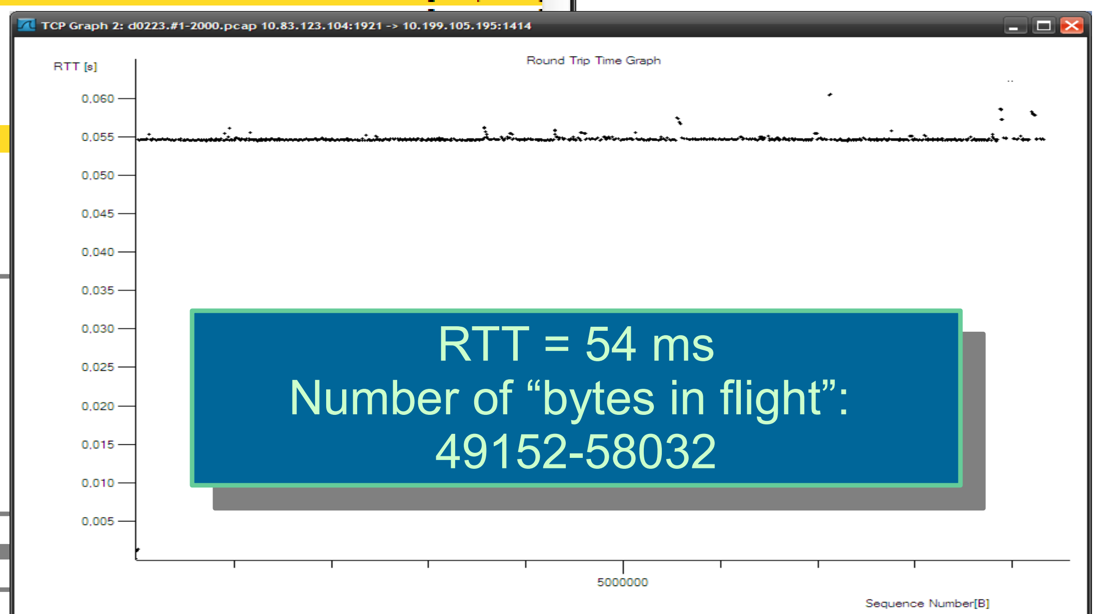
MQ-FTE Performance Problem - RTT

Statistics → TCP Stream Graph → Round Trip Time



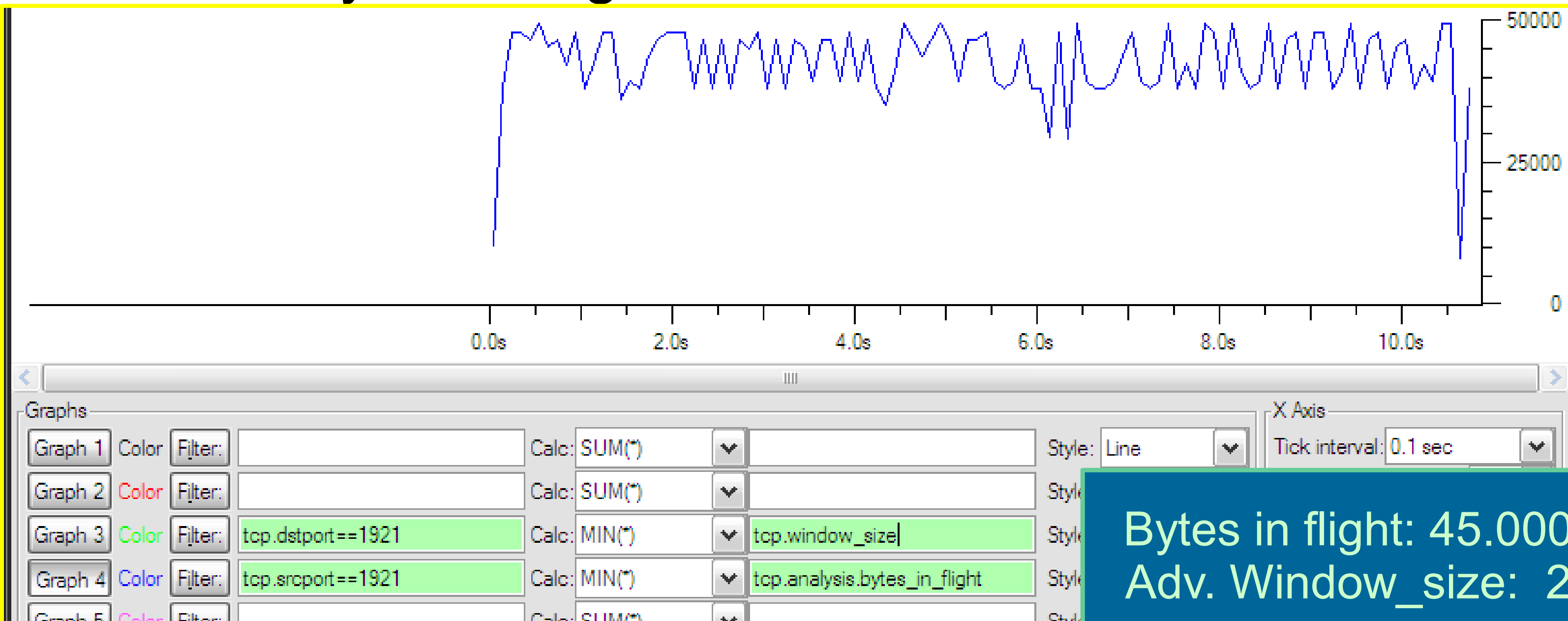
No.	Time	ip.ttl	ip.len	ip.id	src.port	dst.port	in_flight	tcp.ws	tsval	Info
76	0.001416	200	8932	0xfe4b	1921	1414	58032	16383	2905849884	1921 > 1414 [PSH, ACK
78	0.000040	200	12468	0xfe52	1921	1414	49152	16383	2905849884	1921 > 1414 [PSH, ACK
80	0.051214	200	7180	0xfe5c	1921	1414	49152	16383	2905849934	1921 > 1414 [PSH, ACK
82	0.000588	200	9940	0xfe61	1921	1414	47744			
84	0.001196	200	10892	0xfe68	1921	1414	49152			
86	0.001496	200	19068	0xfe70	1921	1414	59288			
88	0.000057	200	2332	0xfe7e	1921	1414	49152			
90	0.051113	200	7180	0xfe80	1921	1414	49152			
92	0.000595	200	9940	0xfe85	1921	1414	49152			
94	0.001228	200	8076	0xfe8c	1921	1414	46336			
96	0.001534	200	2948	0xfe92	1921	1414	49232			
98	0.000030	200	21268	0xfe94	1921	1414	49152			





IO Graph – bytes in flight vs adv. windowsize

Outbound Bytes in flight – inbound advertised windowsize



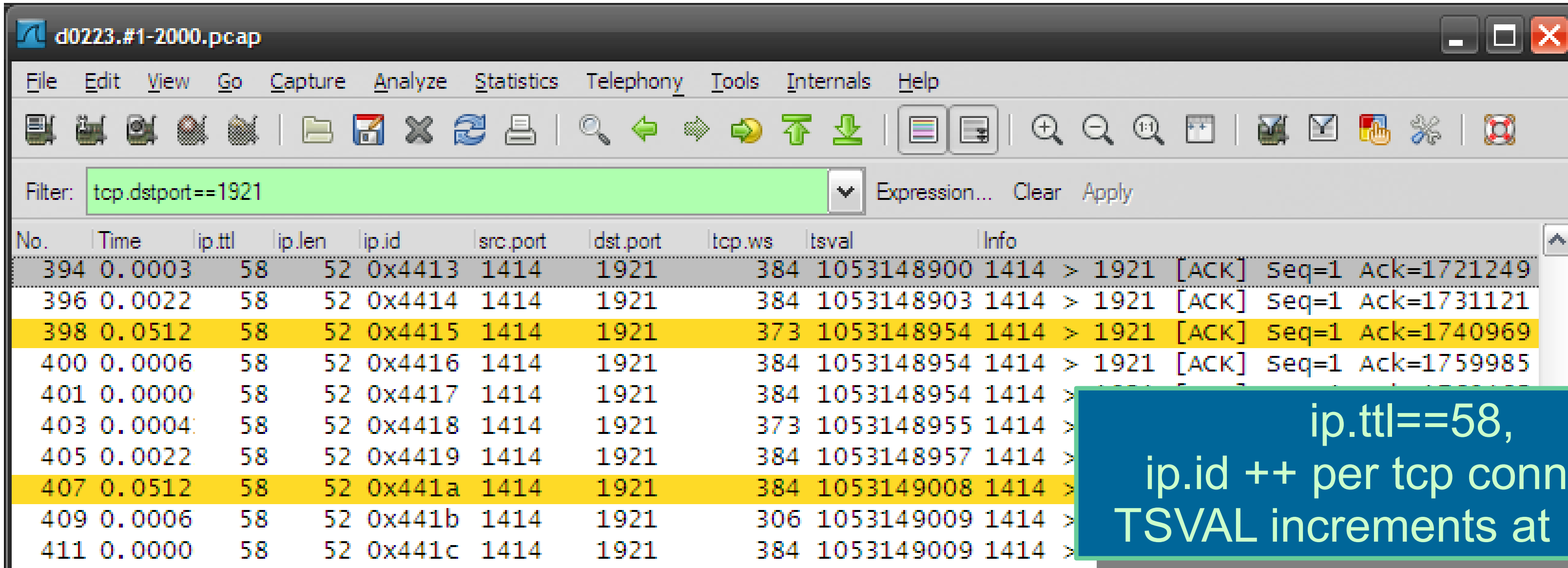
Bytes in flight: 45.000-50.000
Adv. Window_size: 295-384

11	0.051886	58	52	0x434b	1414	1921	295	1053147044	1414 >	1921
13	0.000929	58	52	0x434c	1414	1921	362	1053147045	1414 >	1921
15	0.000045	58	52	0x434d	1414	1921	384	1053147045	1414 >	1921
17	0.001288	58	52	0x434e	1414	1921	373	1053147046	1414 >	1921
19	0.000307	58	52	0x434f	1414	1921	384	1053147047	1414 >	1921
21	0.051967	58	52	0x4350	1414	1921	384	1053147099	1414 >	1921
23	0.000938	58	52	0x4351	1414	1921	306	1053147100	1414 >	1921
25	0.001253	58	52	0x4352	1414	1921	362	1053147101	1414 >	1921
26	0.000001	58	52	0x4353	1414	1921	384	1053147101	1414 >	1921
28	0.000323	58	52	0x4354	1414	1921	373	1053147101	1414 >	1921
30	0.052041	58	52	0x4355	1414	1921	373	1053147153	1414 >	1921
31	0.000003	58	52	0x4356	1414	1921	384	1053147153	1414 >	1921
33	0.000879	58	52	0x4357	1414	1921	384	1053147154	1414 >	1921
35	0.001344	58	52	0x4358	1414	1921	295	1053147155	1414 >	1921
37	0.000981	58	52	0x4359	1414	1921	373	1053147156	1414 >	1921
38	0.000002	58	52	0x435a	1414	1921	384	1053147156	1414 >	1921

Window Scale Factor:128

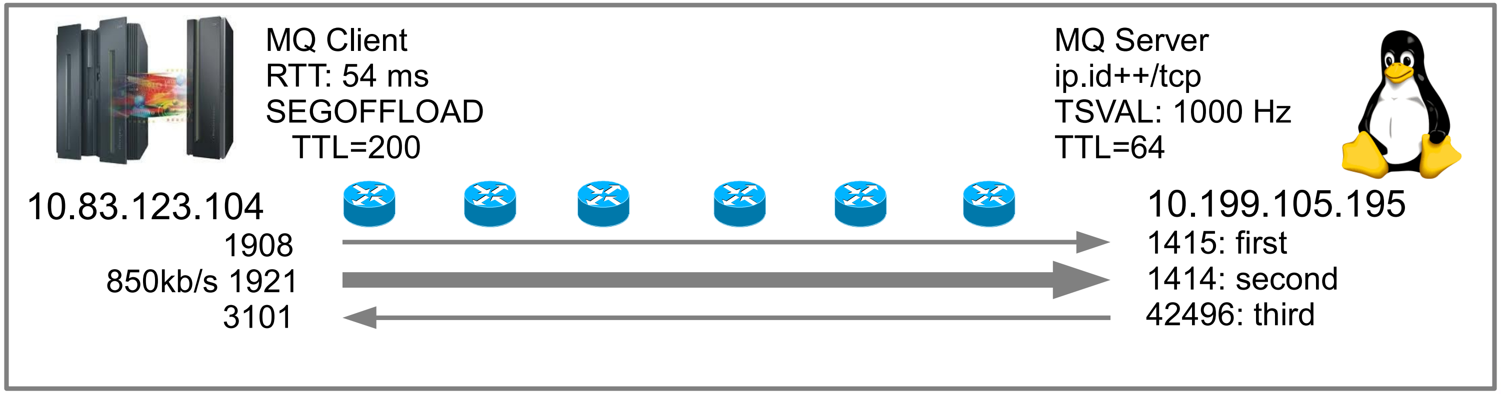
MQ-FTE Performance Problem – Remote Host

Filter on inbound flows – Identify the remote system



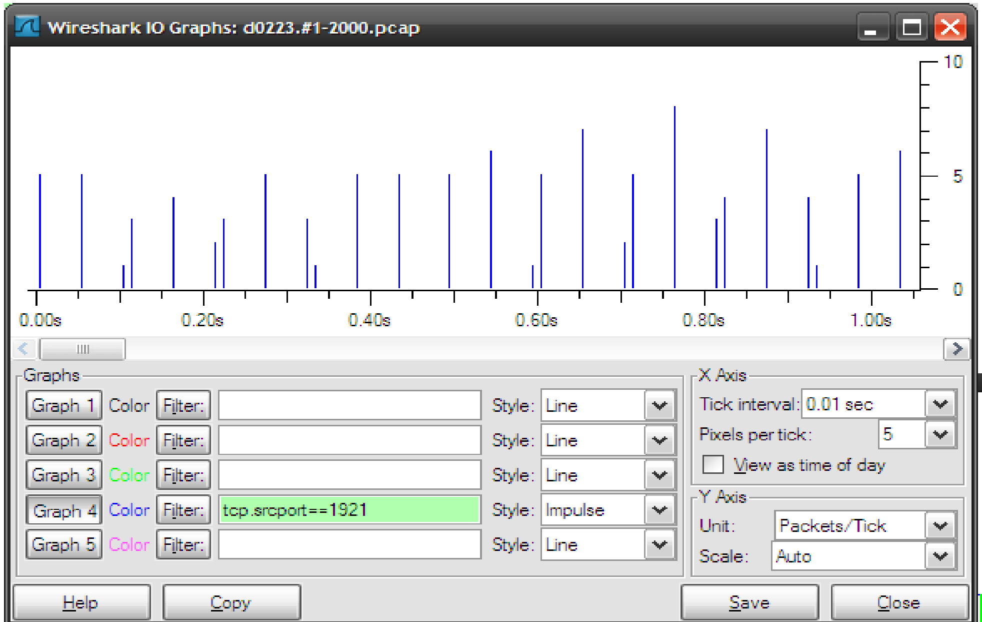
No.	Time	ip.ttl	ip.len	ip.id	src.port	dst.port	tcp.ws	tsval	Info
394	0.0003	58	52	0x4413	1414	1921	384	1053148900	1414 > 1921 [ACK] Seq=1 Ack=1721249
396	0.0022	58	52	0x4414	1414	1921	384	1053148903	1414 > 1921 [ACK] Seq=1 Ack=1731121
398	0.0512	58	52	0x4415	1414	1921	373	1053148954	1414 > 1921 [ACK] Seq=1 Ack=1740969
400	0.0006	58	52	0x4416	1414	1921	384	1053148954	1414 > 1921 [ACK] Seq=1 Ack=1759985
401	0.0000	58	52	0x4417	1414	1921	384	1053148954	1414 >
403	0.0004	58	52	0x4418	1414	1921	373	1053148955	1414 >
405	0.0022	58	52	0x4419	1414	1921	384	1053148957	1414 >
407	0.0512	58	52	0x441a	1414	1921	384	1053149008	1414 >
409	0.0006	58	52	0x441b	1414	1921	306	1053149009	1414 >
411	0.0000	58	52	0x441c	1414	1921	384	1053149009	1414 >

ip.ttl==58,
ip.id ++ per tcp connection
TSVAL increments at 1000Hz

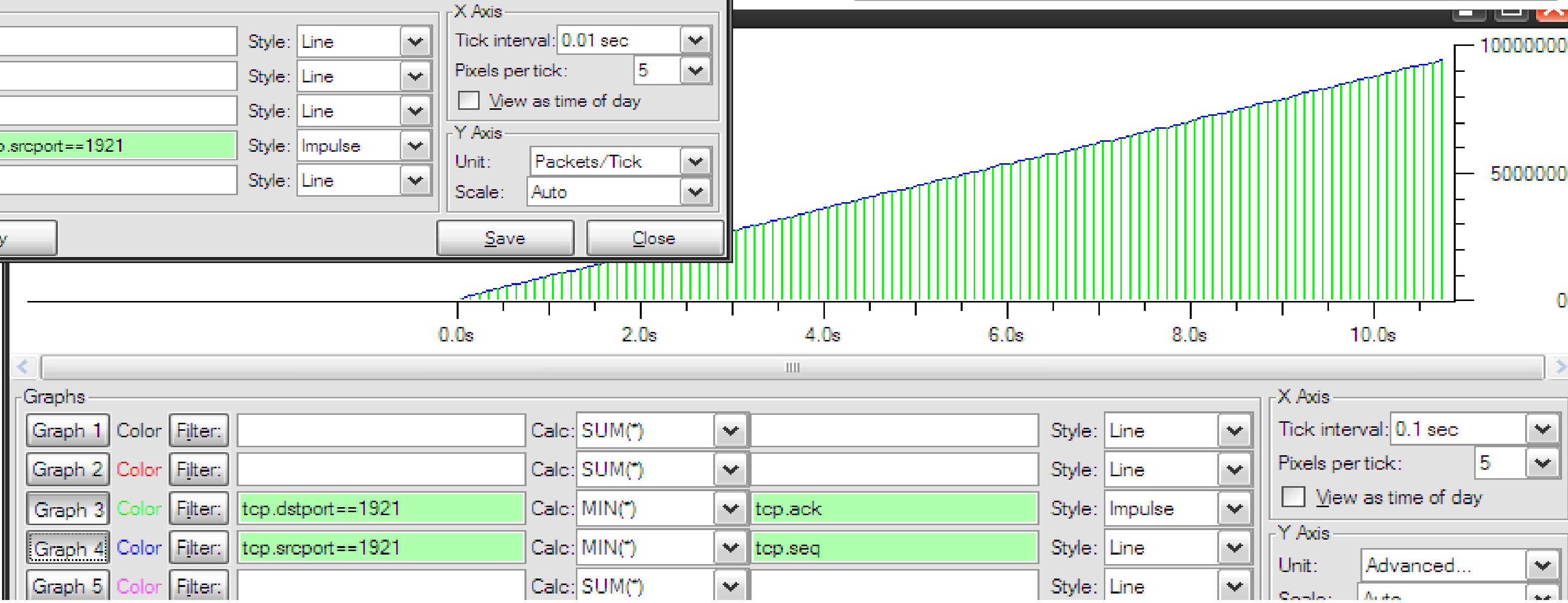


MQ-FTE Performance Problem – IO Graph

Outbound seq# - inbound ack#



Steady flow
Seq# and ack# in synch
51 ms gaps in transmission



MQ-FTE Performance Problem – BDP

http://en.wikipedia.org/wiki/Bandwidth-delay_product

BDP Bandwidth Delay Product
Available Bandwidth * Network Delay = size of TCP Receivebuffers
Example: 10 Mb/s link with a delay of 0.054 secs requires 70KB buffer for a steady TCP flow, for faster links even more...

A high bandwidth-delay product is an important problem case ... because the protocol can only achieve optimum throughput if a sender sends a sufficiently large quantity of data before being required to stop and wait until a confirming message is received from the receiver, acknowledging successful receipt of that data.
If the quantity of data sent is insufficient compared with the bandwidth-delay product, then the link is not being kept busy and the protocol is operating below peak efficiency for the link.

Thank You for your time!

Matthias Burkhard
IBM
mburkhar@de.ibm.com

Twitter : mreede

IP Wizards
ip.wizards@groups.facebook.com

SHARE in Atlanta 2012

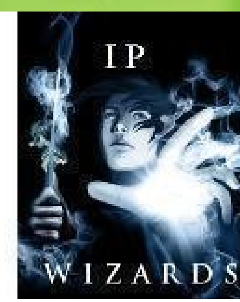
EE Education – IP wizards

ITS53 EE Implementation and Problem Determination

4 days ITSO Workshop – 30.April 2012 Miami,FL

Register at http://greenhouse.lotus.com

Join the IP wizards community



http://tinyurl.com/ipwizards

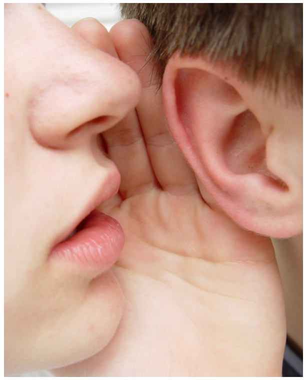
to download wireshark profiles and p0f fingerprints

Evaluation Forms – Hands On Lab

We really value your feedback!

Please take a minute to fill out the evaluation form - leave comments where appropriate.

http://atlanta.SHARE.org/SessionEvaluation



Session 10828: Taming the Shark lotus.
You have used wireshark before and found it valuable in your job as a z/OS TCPIP System Administrator?
Chances are, you've just scratched at the tip of the iceberg.
Come to this session to gain some hands-on experience and see how you really save time in trouble shooting by using some not-so-obvious filters, coloring rules and graphical features of the wireshark tool.

You're invited to bring and use your own computer to look at some trace examples showing real TCP/IP problems.