



# 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?
- 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

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

Performance problem! Classic! Let's get started...





#### 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
  - ☑ **<u>Reliable</u>** leveraging the WebSphere MQ transport
  - ☑ <u>Multi-purpose</u> transfers both messages and files
  - Auditable logging subsystem tracks transfer at source and destination for audit purposes
  - ☑ **<u>Centralized</u>** monitoring, control and configuration
  - ☑ No need to program no MQ skills required
  - ☑ **<u>Graphical tooling</u>** visual configuration and status
  - ☑ **<u>Moves Massive files</u>** 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
  - ☑ <u>Security</u> of file payload using SSL
  - ☑ <u>Support</u> for wide range of platforms







#### What Performance can be expected?

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

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

Let's see what we have

• FP11 System Z

Transfer Rate to Agent on Linux using Local Queue Using Sender-Receiver Channels 3 CP LPAR on z10 EC64 running zOS 1.11



Write from MVS datasets using BINARY mode Write from USS files using BINARY mode Write from MVS datasets using TEXT mode Write from USS files using TEXT mode



#### **SYSTCPDA** Packet Trace – Session Report



🗉 in Atlanta

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

28160 packets summarized

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



OPTIONS (SESSION SPEED (10, 10)))



#### **SYSTCPDA** Packet Trace – Conversion



#### **MQ-FTE Performance Problem**



#### Statistics $\rightarrow$ Conversations

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

Conversations: dozz5.#1-2000.pcap			
Ethernet: 1 Fibre Channel FDDI IPv4: 1 IPv6 IPX JXTA	NCP RSVP SCTP TCP: 3 T	oken Ring   UDP   USB   WLAN	
TCP CC	nversations		
Address A • Port A • Address B • Port B • Packe	ts ↓ Bytes ↓ Packets A →B ↓	Bytes 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	5 1972 4	
10.83.123.104 1908 10.199.105.195 1415	10 3 324 6	o 2.796 4	
		>	
Name resolution	Limit to display filter		
<u>Н</u> еlp <u>С</u> ору		Follow Stream	
10.83.123.104 1908 1921 3101		10.199.105.19 1415: first 1414: second 42496: third	5
٥			SHARE in Atlanta

#### **MQ-FTE Performance Problem**



n Atlanta

#### Filter on outbound packets using ip.ttl

d0223.#1-2000.pcap

10

ip.len greater than the standard  $\rightarrow$  Segmentation Offload

<u>Eile E</u> dit	<u>V</u> iew <u>G</u> o	<u>Capture</u>	e <u>A</u> nalyze	<u>S</u> tatistics	Teleph	on <u>y T</u> ools	Internals <u>H</u> e	elp						
			<b>3</b> ×	28	् 🤞	a 🔹 🎝	7 Ł I			¥	) 🖪 💥	a 📕		
Filter: ip.ttl	==200						✓ Expn	ession Clea	r Apply					
No. Time	e 001416	ip.ttl ip. 200	len 8932	ip.id 0xfe4b	src.port	dst.port	lin_flight 58032	tcp.ws 16383	tsval 29058498	Info 84 1921	> 1414	ĨPSH.		
78 0.	000040	200	12468	0xfe52	1921	1414	49152	16383	29058498	84 1921	> 1414	[PSH,	ACK	
80 0. 82 0.	051214	200 200	7180 9940	0xfe5c 0xfe61	1921 1921	1414 1414	49152	16383 16383	29058499 29058499	34 1921 35 1921	> 1414 > 1414	[PSH, [PSH,	ACK ACK	
84 0. 86 0.	001196	200	10892 19068	0x1e68 0xfe70	1921 1921	1414 1414 1414	49152 59288 40152	16383	29058499	36 1921 37 1921	> 1414 > 1414	[PSH, [PSH,	ACK	
90 0. 92 0.	051113 000595	200 200 200	7180 9940	Oxfe80 0xfe85	1921 1921 1921	1414 1414 1414	49152 49152 49152	16383 16383	29058499	87 1921 87 1921 88 1921	> 1414 > 1414 > 1414	[PSH, [PSH,		
94 0. 96 0.	001228	200	8076 2948	0xfe8c 0xfe92	1921 1921	1414 1414	46336 49232	16383 16383	29058499 29058499	89 1921 90 1921	<pre>&gt; 1414 &gt; 1414 &gt; 1414</pre>	[PSH, [PSH,	ACK ACK	
		MC d SE	ຊ Clier elays: GOFF	nt 51 ms <sup>:</sup> LOAD	)									
10.83.	123.1	04 1908 1921 3101	•								10 141 141 424	.199.  5: fir:  4: se  96: t	.105.198 st cond hird	5

## **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

d0223.#1-2000.pcap	Sequence number[B]	.#1-2000.pcap 10.63.123.104:	1921 -> 10.199.103.193:14 Tim	e/Sequence Graph (Stevens)		2			
Wireshark IO Graphs: d0223.#1-2000.pca	ър	_	_		_		_	- 🔀	
<				0.0s		5	10.0s	000000	• M 0
Graphs-						X Axis			bytes
Graph 1 Color Filter:	Calc	: SUM(*) 🗸 🗸	·	Style:	Line 🗸	Tick inter	val: 0.1 sec	<b>~</b>	0
Graph 2 Color Filter:	Calc	: SUM(*) 🛛 🗸		Style:	Line 🗸	Pixels per	tick: 2	~	0
Graph 3 Color Filter: tcp.dstport==1921	Calc	: MIN(*) 🔽	tcp.ack	Style:	Dot 🗸	U <u>V</u> iew	as time of day		
Graph 4 Color Filter: tcp.srcport==1921	Calc	: MIN(*) 🗸	tcp.seq	Style:	Impulse 🗸	Y Axis —	Advanced		
Graph 5 Color Filter:	Calc	: SUM(*) 🗸		Style:	Line 🗸	Scale:	Advanced Auto	~	
Help Copy						Save		e	J



#### **MQ-FTE Performance Problem - RTT**



Statistics  $\rightarrow$  TCP Stream Graph  $\rightarrow$  Round Trip Time

<pre>ite ist liver iso Capture instruct instruction in fight to put iterals lipelp iter iptim=200</pre>	d0223.#1-2000.pcap	
Image: pit pit pit pit pit pit pit stopot dt pot njøt too vs tord       From too vs to	<u>File Edit View Go Capture Analyze Statistics Telephony Tools Internals He</u>	Help
Inter       ptil       pld       stopot       data stop         20       0.001416       200       8932       0x5644b       1921       1414       58032       16383       2905549854       1921       1414       1915       16383       2905549854       1921       1414       49152       16383       2905549854       1921       1414       49152       16383       2905549854       1921       1414       49152       16383       2905549854       1921       1414       49152       16383       2905549894       1921       1414       49152       16383       2905549894       1921       1414       49152       16383       2905549894       1921       1414       49152       16383       2905549894       1921       1414       49152       16383       290549894       1921       1414       49152       16383       290549894       1921       1414       49152       16383       290549984       1921       1414       49152       16383       290549894       1921       1414       49152       16383       290549894       1921       1414       49152       16383       290549894       1921       1414       49152       16383       2905498494       1921       1414       49152       163	2 2 4 4 4 4 1 1 1 2 2 2 2 2 2 4 4 4 4 4	
2. The ptl plan pd srcool dspot in fight top us tand the fight top us tand top us top us the fight top us tand top us	Filter: ip.ttl==200	pression Clear Apply
<sup>1</sup> / <sub>0</sub> 0.001416 <sup>1</sup> / <sub>0</sub> 200 <sup>1</sup> / <sub>0</sub> 3932 <sup>1</sup> / <sub>0</sub> 444 <sup>1</sup> / <sub>0</sub>	No. Time lip.ttl lip.len lip.id lsrc.port ldst.port lin_flight	ltcp.ws Itsval Info
10 000000 200 12400 01422 1901 1444 9122 1033 2900490041921 91444 [PSH, ACK]         80 0.05214 200 7180 0xfest 1921 1414 49152 1033 29058493041921 >1414 [PSH, ACK]         82 0.00058 200 9940 0xfest 1921 1414 49152 1038 29058493041921 >1414 [PSH, ACK]         84 0.001196 200 10892 0xfest 1921 1414 49152 1038 29058493041921 >1414 [PSH, ACK]         90 0.051113 200 7180 0xfest 1921 1414 49152 1000 (reg (r) 0.05121.000 (reg (r) 0.05111.000 (reg (r) 0.05111.0000 (reg (r) 0.05111.000 (reg (r) 0.05111.00	76 0.001416 200 8932 0xte4b 1921 1414 58032 78 0.000040 200 12468 0xfo52 1921 1414 49152	$\frac{12}{16383} = \frac{16383}{2905849884} = \frac{1921}{1921} > \frac{1414}{195H} = \frac{16283}{400}$
82 0.000588       200       9940 0xfe61       1921       1414       47744         84 0.001196       200       19068 0xfe70       1921       1414       49152         90 0.051113       200       7180 0xfe80       1921       1414       49152         90 0.051113       200       7180 0xfe80       1921       1414       49152         90 0.051113       200       7180 0xfe80       1921       1414       49152         90 0.00134       200       2948 0xfe92       1921       1414       49152         90 0.00134       200       2948 0xfe92       1921       1414       49152         96 0.001334       200       2948 0xfe92       1921       1414       49152         98 0.000070       RTT: 54 ms SEGOFFLOAD TTL=200       Number of "bytes in flight": 49152-58032       49152-58032         0.83.123.104       1908       1921       1921       1921       1921       1921         3101       9000000000000000000000000000000000000	80 0.051214 200 7180 0xfe5c 1921 1414 49152	52  16383  2905849864 1921 > 1414 [PSH, ACK]
84 0.001196       200       10892 0xfe68       1921       1414       49152         86 0.000057       200       2332 0xfe70       1921       1414       49152         90 0.051113       200       7180 0xfe80       1921       1414       49152         92 0.000595       200       9940 0xfe85       1921       1414       49152         92 0.000595       200       9940 0xfe85       1921       1414       49152         94 0.00128       200       8076 0xfe8c       1921       1414       49152         96 0.000134       200       2948 0xfe82       1921       1414       49232         96 0.000134       200       2948 0xfe82       1921       1414       49232         030       TTL=200       NQ Client       RTT: 54 ms       SEGOFFLOAD         TTL=200       Name       1908       1921       1908       1921         3101       908       1921       900       908       9000       908       9000         1921       3101       908       902       908       9000       908       9000         1921       3101       908       9000       9000       90000       90000       90000	82 0.000588 200 9940 0xfe61 1921 1414 47744	4 🔽 TCP Graph 2: d0223.#1-2000.pcap 10.83.123.104:1921 -> 10.199.105.195:1414
86 0.001496       200       19068 0xfe70       1921       1414       49152         90 0.051113       200       7180 0xfe80       1921       1414       49152         92 0.000595       200       9940 0xfe80       1921       1414       49152         94 0.001228       200       8076 0xfe80       1921       1414       49152         94 0.00128       200       2948 0xfe92       1921       1414       49232         08 0.00030       200       2948 0xfe92       1921       1414       49232         08 0.00030       200       2948 0xfe92       1921       1414       49232         08 0.00030       200       2948 0xfe92       1921       1414       49152         080 0.00030       70       71268 0xfe4u       1031       1414       49232         080 0.00030       70       71268 0xfe4u       1031       1414       49152         080 0.00030       70       71268 0xfe4u       1031       1414       49152         080 0.00030       70       71268 0xfe4u       1031       1414       49152         081 0.122       90       90       9152-58032       9152-58032         081 0.122       91       91	84 0.001196 200 10892 0xfe68 1921 1414 49152	52 Round Trip Time Graph
88 0.000057       200       2332 0xte7e 1921       1414       49152         90 0.051113       200       7180 0xte85 1921       1414       49152         94 0.001228       200       8076 0xte8c 1921       1414       49152         94 0.001334       200       2948 0xte8c 1921       1414       49152         96 0.00134       200       2948 0xte8c 1921       1414       49152         96 0.00134       200       2948 0xte8c 1921       1414       49152         97 0.00037       200       2948 0xte8c 1921       1414       49152         98 0.000037       200       2948 0xte8c 1921       1414       49152         98 0.00033       200       2948 0xte8c 1921       1414       49152         98 0.00030       200       2948 0xte8c 1921       1414       49152         98 0.00030       200       2948 0xte8c 1921       1414       49152         900       200       2948 0xte8c 1921       1414       49152         900       TTL=200       TTL=200       200       20152-580032         91921       3101       50000       50000       50000       50000         900       1921       3101       50000       50000       <	86 0.001496 200 19068 0xfe70 1921 1414 59288	38
90.0001113       200       7180 001680       1921       1414       4912         92.0.000595       200       8076 0xfe8c       1921       1414       4912         94.0.001228       200       8076 0xfe8c       1921       1414       49336         96.0.001534       200       2948 0xfe92       1921       1414       49132         98.0       0.001334       200       2948 0xfe92       1921       1414       49132         98.0       MQ Client       RTT: 54 ms       SEGOFFLOAD       Number of "bytes in flight": 49152-58032         1908       1921       3101	88 0.000057 200 2332 0xte7e 1921 1414 49152	
94 0.0001228       200       8076 0xfe8c 1921       1414       46326         96 0.001234       200       2948 0xfe92       1921       1414       49322         98 0       0.00030       200       21368 0xfe92       1921       1414       49322         98 0       0.00030       200       21368 0xfe92       1921       1414       49322         98 0       0.00030       200       21368 0xfe92       1921       1414       49322         98 0       0.00030       200       21368 0xfe92       1921       1414       49322         98 0       0.00030       200       21368 0xfe92       1921       1414       49322         98 0       MQ Client       RTT: 54 ms       5EGOFFLOAD       TTL=200       000-         0.83.123.104       1908       000-       000-       000-       000-         1908       1921       000-       000-       000-       000-       000-         1921       3101       000-       50000       50000       50000       50000         90000       9000       9000       9000       90000       9000       90000       9000         19008       19000       90000	90 0.051113 200 /180 0X1680 1921 1414 49152 92 0.000595 200 9940 0xfe85 1921 1414 49152	22 0.000
96 0.001534 200 2948 0xfe92 1921 1414 49232 98 0 000030 200 21268 0xfe92 1921 1414 49232 98 0 000030 200 21268 0xfe94 1921 1414 49152 MQ Client RTT: 54 ms SEGOFFLOAD TTL=200 0.83.123.104 1908 1921 3101 - RTT = 54 ms Number of "bytes in flight": 49152-58032 Secure Number[]	94 0.001228 200 8076 0xfe8c 1921 1414 46336	36
<b>98 0 000030</b> 200       21268 0vf eqd 1921       1414       49152 <b>MQ Client</b> RTT: 54 ms SEGOFFLOAD TTL=200 <b>RTT = 54 ms</b> Number of "bytes in flight": 49152-58032         0.83.123.104       1908 1921 3101       1908         1921 3101	96 0.001534 200 2948 0xfe92 1921 1414 49232	0.045
MQ Client RTT: 54 ms SEGOFFLOAD TTL=200 0.83.123.104 1908 1921 3101 MQ Client RTT = 54 ms Number of "bytes in flight": 49152-58032 Second Lience Number 500000 Second Lience Number 50000 Second Lience Number 500000 Second Lience	98 0 000030 200 21268 0xfe94 1921 1414 49152	<b>57</b> 0.040 —
MQ Client RTT: 54 ms SEGOFFLOAD TTL=200 0.83.123.104 1908 1921 3101 MQ Client RTT = 54 ms Number of "bytes in flight": 49152-58032 Sequence Number[5]	and the state	0.035
RTT: 54 ms SEGOFFLOAD TTL=200 0.83.123.104 1921 3101 RTT: 54 ms SEGOFFLOAD TTL=200 0.83.123.104 1921 3101 RTT – 54 MS Number of "bytes in flight": 49152-58032	MQ Client	DTT = 54 mo
SEGOFFLOAD TTL=200 0.83.123.104 1908 1921 3101	RTT: 54 ms	0.025 - RII - 34 IIIS
TTL=200     0.015     49152-58032       1908     0.015     0.015       1921     0.015     0.005       3101     Sequence Number[B]	SEGOEFLOAD	Number of "bytes in flight":
0.83.123.104 1908 1921 3101		
1908 1921 3101	0.83.103.104	49152-58032
1908 1921 3101  Sequence Number[B]	1000	0,010
1921 3101 Sequence Number[B]	1908	0.005
3101 Sequence Number[B]	1921	
SHARE in Atlanta	3101	Suuuuuu Sequence Number[B]
SHARE in Atlanta		No. 20
SHARE in Atlanta		
		: SHARE in Atlanta

### IO Graph – bytes in flight vs adv. windowsize



Outbound Bytes in flight – inbound advertised windowsize

	m M	www.///	
0.0s	20s	40s 6(	la 80a 100a
₹			
- Graphs			X Axis
Graph 1 Color Filter:	Calc: SUM(*)	~	Style: Line V Tick interval: 0.1 sec V
Granda 2 Calara Elitaria			
		·	
Graph 3 Color Filter: tcp.dstport==1921	Cale: MIN(*)	✓ tcp.window_size	<b>Bytes in Tilght: 45.000-50.000</b>
Graph 4 Color Filter: tcp.srcport==1921	Calc: MIN(*)	✓ tcp.analysis.bytes_in_flight	M Adv Window size: 295-384
Graph 5. Color Eilter			
11 0.001000 00 02 084040	1414 1961	290 1000147044 1414	2 194
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 Window Scale
21 0.051967 58 52 0x4350	1414 1921	384 1053147099 1414	> 1921
23 0.000938 58 52 0x4351	1414 1921	306 1053147100 1414	> 1921 Factor:128
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	3/3 105314/153 1414	> 1921
31 0.000003 58 52 0x4350	1414 1921	384 1053147153 1414	> 1921
55 0.0008/9 58 52 0X435/	1414 1921	584 1053147154 1414	> 1921
55 0.001344 56 52 0X4358 27 0.000081 58 52 0X4350	1414 1921	295 105514/155 1414	1021
38.0.000002 58 52.0v4359	1414 1921	384 1053147156 1414	1921
∎ 50 0.00002 50 52 0X455a	1414 1961	564 1055147150 1414	SHARE in Atlanta

## MQ-FTE Performance Problem – Remote Host

Filter on inbound flows – Identify the remote system

	1022	3.#1-2	000.p	осар													_ 🗆 🔀	
Eile	E	dit <u>V</u> ie	W	<u>G</u> o <u>C</u>	apture	<u>A</u> nalyze	$\underline{S} tatistics$	Telephon	<u>y T</u> ools <u>I</u> r	nternal	s <u>H</u> elp							
	ë	1	<b>@</b> [			3 🗙 🕯	28	୍ଦ୍ 🖕	🕸 🌍 🕇	F 🕹		) <del>(</del>	$\Theta$	**	<b>X</b>	<b>1</b>	6 🛛 🖾	
Filte	r: t	cp.dstp	ort==	1921						~	Expression	Clear	Apply					
No.	1	Time	lip	.ttl li	p.len	lip.id	src.port	dst.port	tcp.ws	tsval	21 4 8000	Info	1001	TACKI	5 og 1	Aclas	1721240	
2	94	0.000	22	58	52	0x4413	1/1/	1021	284	105	21/8002	1414 >	1021	[ACK]	Seq=1	ACK=	1721249	
2	98	0.051	12	58	52	0x4415	1414	1921	304	105	3148954	1414	> 1921	[ACK]	Seg=1	Ack=	1740969	
4	00	0.000	06	58	52	0x4416	1414	1921	384	105	3148954	1414	> 1921	[ACK]	Seq=1	Ack=	1759985	
4	01	0.000	00	58	52	0x4417	1414	1921	384	105	3148954	1414 >		5				
4	03	0.000	04:	58	52	0x4418	1414	1921	373	105	3148955	1414 >	<b>,</b>		qi	∙.ttl=	=58,	
4	05	0.002	22	58	52	0x4419	1414	1921	384	105	3148957	1414 >		, id	י דד מי	or to		ootion
4	07	0.051	L2	58	52	0x441a	1414	1921	384	105	3149008	1414 >	, դ	).iu	ττ p		p conne	
4	09	0.000	06	58	52	0x441b	1414	1921	306	105	3149009	1414 >	TS	VAL	_ incr	eme	ents at 1	1000Hz
4	11	0.000	00	58	52	0x441c	1414	1921	384	105	3149009	1414 >						
				MQ RT SEC	Clier T: 54 GOFF TL=2	nt ms FLOAD 00						MQ S ip.id- TSV/ TTL=	Server ++/tcp AL: 100 =64	0 Hz			Tiny	
10.8	33.	123	.10	4	X				23	$\gtrsim$	20		10.19	99.10	)5.195	5	Window	<mark>/size</mark>
			1	908									1415:	first				
	8	50kb	/s 1	921									1414:	seco	nd			
			3	101	•								42496	6: thire	d			
15																	SHAP	RE in Atlanta



### **MQ-FTE Performance Problem – IO Graph**

Outbound seq# - inbound ack#



### **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!**



n Atlanta



#### **EE Education – IP wizards**

![](_page_18_Picture_2.jpeg)

#### **ITS53 EE Implementation and Problem Determination**

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

#### Register at http://greenhouse.lotus.com

Lotus Greenhouse<sup>™1</sup> Solutions Products Labs FAQs Sign Up Log In

Join the IP wizards community

http://tinyurl.com/ipwizards

![](_page_18_Picture_9.jpeg)

to download wireshark profiles and p0f fingerprints

![](_page_18_Picture_11.jpeg)

#### **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.

![](_page_19_Picture_6.jpeg)

![](_page_19_Picture_7.jpeg)

![](_page_19_Picture_8.jpeg)