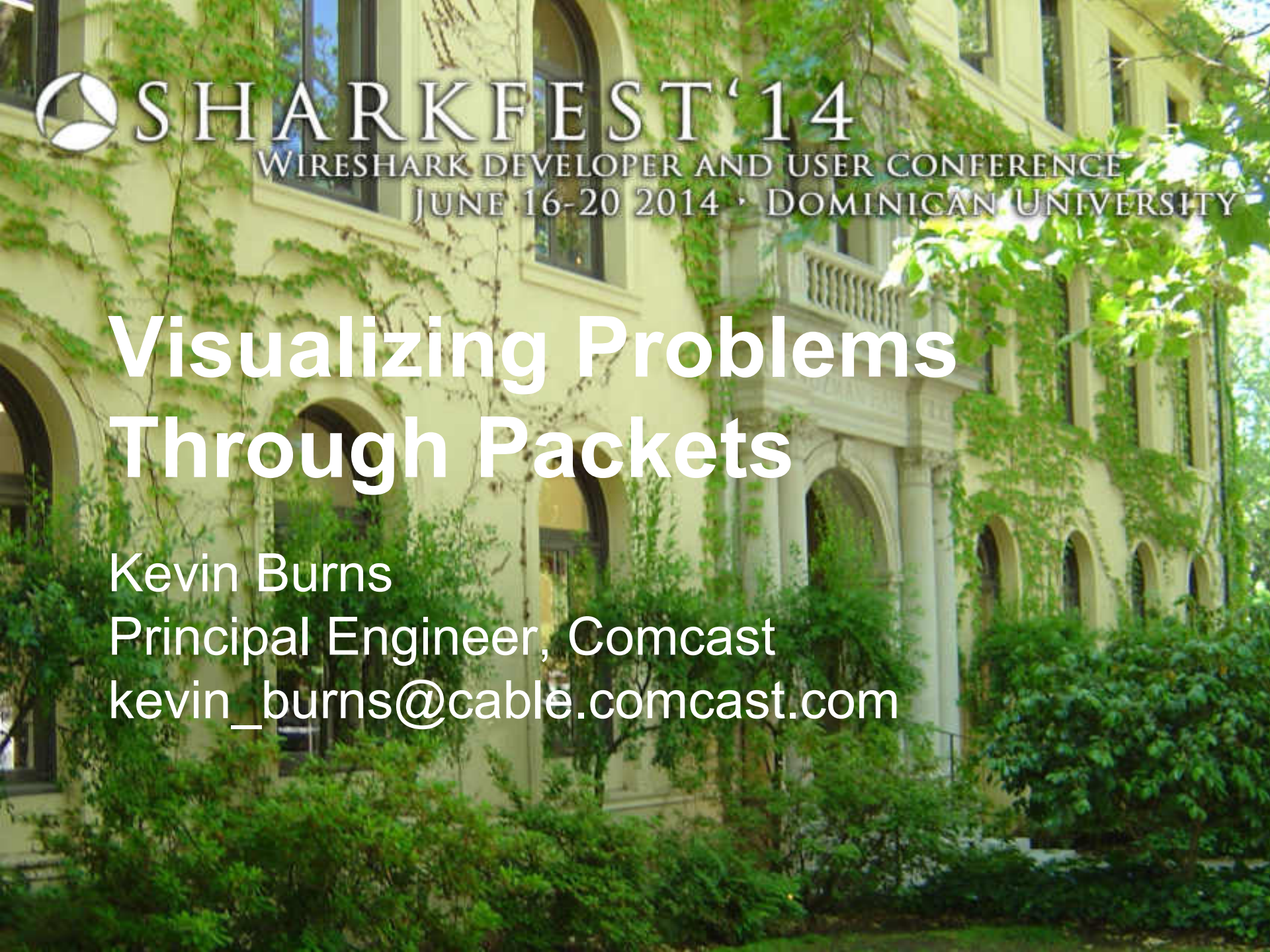


Visualizing Problems Through Packets

Kevin Burns
Principal Engineer, Comcast
kevin_burns@cable.comcast.com

Comcast Background

15M+ OnDemand Views

12M+ Voicemails Received

145M+ Emails Delivered

71 Million IP Addresses

220 Million Menu Navigations/day

270+ Billion DNS Look-Ups

184M+ Phone Calls

8 Million Wi-Fi hotspots (end 2014)

Why am I talking about Visualization?

- Troubleshooting is more of an art than a science. This presentation is about how I describe my own “art”. Everyone will develop their own art (ie: methodologies).
- Nobody can teach you this. They can only help you learn to how to incorporate ideas and techniques into your own art.
- A lot can be gained from looking at different types of thinking and methods to incorporate into your own set of tools and techniques.
- Visualizing problems is the most common process people are involved in during a troubleshooting effort.
- To be a successful problem solver you need to understand how the components of visualization fit together.

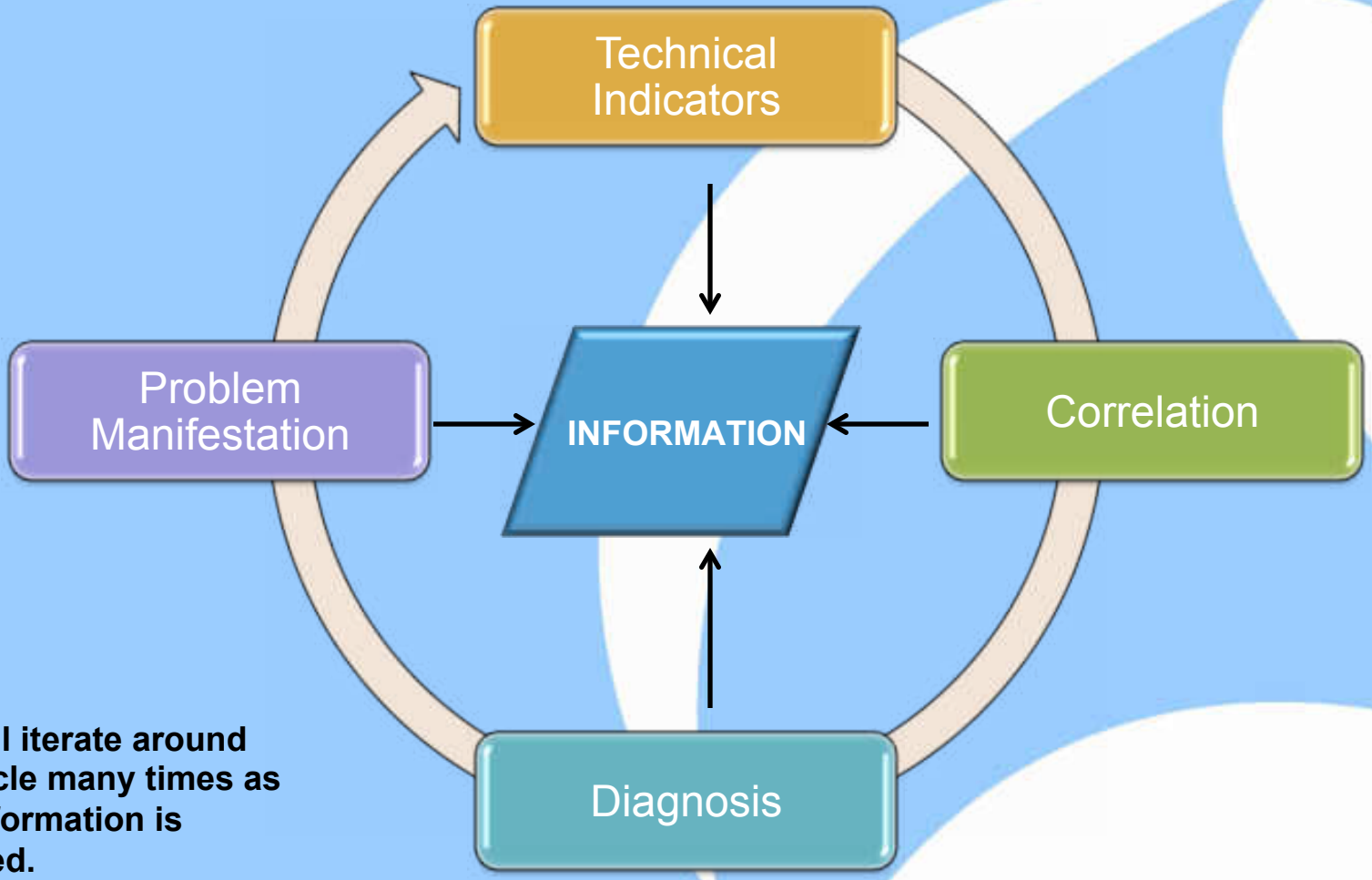
Understanding Visualization Components

- In order to visualize a problem we must:
  - Understand how the problem manifests itself to it's users, engineers and inside of packet captures.
  - Categorize the problem based on it's manifestation behavior to users and protocol interactions
  - Determine what technical indicators exist that allow us to correlate information to visualize the problem.
- The goal of visualization is to determine how a problem manifests itself and correlate it's technical indicators to produce a diagnosis.
- Visualization is about seeing and recognizing patterns on several different levels.
- Problem solving is about utilizing visualization techniques to resolve an issue.

Components of Visualization

- Problem Manifestation
  - The outward or perceptible indication of a problem.
  - Determine how the problem manifests inside of a packet capture.
  - Categorize of the problem and it's behavior.
- Technical Indicators
  - Characteristics of a problem's manifestation.
  - Identify a problem's technical indicators
- Correlation
  - Correlation of various technical indicators.
  - Correlate technical indicators with a problem's manifestation
  - Look for repeatable patterns.
- Diagnosis
  - The foundation of a definitive diagnosis is based on correlation of a problem's manifestation and it's technical indicators.

Problem Solving Cycle



You will iterate around this cycle many times as new information is gathered.

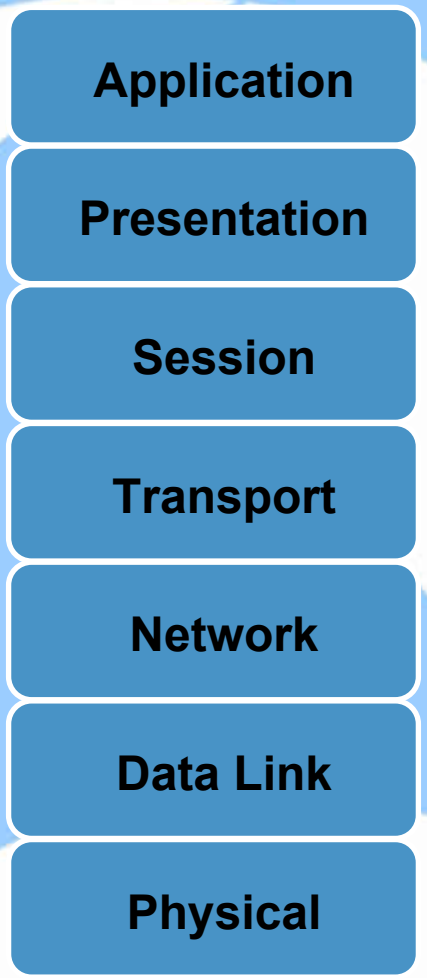
Problem Manifestation

- How is it known the problem exists?
- How is the problem viewed?
  - By users
  - By engineers
  - In packets
- What technical indicators does the problem manifest itself through?
  - Retransmissions
  - Time-outs, Delays
  - Application Messages
- What tools can help you uncover more methods of how the problem manifests itself?
- What techniques can you use to look for patterns?
- Understand how different technical indicators relate to impact.

Often different perspectives (at first)

Problem Categorization by OSI Model

- Problems will manifest themselves in one or more layers of the OSI Model.
- Problems are almost always isolated to a single layer.
- The first and most important step in troubleshooting is to determine what layer of the OSI model the problem lives in. If you don't want to understand the OSI model at least understand the protocol dependancies you are dealing with
- OSI teaches us about dependancies, that is why it's useful.



Problem Categorization by Type

- Loss of Connectivity
  - Complete and total loss of end to end connectivity at one or more layers.
  - Application failures, TCP Resets, Ping failures
- Intermittent Connectivity
  - Inconsistent end to end connectivity at one or more layers.
  - Dropped packets, sessions
- Degraded Performance
  - End to end connectivity is good but performance over the connection is suffering
  - Low Throughput, Latency impact
- Unknown
  - Technical indicators are unknown.

Case Study: Manifestation

Application

Remedy Ticketing System

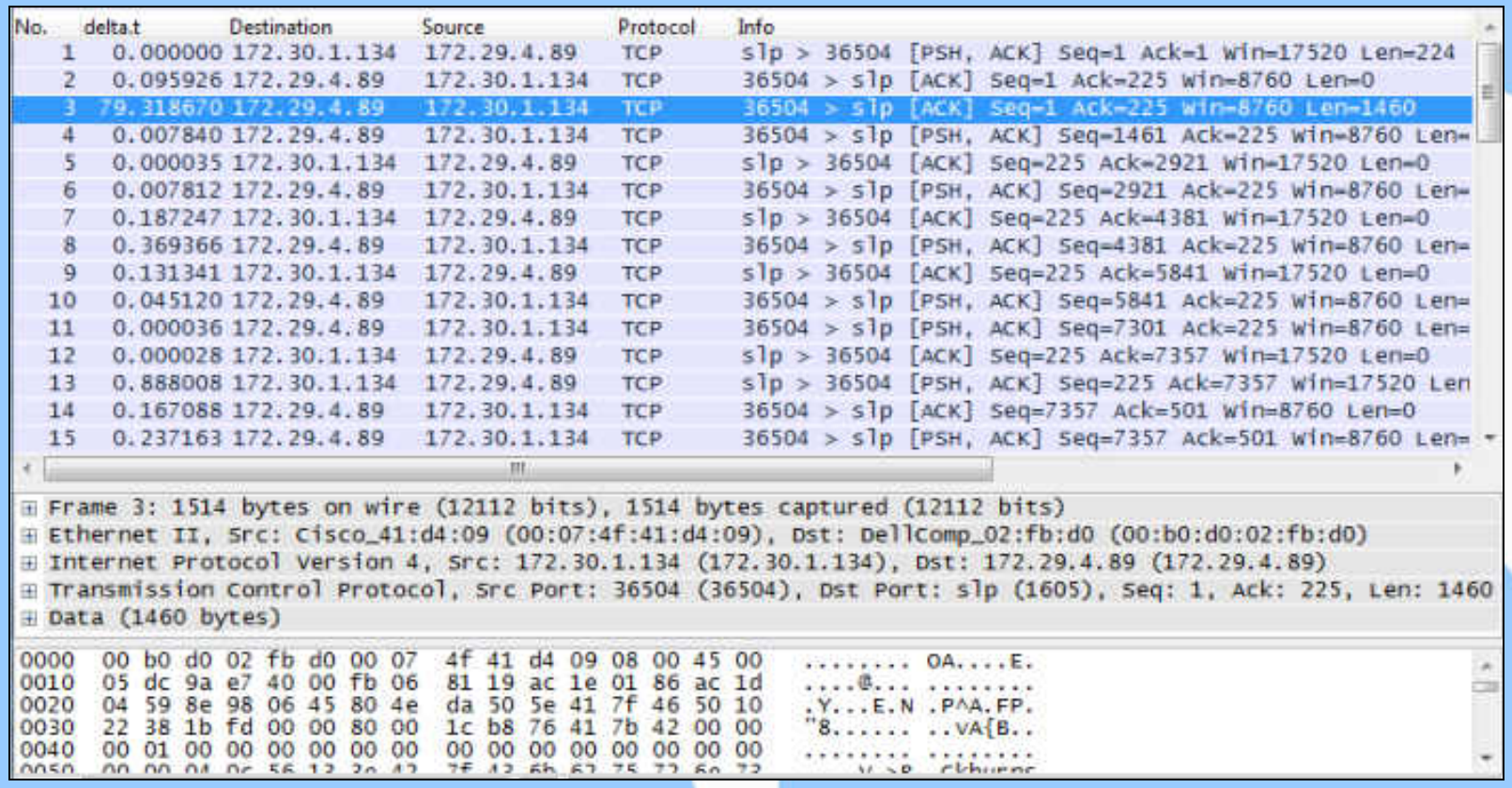
Symptoms

- User experiencing minute long delays when performing lookups.
- Network path appears to be clean. No loss or latency.

Manifestation

- Problem manifests as delay
- Delay is obvious in the packets

Case Study: Remedy Ticketing System



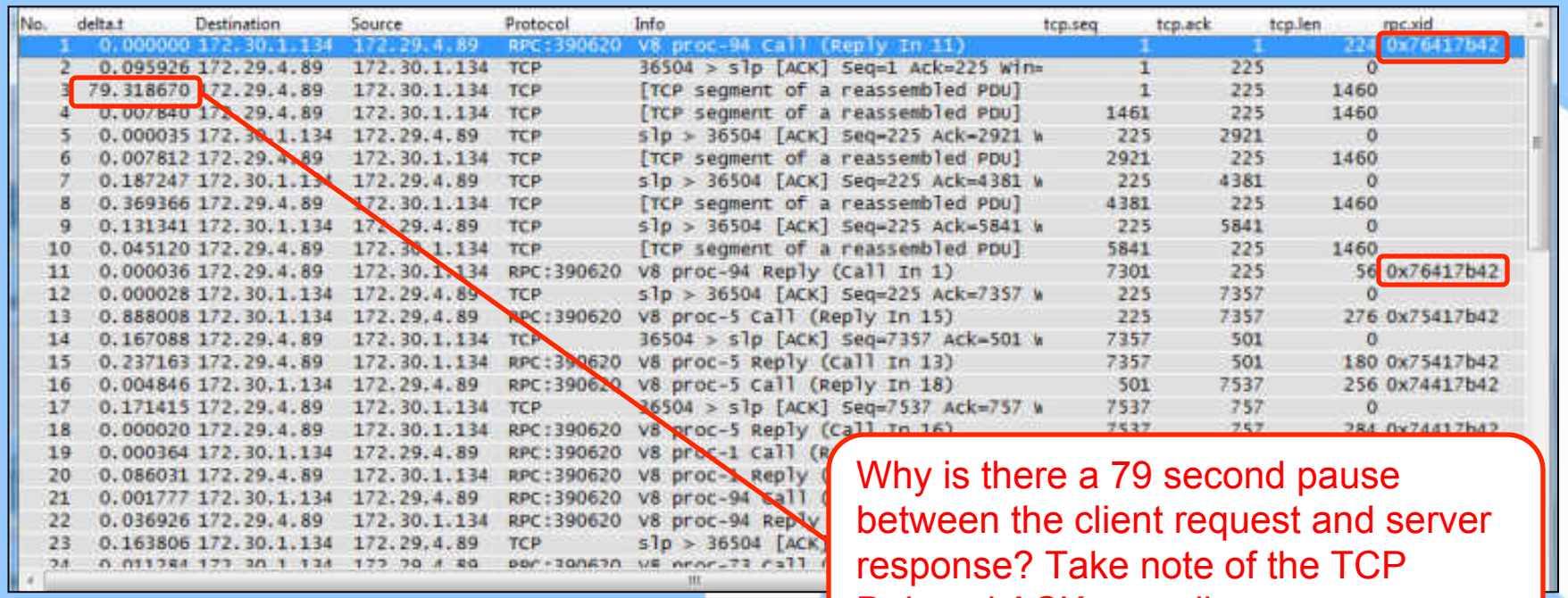
No.	delta.t	Destination	Source	Protocol	Info
1	0.000000	172.30.1.134	172.29.4.89	TCP	slp > 36504 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=224
2	0.095926	172.29.4.89	172.30.1.134	TCP	36504 > slp [ACK] Seq=1 Ack=225 Win=8760 Len=0
3	79.318670	172.29.4.89	172.30.1.134	TCP	36504 > slp [ACK] Seq=1 Ack=225 Win=8760 Len=1460
4	0.007840	172.29.4.89	172.30.1.134	TCP	36504 > slp [PSH, ACK] Seq=1461 Ack=225 Win=8760 Len=
5	0.000035	172.30.1.134	172.29.4.89	TCP	slp > 36504 [ACK] Seq=225 Ack=2921 Win=17520 Len=0
6	0.007812	172.29.4.89	172.30.1.134	TCP	36504 > slp [PSH, ACK] Seq=2921 Ack=225 Win=8760 Len=
7	0.187247	172.30.1.134	172.29.4.89	TCP	slp > 36504 [ACK] Seq=225 Ack=4381 Win=17520 Len=0
8	0.369366	172.29.4.89	172.30.1.134	TCP	36504 > slp [PSH, ACK] Seq=4381 Ack=225 Win=8760 Len=
9	0.131341	172.30.1.134	172.29.4.89	TCP	slp > 36504 [ACK] Seq=225 Ack=5841 Win=17520 Len=0
10	0.045120	172.29.4.89	172.30.1.134	TCP	36504 > slp [PSH, ACK] Seq=5841 Ack=225 Win=8760 Len=
11	0.000036	172.29.4.89	172.30.1.134	TCP	36504 > slp [PSH, ACK] Seq=7301 Ack=225 Win=8760 Len=
12	0.000028	172.30.1.134	172.29.4.89	TCP	slp > 36504 [ACK] Seq=225 Ack=7357 Win=17520 Len=0
13	0.888008	172.30.1.134	172.29.4.89	TCP	slp > 36504 [PSH, ACK] Seq=225 Ack=7357 Win=17520 Len
14	0.167088	172.29.4.89	172.30.1.134	TCP	36504 > slp [ACK] Seq=7357 Ack=501 Win=8760 Len=0
15	0.237163	172.29.4.89	172.30.1.134	TCP	36504 > slp [PSH, ACK] Seq=7357 Ack=501 Win=8760 Len=

```
⊞ Frame 3: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
⊞ Ethernet II, Src: Cisco_41:d4:09 (00:07:4f:41:d4:09), Dst: DellComp_02:fb:d0 (00:b0:d0:02:fb:d0)
⊞ Internet Protocol Version 4, Src: 172.30.1.134 (172.30.1.134), Dst: 172.29.4.89 (172.29.4.89)
⊞ Transmission Control Protocol, Src Port: 36504 (36504), Dst Port: slp (1605), Seq: 1, Ack: 225, Len: 1460
⊞ Data (1460 bytes)
```

```
0000  00 b0 d0 02 fb d0 00 07  4f 41 d4 09 08 00 45 00   ........ OA....E.
0010  05 dc 9a e7 40 00 fb 06  81 19 ac 1e 01 86 ac 1d   ....@... ........
0020  04 59 8e 98 06 45 80 4e  da 50 5e 41 7f 46 50 10   .Y...E.N .P^A.FP.
0030  22 38 1b fd 00 00 80 00  1c b8 76 41 7b 42 00 00   "8...... ..vA{B..
0040  00 01 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0050  00 00 04 0c 56 13 3e 42  7f 43 6b 67 75 72 6e 73      V.>B .Ckburns
```

What do you see as the manifestation of the problem?
Does it correlate with the user experience?

Visualizations and Technical Indicators



No.	delta.t	Destination	Source	Protocol	Info	tcp.seq	tcp.ack	tcp.len	rpc.xid
1	0.000000	172.30.1.134	172.29.4.89	RPC:390620	V8 proc-94 Call (Reply In 11)	1	1	22	0x76417b42
2	0.095926	172.29.4.89	172.30.1.134	TCP	36504 > slp [ACK] Seq=1 Ack=225 Win=	1	225	0	
3	79.318670	172.29.4.89	172.30.1.134	TCP	[TCP segment of a reassembled PDU]	1	225	1460	
4	0.007840	172.29.4.89	172.30.1.134	TCP	[TCP segment of a reassembled PDU]	1461	225	1460	
5	0.000035	172.30.1.134	172.29.4.89	TCP	slp > 36504 [ACK] Seq=225 Ack=2921 W	225	2921	0	
6	0.007812	172.29.4.89	172.30.1.134	TCP	[TCP segment of a reassembled PDU]	2921	225	1460	
7	0.187247	172.30.1.134	172.29.4.89	TCP	slp > 36504 [ACK] Seq=225 Ack=4381 W	225	4381	0	
8	0.369366	172.29.4.89	172.30.1.134	TCP	[TCP segment of a reassembled PDU]	4381	225	1460	
9	0.131341	172.30.1.134	172.29.4.89	TCP	slp > 36504 [ACK] Seq=225 Ack=5841 W	225	5841	0	
10	0.045120	172.29.4.89	172.30.1.134	TCP	[TCP segment of a reassembled PDU]	5841	225	1460	
11	0.000036	172.29.4.89	172.30.1.134	RPC:390620	V8 proc-94 Reply (Call In 1)	7301	225	56	0x76417b42
12	0.000028	172.30.1.134	172.29.4.89	TCP	slp > 36504 [ACK] Seq=225 Ack=7357 W	225	7357	0	
13	0.888008	172.30.1.134	172.29.4.89	RPC:390620	V8 proc-5 Call (Reply In 15)	225	7357	276	0x75417b42
14	0.167088	172.29.4.89	172.30.1.134	TCP	36504 > slp [ACK] Seq=7357 Ack=501 W	7357	501	0	
15	0.237163	172.29.4.89	172.30.1.134	RPC:390620	V8 proc-5 Reply (Call In 13)	7357	501	180	0x75417b42
16	0.004846	172.30.1.134	172.29.4.89	RPC:390620	V8 proc-5 Call (Reply In 18)	501	7537	256	0x74417b42
17	0.171415	172.29.4.89	172.30.1.134	TCP	36504 > slp [ACK] Seq=7537 Ack=757 W	7537	757	0	
18	0.000020	172.29.4.89	172.30.1.134	RPC:390620	V8 proc-5 Reply (Call In 16)	7537	757	284	0x74417b42
19	0.000364	172.30.1.134	172.29.4.89	RPC:390620	V8 proc-1 Call (R				
20	0.086031	172.29.4.89	172.30.1.134	RPC:390620	V8 proc-1 Reply (				
21	0.001777	172.30.1.134	172.29.4.89	RPC:390620	V8 proc-94 Call (				
22	0.036926	172.29.4.89	172.30.1.134	RPC:390620	V8 proc-94 Reply				
23	0.163806	172.30.1.134	172.29.4.89	TCP	slp > 36504 [ACK]				
24	0.011284	172.30.1.134	172.29.4.89	RPC:390620	V8 proc-73 Call (				

Why is there a 79 second pause between the client request and server response? Take note of the TCP Delayed ACK as well.

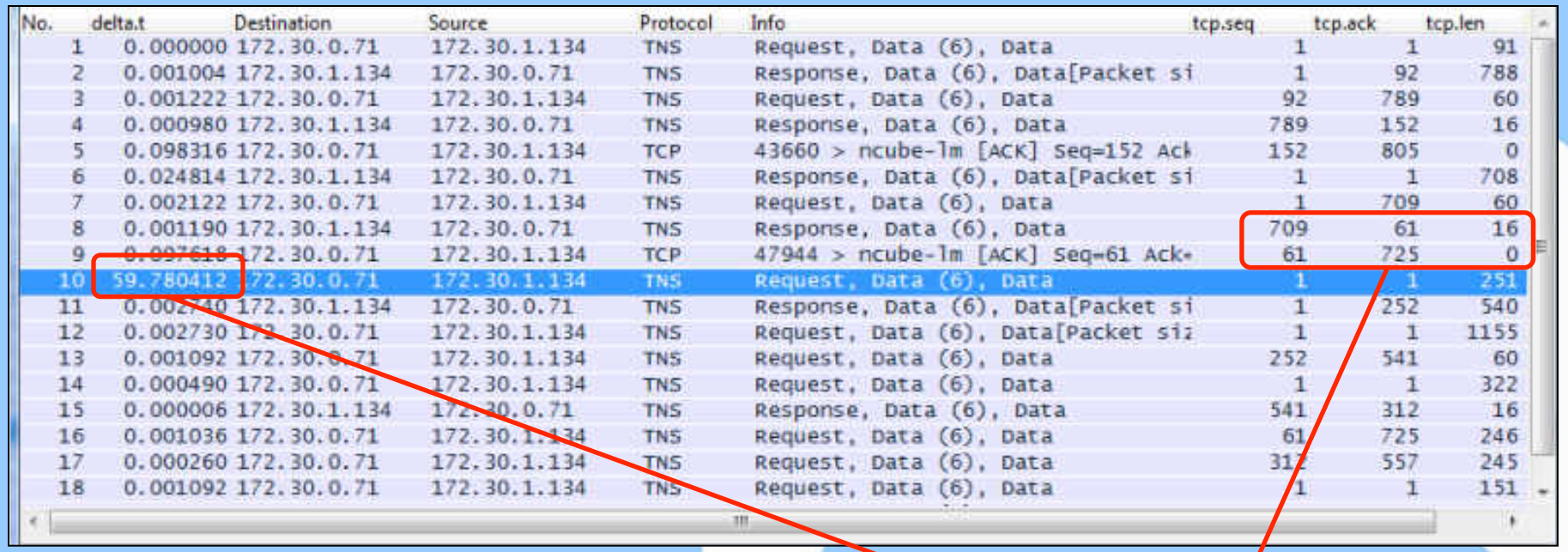
Visualization Techniques:

Protocol Decode (forced to RPC)
TCP SEQ+LEN=ACK
Application Transaction ID Column

Technical Indicators:

TCP ACK
TCP Delayed ACK
Application Delay

Correlation of Technical Indicators



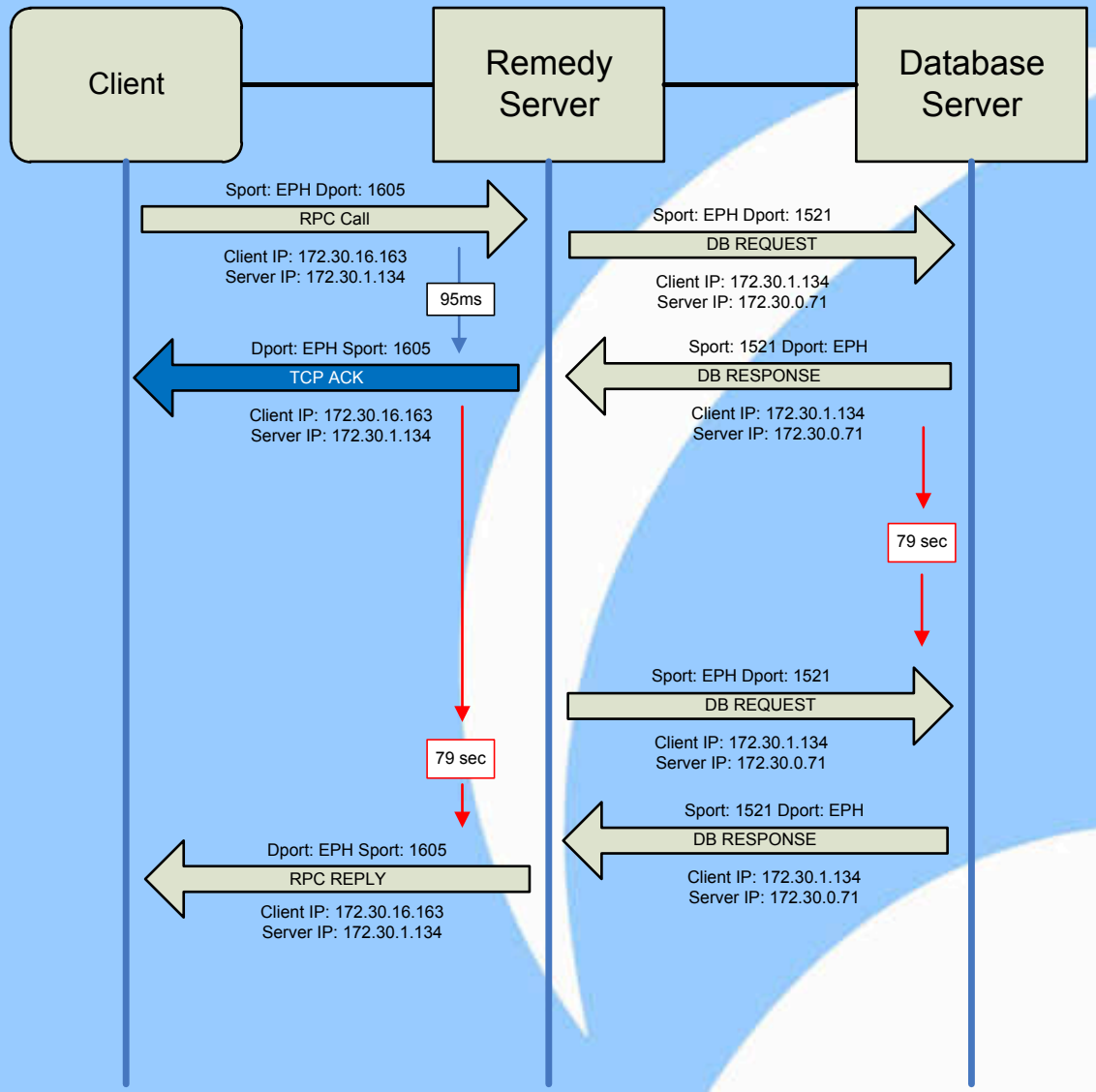
No.	delta.t	Destination	Source	Protocol	Info	tcp.seq	tcp.ack	tcp.len
1	0.000000	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data	1	1	91
2	0.001004	172.30.1.134	172.30.0.71	TNS	Response, Data (6), Data[Packet si	1	92	788
3	0.001222	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data	92	789	60
4	0.000980	172.30.1.134	172.30.0.71	TNS	Response, Data (6), Data	789	152	16
5	0.098316	172.30.0.71	172.30.1.134	TCP	43660 > ncube-lm [ACK] Seq=152 Ack	152	805	0
6	0.024814	172.30.1.134	172.30.0.71	TNS	Response, Data (6), Data[Packet si	1	1	708
7	0.002122	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data	1	709	60
8	0.001190	172.30.1.134	172.30.0.71	TNS	Response, Data (6), Data	709	61	16
9	0.097618	172.30.0.71	172.30.1.134	TCP	47944 > ncube-lm [ACK] Seq=61 Ack=	61	725	0
10	59.780412	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data	1	1	251
11	0.002740	172.30.1.134	172.30.0.71	TNS	Response, Data (6), Data[Packet si	1	252	540
12	0.002730	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data[Packet siz	1	1	1155
13	0.001092	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data	252	541	60
14	0.000490	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data	1	1	322
15	0.000006	172.30.1.134	172.30.0.71	TNS	Response, Data (6), Data	541	312	16
16	0.001036	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data	61	725	246
17	0.000260	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data	312	557	245
18	0.001092	172.30.0.71	172.30.1.134	TNS	Request, Data (6), Data	1	1	151

Technical Indicators:

Delay
TCP SEQ+LEN=ACK
Application Request/Response Behavior

Why does the Remedy Server stop talking to the Database for 59 seconds after ACKing all responses???

End to End Visualization

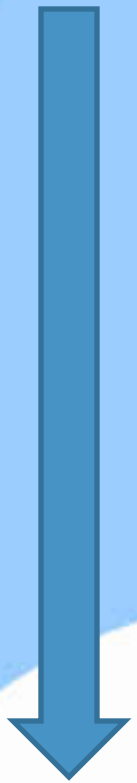


What are Technical Indicators?

- Assuming the correct packets have been captured, the problem will always exist inside of the packets.
- Technical Indicators are feedback mechanisms found in packet communications. *(sometimes you really have to dig for them)*
- They are not symptoms.
  - *I tend to avoid using the word symptom as people tend to associate it with being the cause.*
- Problems will exist inside of packets in several ways
  - Explicit packet feedback mechanisms
  - Implicit packet feedback mechanisms
  - Extrapolated Data and Measurements
  - Behavior and Relationship Based (Correlation)

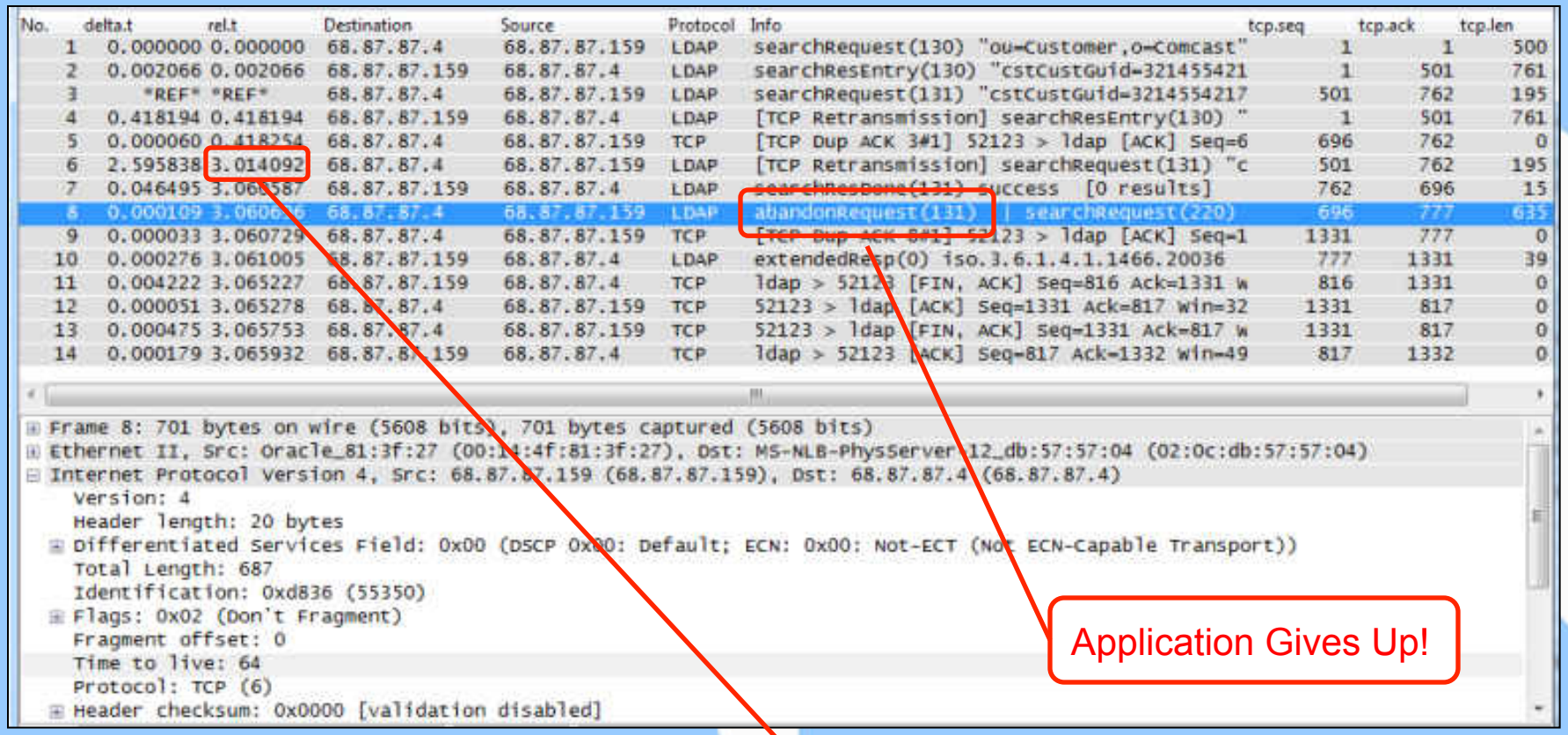
Feedback Mechanisms

- Asssuming the correct packets have been captured, the problem will always exist inside of the packets.
- Problems will exist inside of packets in several ways
  - Explicit packet feedback mechanisms:
    - TCP (FIN, RST)
    - Application Messages
    - ICMP return types/codes.
  - Implicit packet feedback mechanisms:
    - Timing
    - Behavior
    - Other Correlative Factors
  - Extrapolated Data and Measurements
    - Latency
    - Throughtput
    - Examples, Behavior, Relationships



Complexity Increases

Explicit Feedback Mechanisms



No.	delta.t	rel.t	Destination	Source	Protocol	Info	tcp.seq	tcp.ack	tcp.len
1	0.000000	0.000000	68.87.87.4	68.87.87.159	LDAP	searchRequest(130) "ou=Customer,o=Comcast"	1	1	500
2	0.002066	0.002066	68.87.87.159	68.87.87.4	LDAP	searchResEntry(130) "cstCustGuid=321455421	1	501	761
3	*REF*	*REF*	68.87.87.4	68.87.87.159	LDAP	searchRequest(131) "cstCustGuid=3214554217	501	762	195
4	0.418194	0.418194	68.87.87.159	68.87.87.4	LDAP	[TCP Retransmission] searchResEntry(130) "	1	501	761
5	0.000060	0.418254	68.87.87.4	68.87.87.159	TCP	[TCP Dup ACK 3#1] 52123 > ldap [ACK] Seq=6	696	762	0
6	2.595838	3.014092	68.87.87.4	68.87.87.159	LDAP	[TCP Retransmission] searchRequest(131) "c	501	762	195
7	0.046495	3.060587	68.87.87.159	68.87.87.4	LDAP	searchResDone(131) success [0 results]	762	696	15
8	0.000109	3.060696	68.87.87.4	68.87.87.159	LDAP	abandonRequest(131) | searchRequest(220)	696	777	635
9	0.000033	3.060729	68.87.87.4	68.87.87.159	TCP	[TCP Dup ACK 8#1] 52123 > ldap [ACK] Seq=1	1331	777	0
10	0.000276	3.061005	68.87.87.159	68.87.87.4	LDAP	extendedResp(0) iso.3.6.1.4.1.1466.20036	777	1331	39
11	0.004222	3.065227	68.87.87.159	68.87.87.4	TCP	ldap > 52123 [FIN, ACK] Seq=816 Ack=1331 w	816	1331	0
12	0.000051	3.065278	68.87.87.4	68.87.87.159	TCP	52123 > ldap [ACK] Seq=1331 Ack=817 Win=32	1331	817	0
13	0.000475	3.065753	68.87.87.4	68.87.87.159	TCP	52123 > ldap [FIN, ACK] Seq=1331 Ack=817 w	1331	817	0
14	0.000179	3.065932	68.87.87.159	68.87.87.4	TCP	ldap > 52123 [ACK] Seq=817 Ack=1332 Win=49	817	1332	0

```
Frame 8: 701 bytes on wire (5608 bits), 701 bytes captured (5608 bits)
Ethernet II, Src: Oracle_81:3f:27 (00:14:4f:81:3f:27), Dst: MS-NLB-PhysServer-12_db:57:57:04 (02:0c:db:57:57:04)
Internet Protocol Version 4, Src: 68.87.87.159 (68.87.87.159), Dst: 68.87.87.4 (68.87.87.4)
    Version: 4
    Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 687
    Identification: 0xd836 (55350)
  Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
  Header checksum: 0x0000 [validation disabled]
```

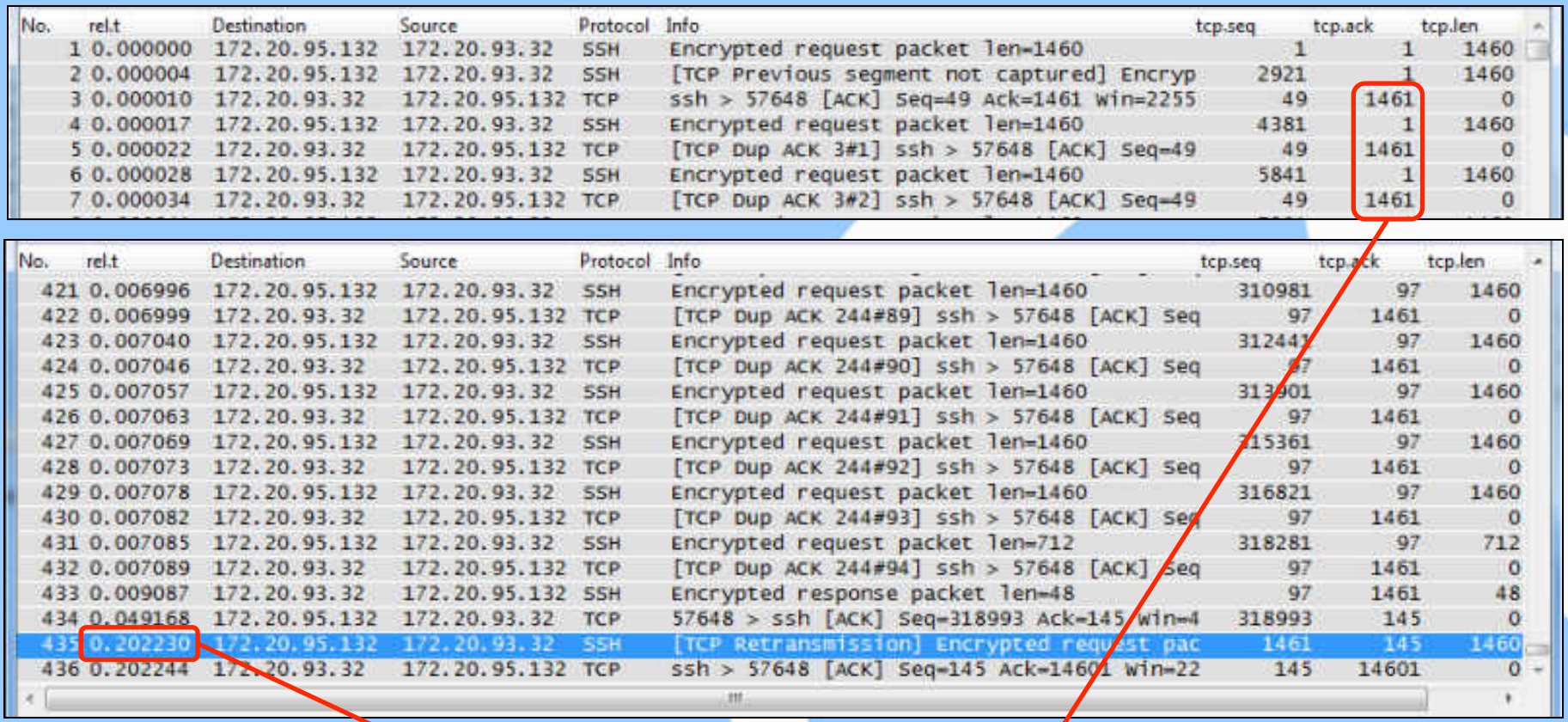
Application Gives Up!

Technical Indicators:

Explicit Application Feedback Timing (Delay)

Why is TCP waiting 3 seconds to retransmit the first lost segment?

Implicit Feedback Mechanisms



No.	rel.t	Destination	Source	Protocol	Info	tcp.seq	tcp.ack	tcp.len
1	0.000000	172.20.95.132	172.20.93.32	SSH	Encrypted request packet len=1460	1	1	1460
2	0.000004	172.20.95.132	172.20.93.32	SSH	[TCP Previous segment not captured] Encryp	2921	1	1460
3	0.000010	172.20.93.32	172.20.95.132	TCP	ssh > 57648 [ACK] Seq=49 Ack=1461 Win=2255	49	1461	0
4	0.000017	172.20.95.132	172.20.93.32	SSH	Encrypted request packet len=1460	4381	1	1460
5	0.000022	172.20.93.32	172.20.95.132	TCP	[TCP Dup ACK 3#1] ssh > 57648 [ACK] Seq=49	49	1461	0
6	0.000028	172.20.95.132	172.20.93.32	SSH	Encrypted request packet len=1460	5841	1	1460
7	0.000034	172.20.93.32	172.20.95.132	TCP	[TCP Dup ACK 3#2] ssh > 57648 [ACK] Seq=49	49	1461	0

No.	rel.t	Destination	Source	Protocol	Info	tcp.seq	tcp.ack	tcp.len
421	0.006996	172.20.95.132	172.20.93.32	SSH	Encrypted request packet len=1460	310981	97	1460
422	0.006999	172.20.93.32	172.20.95.132	TCP	[TCP Dup ACK 244#89] ssh > 57648 [ACK] Seq	97	1461	0
423	0.007040	172.20.95.132	172.20.93.32	SSH	Encrypted request packet len=1460	312441	97	1460
424	0.007046	172.20.93.32	172.20.95.132	TCP	[TCP Dup ACK 244#90] ssh > 57648 [ACK] Seq	97	1461	0
425	0.007057	172.20.95.132	172.20.93.32	SSH	Encrypted request packet len=1460	313901	97	1460
426	0.007063	172.20.93.32	172.20.95.132	TCP	[TCP Dup ACK 244#91] ssh > 57648 [ACK] Seq	97	1461	0
427	0.007069	172.20.95.132	172.20.93.32	SSH	Encrypted request packet len=1460	315361	97	1460
428	0.007073	172.20.93.32	172.20.95.132	TCP	[TCP Dup ACK 244#92] ssh > 57648 [ACK] Seq	97	1461	0
429	0.007078	172.20.95.132	172.20.93.32	SSH	Encrypted request packet len=1460	316821	97	1460
430	0.007082	172.20.93.32	172.20.95.132	TCP	[TCP Dup ACK 244#93] ssh > 57648 [ACK] Seq	97	1461	0
431	0.007085	172.20.95.132	172.20.93.32	SSH	Encrypted request packet len=712	318281	97	712
432	0.007089	172.20.93.32	172.20.95.132	TCP	[TCP Dup ACK 244#94] ssh > 57648 [ACK] Seq	97	1461	0
433	0.009087	172.20.93.32	172.20.95.132	SSH	Encrypted response packet len=48	97	1461	48
434	0.049168	172.20.95.132	172.20.93.32	TCP	57648 > ssh [ACK] Seq=318993 Ack=145 Win=4	318993	145	0
435	0.202230	172.20.95.132	172.20.93.32	SSH	[TCP Retransmission] Encrypted request pac	1461	145	1460
436	0.202244	172.20.93.32	172.20.95.132	TCP	ssh > 57648 [ACK] Seq=145 Ack=14601 Win=22	145	14601	0

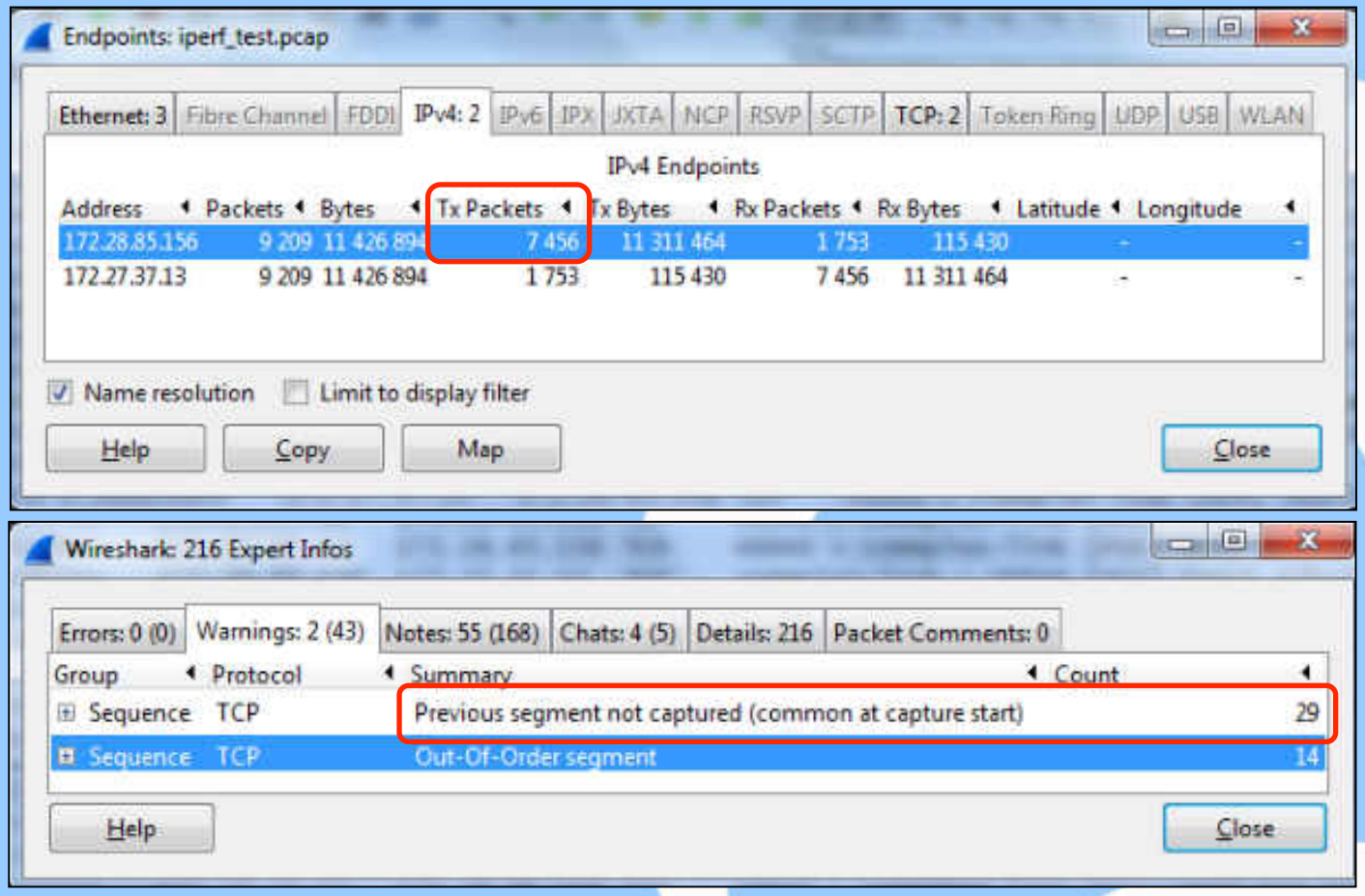
Technical Indicators:

TCP Retransmission
Timing (Delay)
Behavior (not Fast Retransmitting)

Why did .32 not Fast Retransmit after receiving 3 duplicate ACKs?

Why did .32 wait 200ms before retransmitting the lost segment?

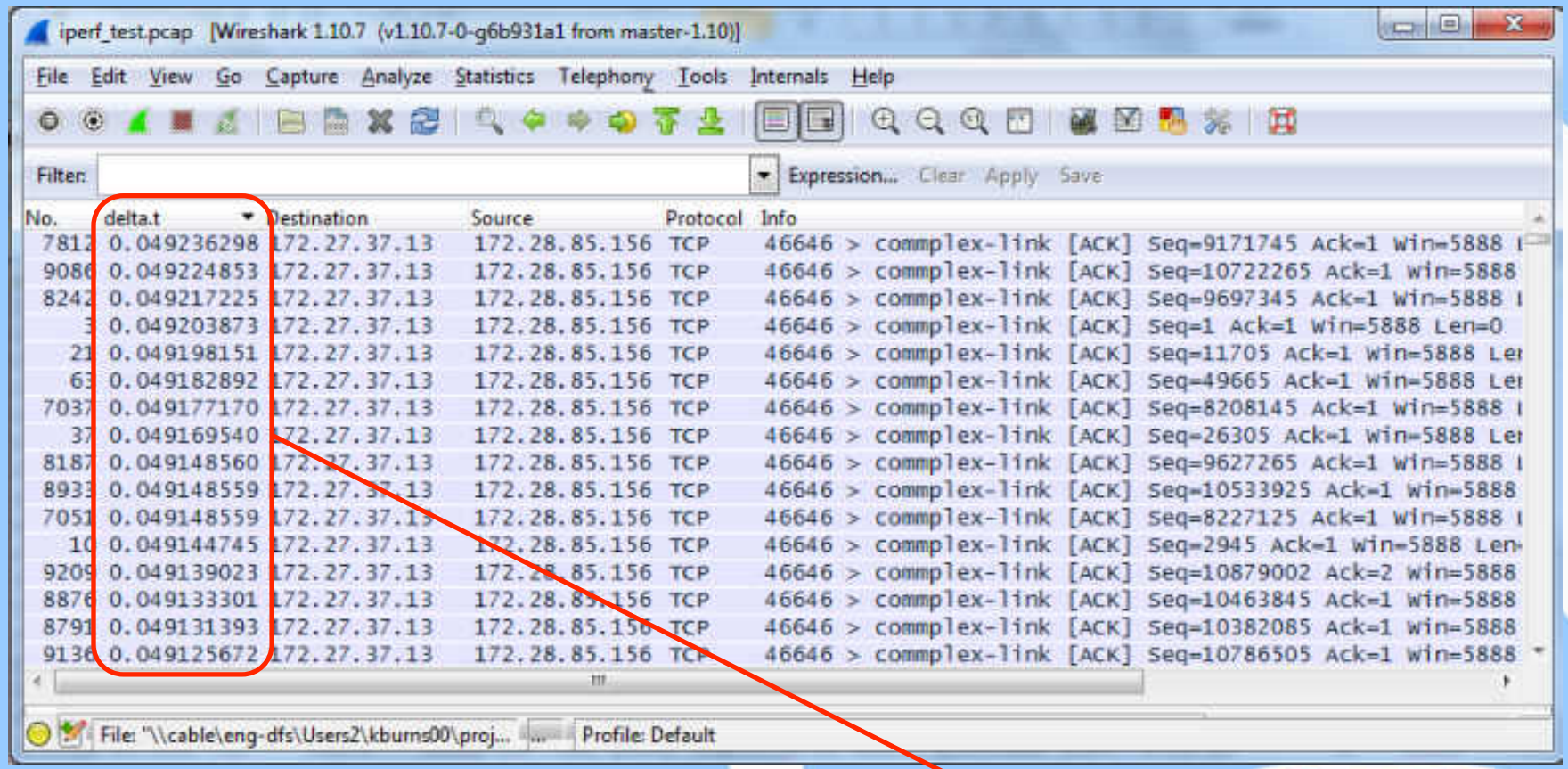
Extrapolated Data & Measurements



Technical Indicators:
Lost Packets and TCP Retransmissions

Packet Loss = 0.003 (.3%)

Finding Round Trip Latency



Round Trip Latency = 49ms

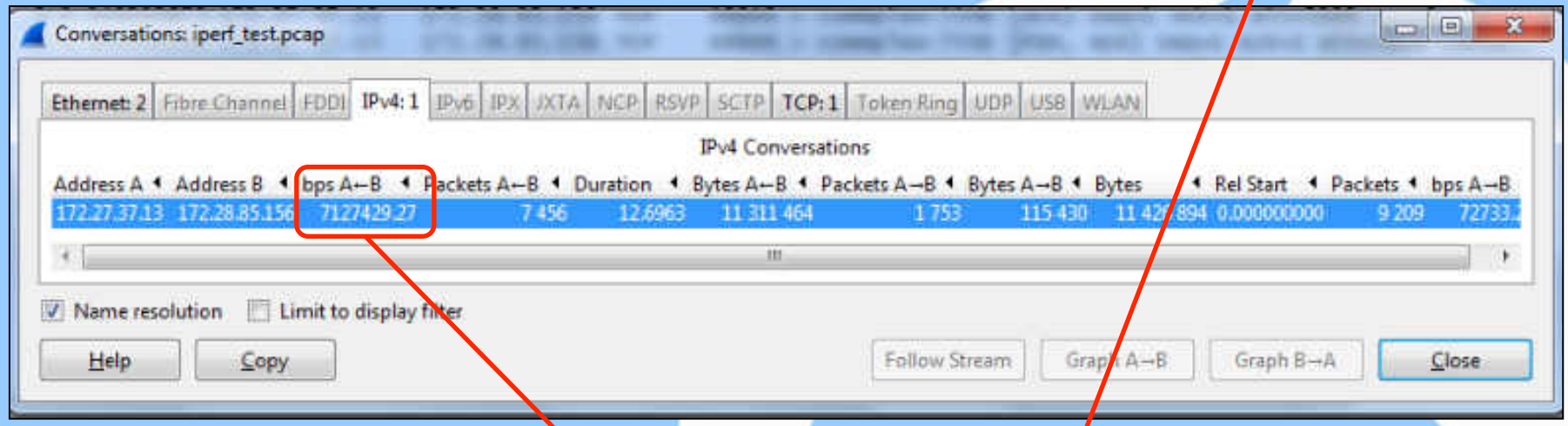
Technical Indicators:

Timing (Round Trip Time)

Sorting by Delta Time lets us see the round trip latency !!

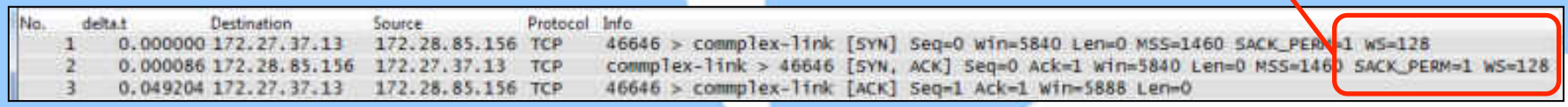
Throughput Measurement

http://www.switch.ch/network/tools/tcp_throughput/
Maximum throughput with a TCP window of 64 KByte and RTT of 49.0 ms <= **10.45 Mbit/sec.**



Throughput is at 70% of theoretical max using 64K Buffers

Window scaling is enabled. Shouldn't it have more TCP tx buffers to use?



Useful Technical Indicators

- Timing Based
  - Delta Time
    - Latency / Delay measurements
  - Relative Time
    - Throughput and Response Times
  - Absolute Time
    - Correlation to log files
- TCP Based
  - SYN, FIN, Reset
  - Retransmissions & Out of Order Packets
  - ACKs: Dup, Triple, Delayed, SACK
  - Windowing: Window Size & Window Full Messages
- Application Based
  - Transaction ID's
  - Control Messages
    - Open, Close, Abort
- Measurements
  - Service Response Time
  - Latency & Throughput
  - Other Delay

Techniques

- Standard Columns
  - Delta Time: Sorting to find latency
  - Relative Time: Find request/response delays
- Custom Columns
  - IP: ip.ttl, ip.id
  - TCP: tcp.seq, tcp.ack, tcp.len, tcp.options.sack
  - Application Specific (transaction/message IDs)
- Service Response Times
  - Use to find application delays
- Expert
  - Best used to look for TCP behavior (reactions to conditions on the wire)

Techniques

- IP Based
  - Use TTL column to visualize packet flow through routers
  - Use IPID column to visualize packet loss.
- TCP Based
  - Out of Order Packets: Look for SACKs in opposite direction. Indicates possible packet loss or network queuing or async routing issues.
  - ACK: Useful to prove a request arrived at a destination
  - Dup ACKs: Triple Dup ACKs indicate host not using Fast Retransmit algorithm.
  - Delayed ACKs: Indicates TCP waiting for an application.
  - Windowing: Full windows may indicate application problems or lack of TCP buffering (scaling needed).

Techniques

- Application Based
  - Always attempt to decode the application layer.
  - Look for hints in the packet hex bytes that may indicate what the protocol is.
  - Look for explicit messages that indicate application behavior or reactions to conditions on the wire.
  - Find protocol fields that allow you to track requests and responses.
  - Associate application messages and behavior to reactions and recovery mechanisms in the transport layer (ie: TCP).

Case Study: Slow Database Transactions

Application

Performance degredation with database transactions.

We were told LDAP was used as the database exchange method.

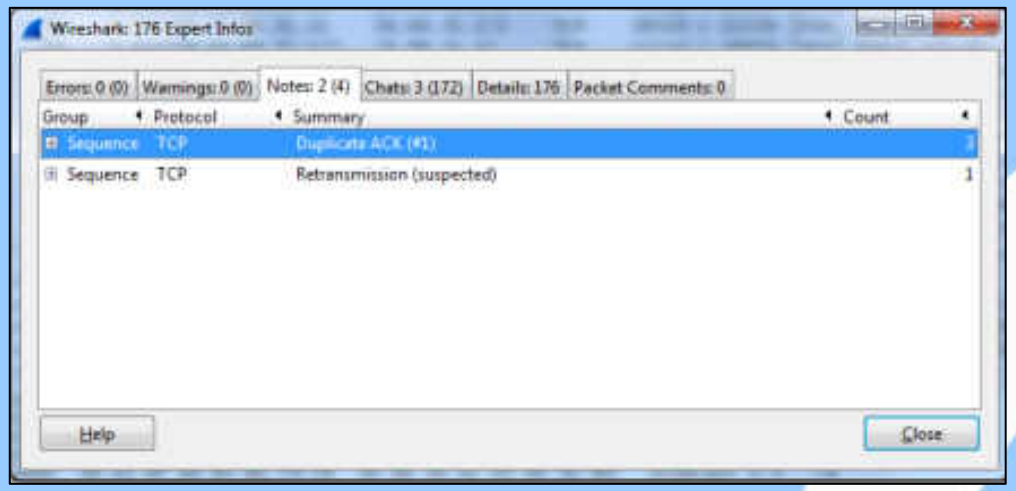
Symptoms

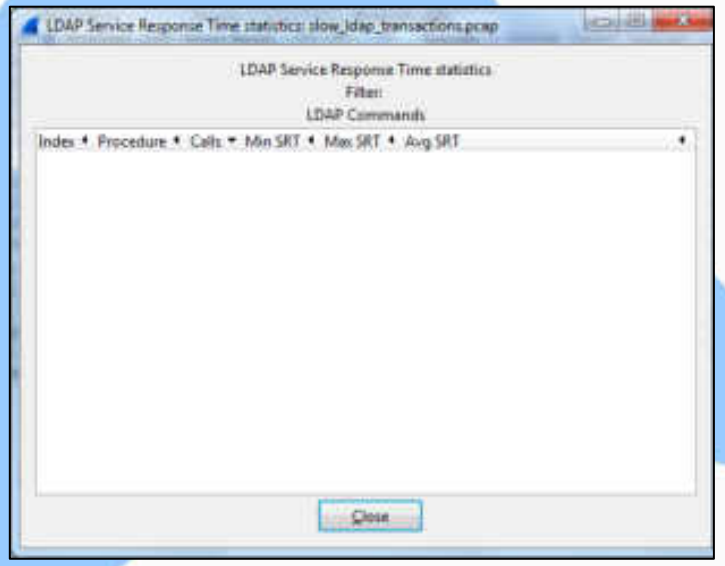
- Transactions which should take less than one second are taking up to (5) seconds causing the application to disconnect.
- Network path appears to be clean. No obvious loss or latency.

Manifestation

- Problem manifests as delay
- Location of Delay uncertain.

Case Study: Slow Database Transactions

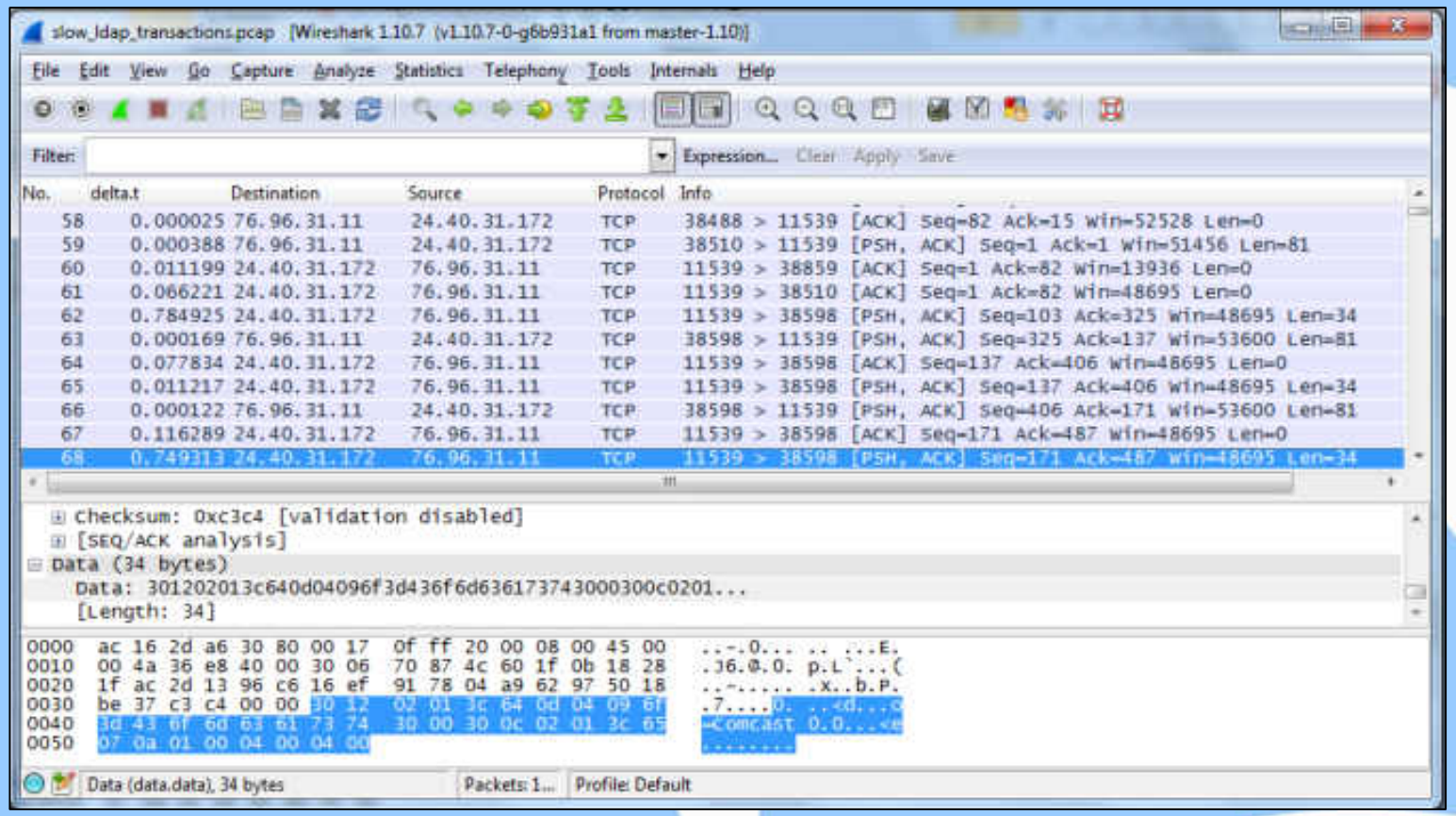




Technical Indicators:

No obvious or relevant indicators found.

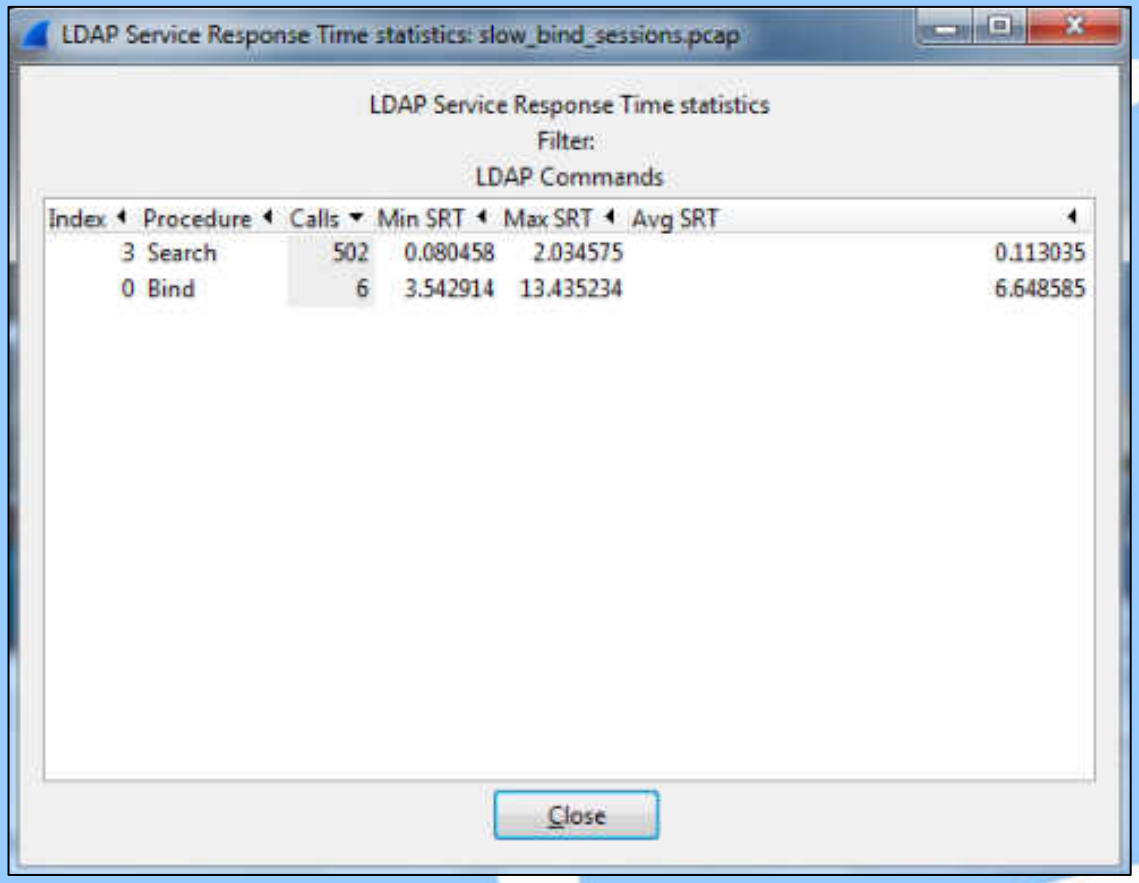
Case Study: Slow Database Transactions



Technique

Look in Hex Data for a hint on what the protocol may be.

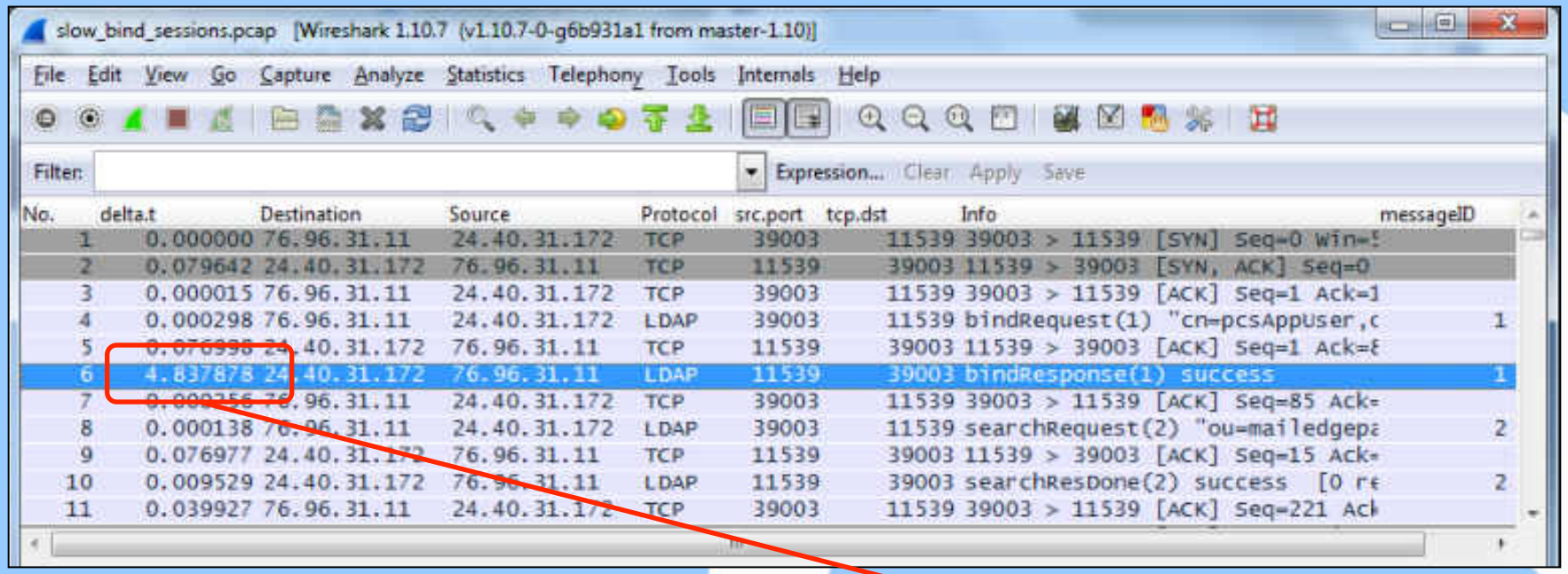
Case Study: Slow Database Transactions



Technical Indicators

LDAP Bind Time is very slow.

Case Study: Slow Database Transactions



Why does it take the LDAP server nearly 5 seconds to respond to the Bind request??

Case Study: Sudo Command Slow

Application

UNIX servers and VMs.

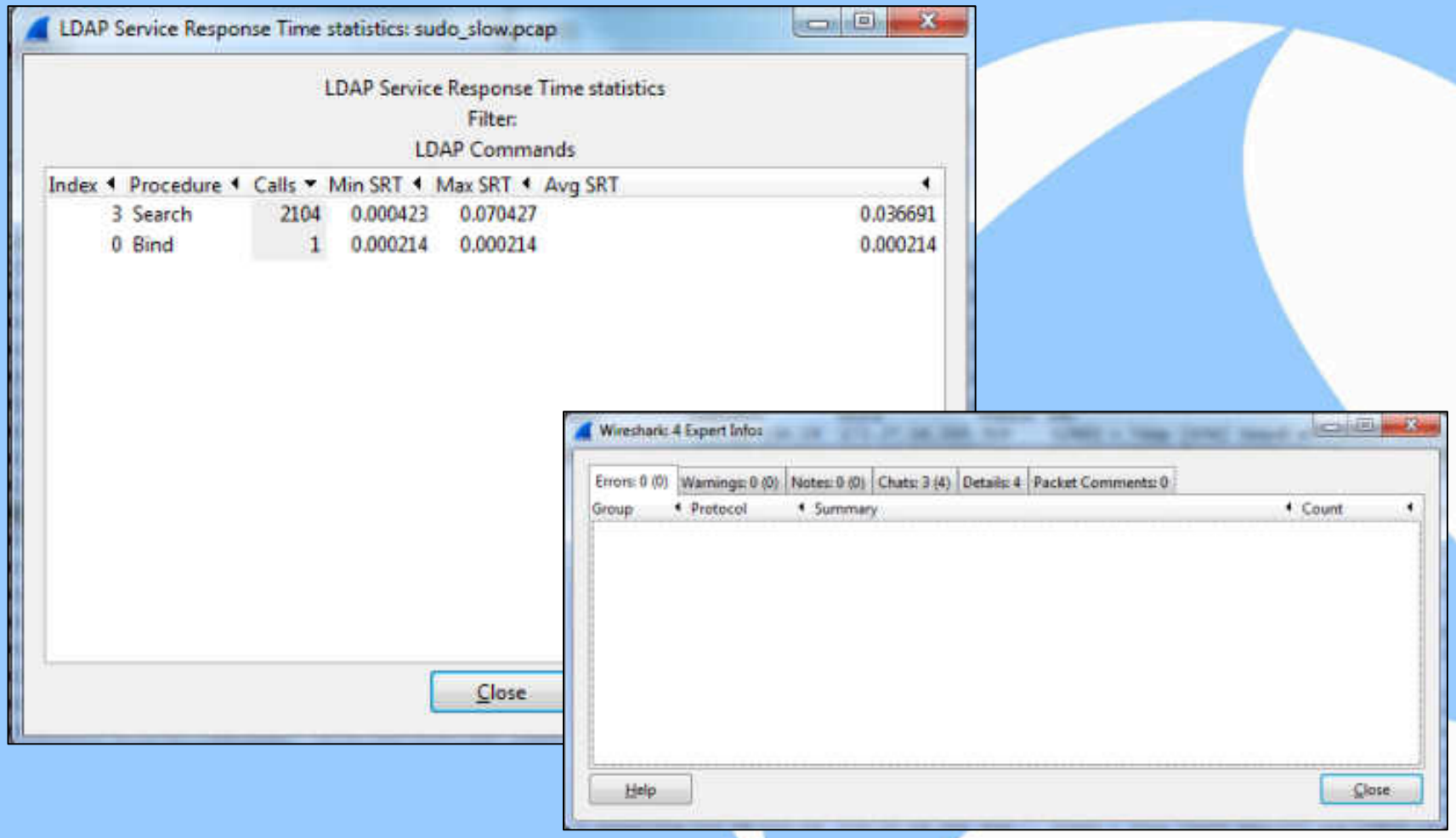
Symptoms

- UNIX admins are reporting very slow response times running SUDO level commands.
- Network path appears to be clean. No obvious loss or latency.

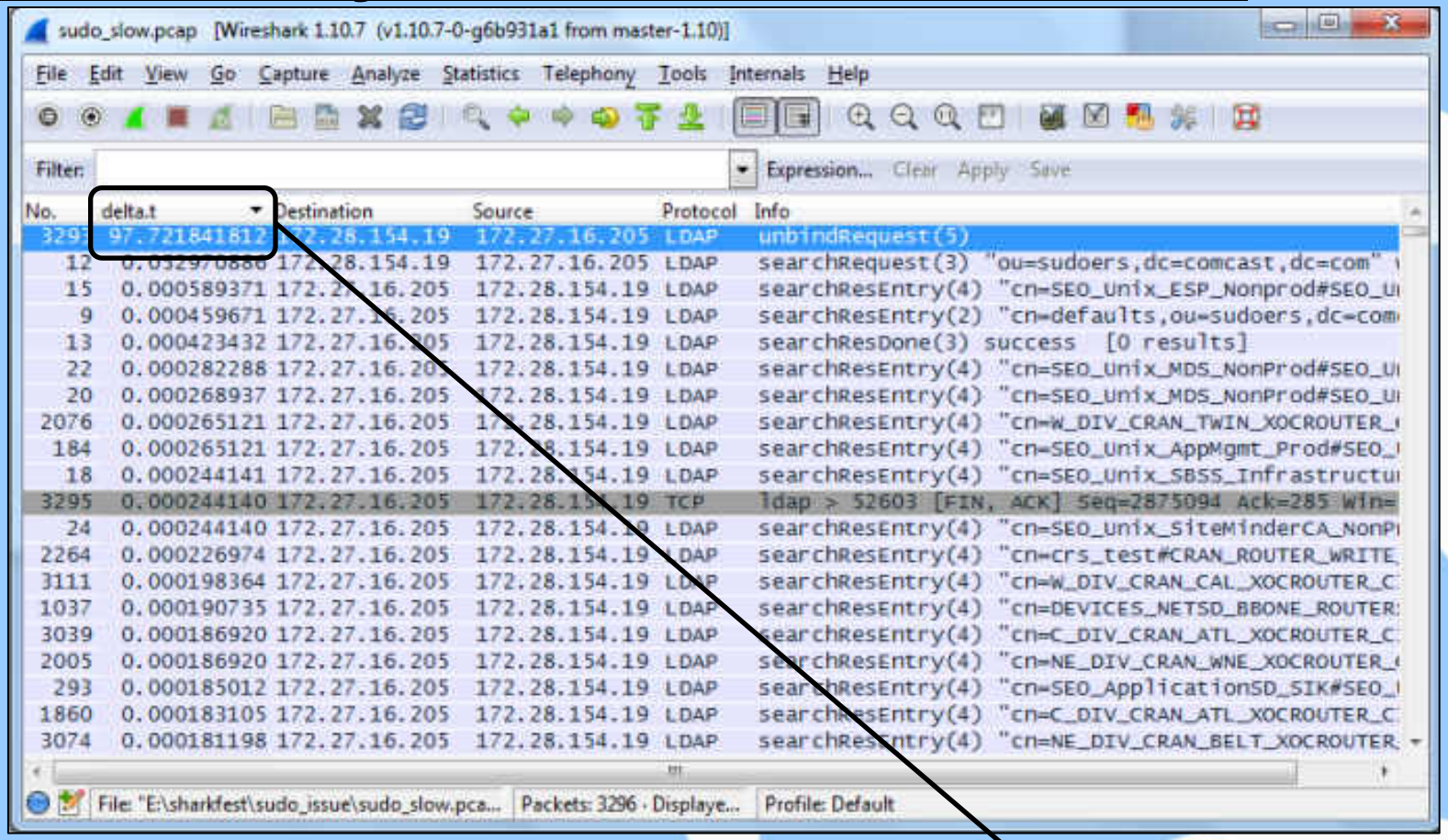
Manifestation

- Problem manifests as delay
- Location of Delay uncertain.

Case Study: Sudo Command Slow



Case Study: Sudo Command Slow

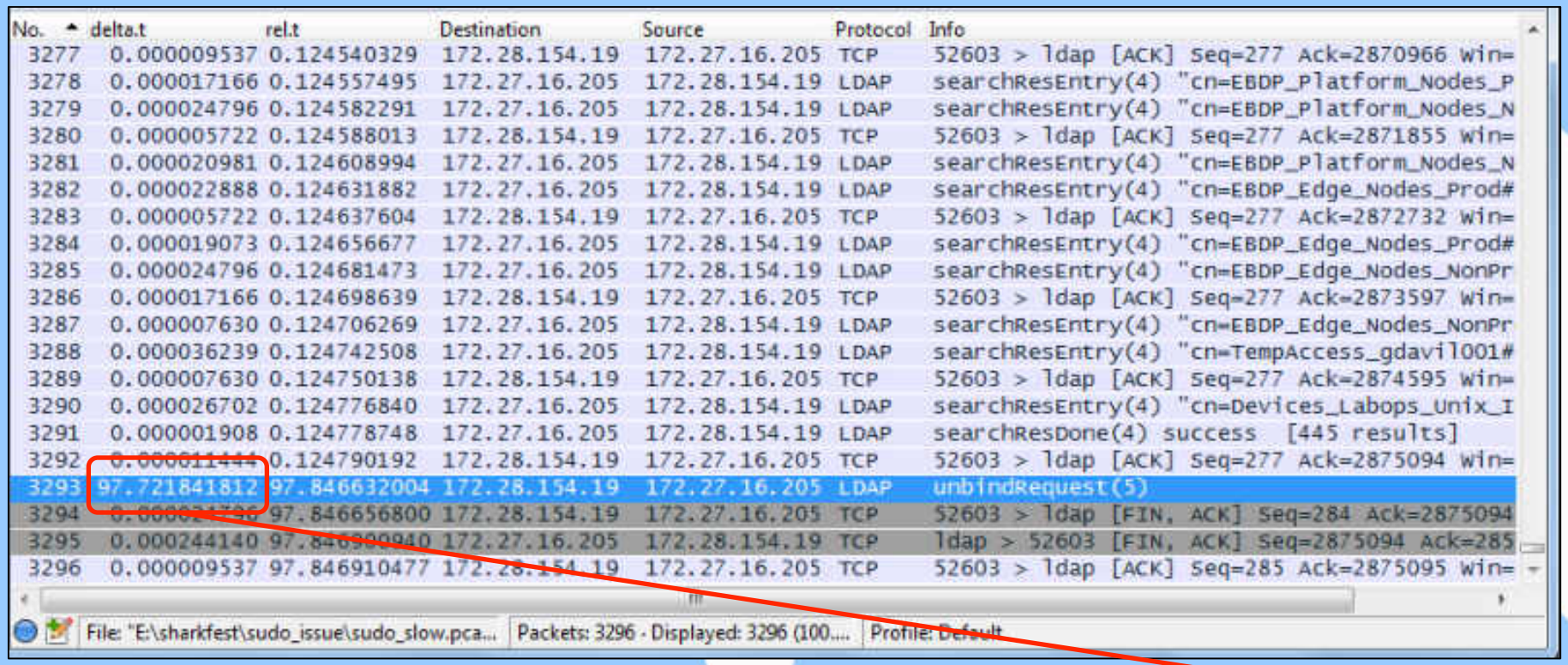


Technical Indicators

Large delay seen in delta time

Sorting by Delta Time manifests obvious delays!

Case Study: Sudo Command Slow



No.	delta.t	rel.t	Destination	Source	Protocol	Info
3277	0.000009537	0.124540329	172.28.154.19	172.27.16.205	TCP	52603 > ldap [ACK] Seq=277 Ack=2870966 Win=
3278	0.000017166	0.124557495	172.27.16.205	172.28.154.19	LDAP	searchResEntry(4) "cn=EBDP_Platform_Nodes_P
3279	0.000024796	0.124582291	172.27.16.205	172.28.154.19	LDAP	searchResEntry(4) "cn=EBDP_Platform_Nodes_N
3280	0.000005722	0.124588013	172.28.154.19	172.27.16.205	TCP	52603 > ldap [ACK] Seq=277 Ack=2871855 Win=
3281	0.000020981	0.124608994	172.27.16.205	172.28.154.19	LDAP	searchResEntry(4) "cn=EBDP_Platform_Nodes_N
3282	0.000022888	0.124631882	172.27.16.205	172.28.154.19	LDAP	searchResEntry(4) "cn=EBDP_Edge_Nodes_Prod#
3283	0.000005722	0.124637604	172.28.154.19	172.27.16.205	TCP	52603 > ldap [ACK] Seq=277 Ack=2872732 Win=
3284	0.000019073	0.124656677	172.27.16.205	172.28.154.19	LDAP	searchResEntry(4) "cn=EBDP_Edge_Nodes_Prod#
3285	0.000024796	0.124681473	172.27.16.205	172.28.154.19	LDAP	searchResEntry(4) "cn=EBDP_Edge_Nodes_NonPr
3286	0.000017166	0.124698639	172.28.154.19	172.27.16.205	TCP	52603 > ldap [ACK] Seq=277 Ack=2873597 Win=
3287	0.000007630	0.124706269	172.27.16.205	172.28.154.19	LDAP	searchResEntry(4) "cn=EBDP_Edge_Nodes_NonPr
3288	0.000036239	0.124742508	172.27.16.205	172.28.154.19	LDAP	searchResEntry(4) "cn=TempAccess_gdavil001#
3289	0.000007630	0.124750138	172.28.154.19	172.27.16.205	TCP	52603 > ldap [ACK] Seq=277 Ack=2874595 Win=
3290	0.000026702	0.124776840	172.27.16.205	172.28.154.19	LDAP	searchResEntry(4) "cn=Devices_Labops_Unix_I
3291	0.000001908	0.124778748	172.27.16.205	172.28.154.19	LDAP	searchResDone(4) success [445 results]
3292	0.000011444	0.124790192	172.28.154.19	172.27.16.205	TCP	52603 > ldap [ACK] Seq=277 Ack=2875094 Win=
3293	97.721841812	97.846632004	172.28.154.19	172.27.16.205	LDAP	unbindRequest(5)
3294	0.000021708	97.846656800	172.28.154.19	172.27.16.205	TCP	52603 > ldap [FIN, ACK] Seq=284 Ack=2875094
3295	0.000244140	97.846900940	172.27.16.205	172.28.154.19	TCP	ldap > 52603 [FIN, ACK] Seq=2875094 Ack=285
3296	0.000009537	97.846910477	172.28.154.19	172.27.16.205	TCP	52603 > ldap [ACK] Seq=285 Ack=2875095 Win=

File: "E:\sharkfest\sudo_issue\sudo_slow.pca... | Packets: 3296 · Displayed: 3296 (100.... | Profile: Default

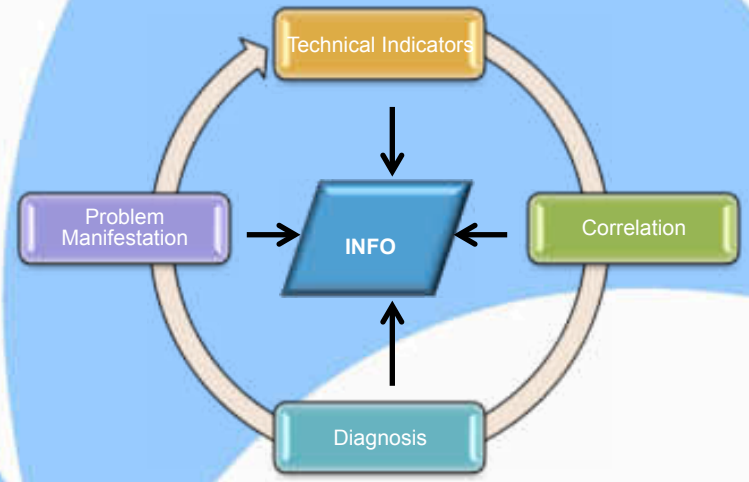
Technical Indicators

LDAP Unbind Time is very slow.

Client waits 97 seconds before unbinding the LDAP connection.

What is Correlation?

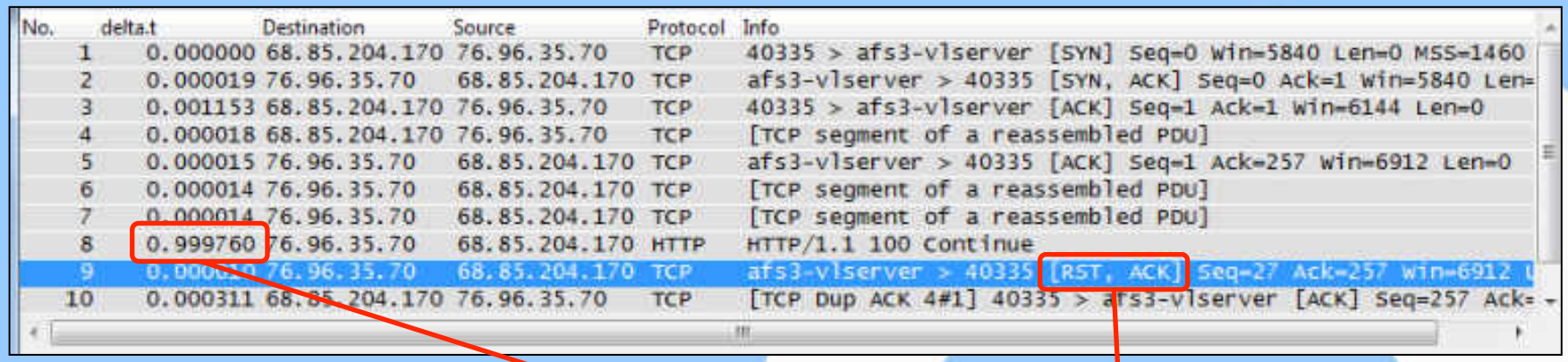
- The goal of correlation is to map the problem's method of manifestation to what is happening in the packets !!
- The process of correlating technical indicators must be understood, you cannot automate anything you have never done manually.
- You need to understand the protocols and the tools, know how Wireshark thinks !!!



Correlation Best Practices

- The correlation process starts by understanding how a problem manifests itself.
- Get as much information from the users and technical staff as possible.
- Ask how it is known the problem actually exists.
- Always analyze from the client's perspective first.
- Look for small patterns that can represent the problem as a whole.
  - A complex problem can often be represented by 10 packets or less.
- Visualize and understand requests and responses. Be the app!!
  - *You cannot automate this part unless you understand how to do it manually.*
- Understand the relationship between different technical indicators.
- Use visualization techniques for large amounts of packets.
  - Graphs, expert, column sorting.

Packet Based Correlations



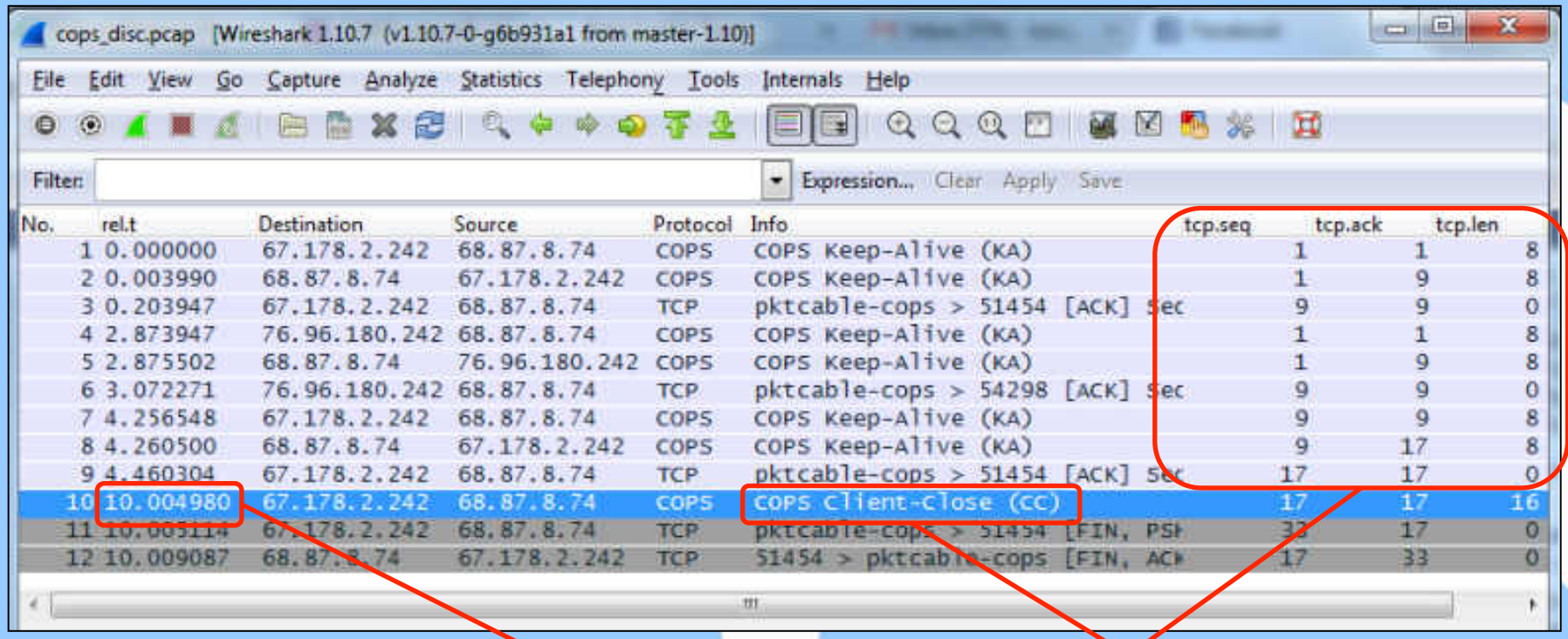
No.	delta.t	Destination	Source	Protocol	Info
1	0.000000	68.85.204.170	76.96.35.70	TCP	40335 > afs3-vlserver [SYN] Seq=0 Win=5840 Len=0 MSS=1460
2	0.000019	76.96.35.70	68.85.204.170	TCP	afs3-vlserver > 40335 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=
3	0.001153	68.85.204.170	76.96.35.70	TCP	40335 > afs3-vlserver [ACK] Seq=1 Ack=1 Win=6144 Len=0
4	0.000018	68.85.204.170	76.96.35.70	TCP	[TCP segment of a reassembled PDU]
5	0.000015	76.96.35.70	68.85.204.170	TCP	afs3-vlserver > 40335 [ACK] Seq=1 Ack=257 Win=6912 Len=0
6	0.000014	76.96.35.70	68.85.204.170	TCP	[TCP segment of a reassembled PDU]
7	0.000014	76.96.35.70	68.85.204.170	TCP	[TCP segment of a reassembled PDU]
8	0.999760	76.96.35.70	68.85.204.170	HTTP	HTTP/1.1 100 Continue
9	0.000019	76.96.35.70	68.85.204.170	TCP	afs3-vlserver > 40335 [RST, ACK] Seq=27 Ack=257 Win=6912 L
10	0.000311	68.85.204.170	76.96.35.70	TCP	[TCP Dup ACK 4#1] 40335 > afs3-vlserver [ACK] Seq=257 Ack=

Technical Indicators

TCP Reset
Delay (delta time)

TCP Reset correlates to a 1 second time out !

Behavior Based Correlations

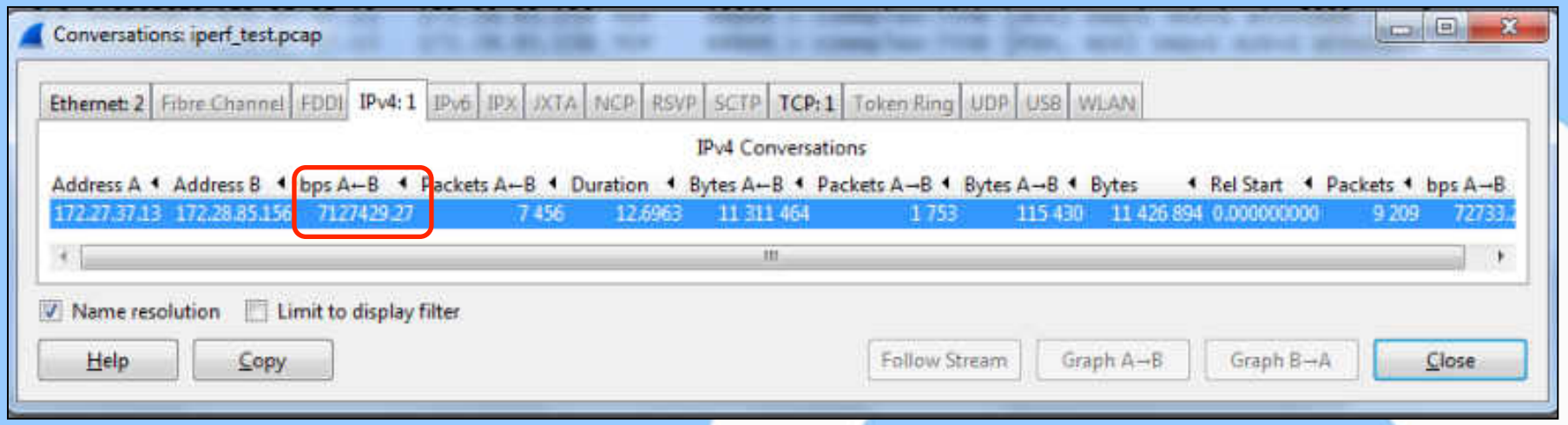


Technical Indicators

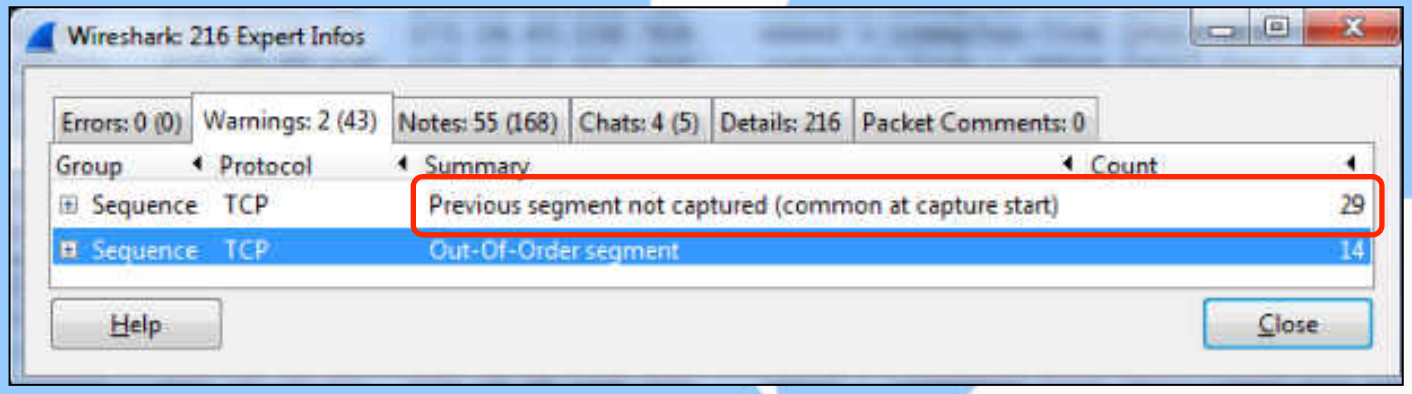
TCP SEQ-ACK-LEN
Relative Time

TCP columns allow us to prove all keepalives were received yet the application still times out after 10 seconds and closes the connection.

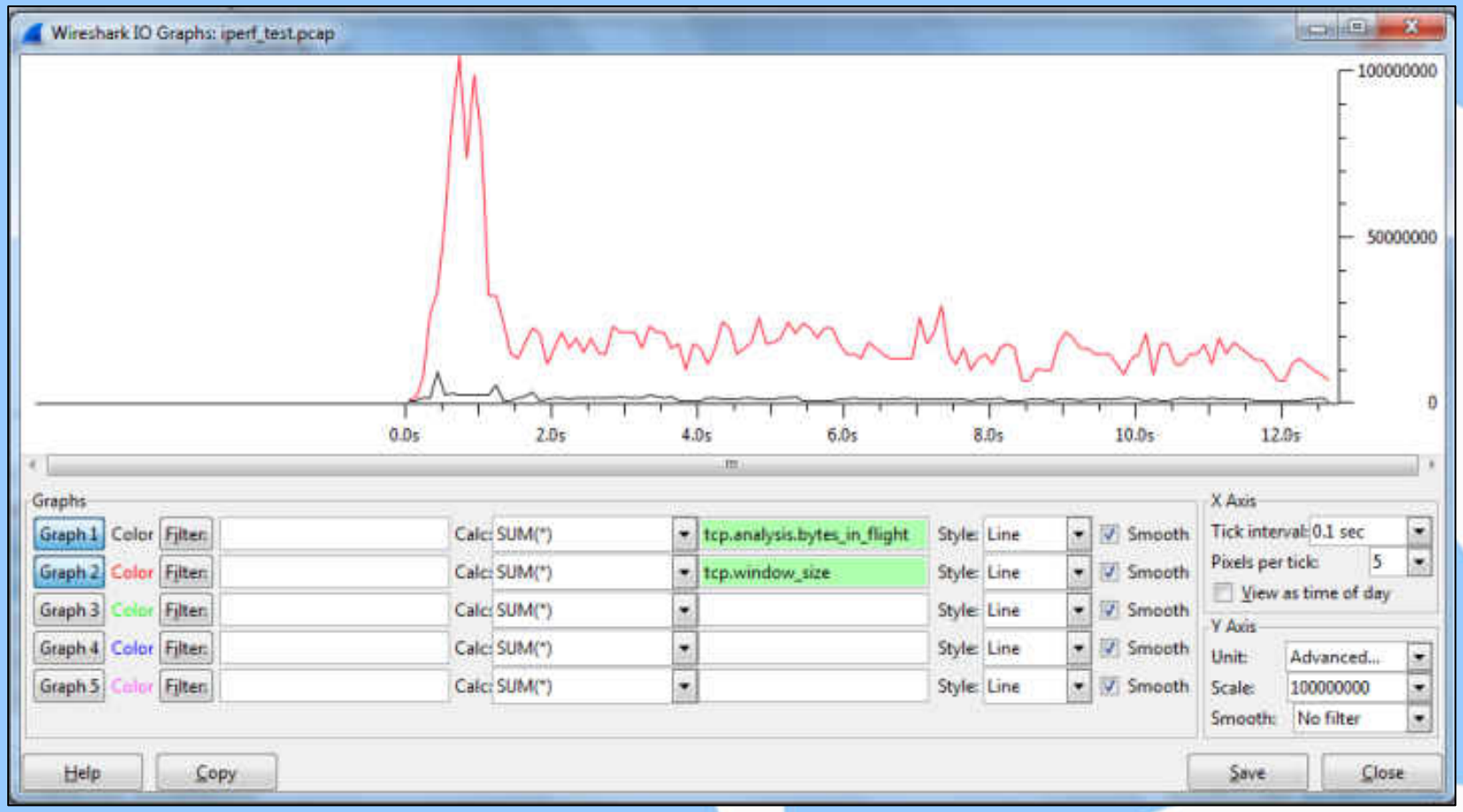
Data Extrapolation Revisited



7Mb/sec with .3% packet loss

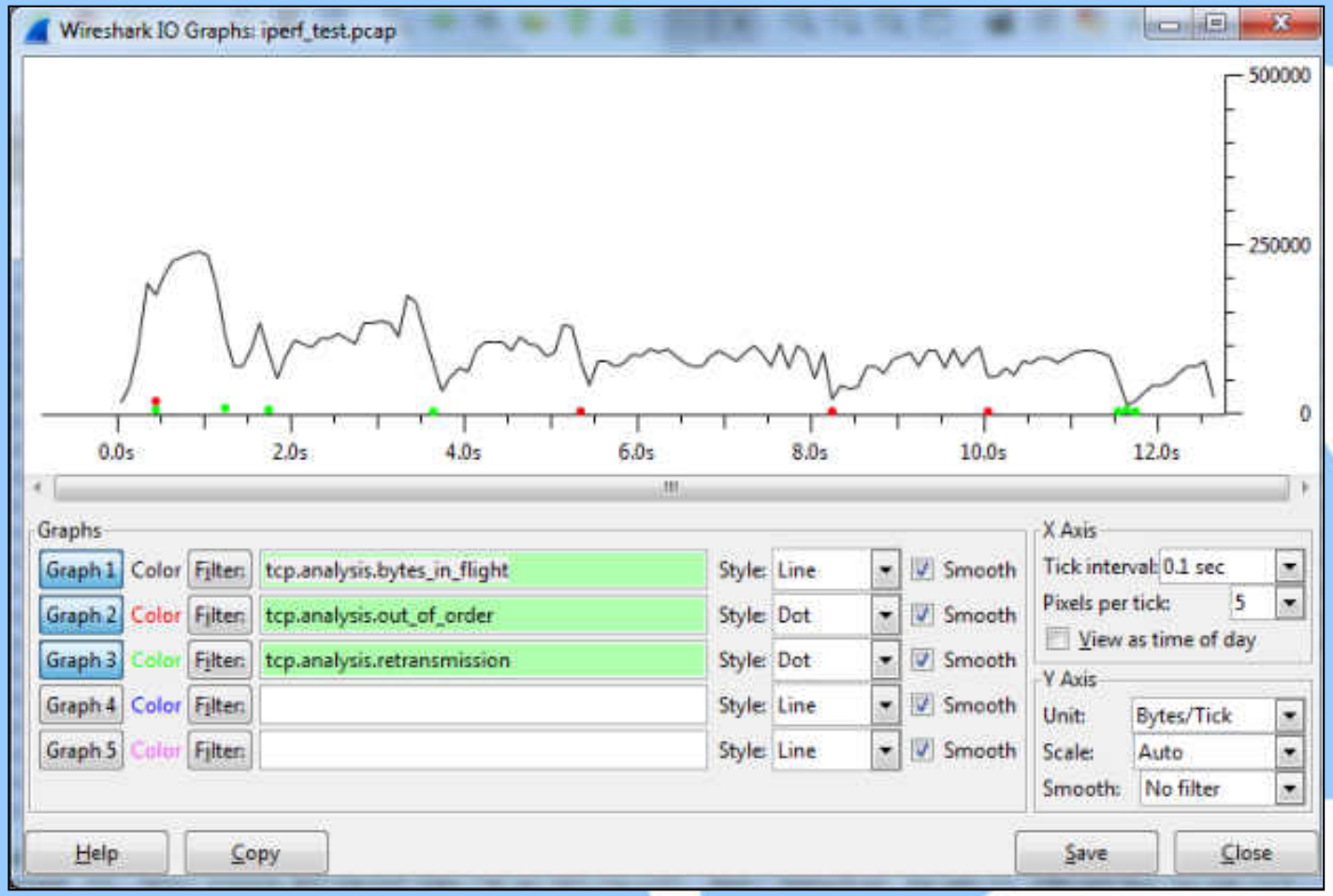


Measurement Based Correlations



Correlation of Bytes in Flight and Receiver Window Size indicates inefficient use of available receiver buffers…… but why?

Measurement Based Correlations



Can you spot the correlation that visualizes the problem?

Correlation Techniques

- Know where the analyzer is
  - Use TTL value to determine the location of packet collection
- Identify Client and Server
  - Always analyze from the perspective of the client first
- Identify Requests and Responses
  - Important to be able to measure transaction times and understand application behavior.
- Associate Packets to Process
  - Look for manifestation behavior in the packets
  - Utilize hex data to learn more about the application
- Look for obvious timing indicators that can be correlated with behavior. Common timers are: 1, 2, 5, 10, 30,60,120… (seconds)
- Reduce the scope of the problem to as few packets as possible.
  - Concentrate on single sessions.

Case Study: Radius Authentication

No.	del.t	rel.t	Destination	Source	Protocol	Info
1	0.000000	0.000000	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=463)
2	0.000008	0.000008	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=463)
3	0.000013	0.000021	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=463)
4	0.000759	0.000780	69.252.208.133	172.30.16.147	RADIUS	Accounting-Response(5) (id=118, l=20)
5	0.000029	0.000809	69.252.208.133	172.30.16.147	RADIUS	Accounting-Response(5) (id=118, l=20)
6	1.454033	1.454842	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=510)
7	0.000013	1.454855	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=510)
8	0.000012	1.454867	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=510)
9	0.000698	1.455565	69.252.208.133	172.30.16.147	RADIUS	Accounting-Response(5) (id=118, l=20)
10	0.673942	2.129507	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=491)
11	0.000012	2.129519	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=491)
12	0.000015	2.129534	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=491)
13	0.011410	2.140944	69.252.208.133	172.30.16.147	RADIUS	Accounting-Response(5) (id=118, l=20)
14	0.000028	2.140972	69.252.208.133	172.30.16.147	RADIUS	Accounting-Response(5) (id=118, l=20)
15	1.354180	3.495152	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=510)
16	0.000017	3.495169	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=510)
17	0.000004	3.495173	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=510)
18	0.001219	3.496392	69.252.208.133	172.30.16.147	RADIUS	Accounting-Response(5) (id=118, l=20)
19	1.693790	5.190182	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=534)
20	0.000013	5.190195	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=534)
21	0.000004	5.190199	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=534)
22	0.000813	5.191012	69.252.208.133	172.30.16.147	RADIUS	Accounting-Response(5) (id=118, l=20)
23	0.892666	6.083678	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=506)
24	0.000015	6.083693	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=506)
25	0.000017	6.083710	172.30.16.147	69.252.208.133	RADIUS	Accounting-Request(4) (id=118, l=506)
26	0.000820	6.084530	69.252.208.133	172.30.16.147	RADIUS	Accounting-Response(5) (id=118, l=20)
27	0.000052	6.084582	69.252.208.133	172.30.16.147	RADIUS	Accounting-Response(5) (id=118, l=20)

How do we find and visualize packet loss?

Case Study: Visualize Sessions

No.	del.t	rel.t	Destination	Source	dst.port	src.port	Protocol	Info
1	0.000000	0.000000	172.30.16.147	69.252.208.133	1813	21503	RADIUS	Accounting-Request(4) (id=118,
2	0.000008	0.000008	172.30.16.147	69.252.208.133	1813	21503	RADIUS	Accounting-Request(4) (id=118,
3	0.000013	0.000021	172.30.16.147	69.252.208.133	1813	21503	RADIUS	Accounting-Request(4) (id=118,
4	0.000759	0.000780	69.252.208.133	172.30.16.147	21503	1813	RADIUS	Accounting-Response(5) (id=118
5	0.000029	0.000809	69.252.208.133	172.30.16.147	21503	1813	RADIUS	Accounting-Response(5) (id=118
6	1.454033	1.454842	172.30.16.147	69.252.208.133	1813	21502	RADIUS	Accounting-Request(4) (id=118,
7	0.000013	1.454855	172.30.16.147	69.252.208.133	1813	21502	RADIUS	Accounting-Request(4) (id=118,
8	0.000012	1.454867	172.30.16.147	69.252.208.133	1813	21502	RADIUS	Accounting-Request(4) (id=118,
9	0.000698	1.455565	69.252.208.133	172.30.16.147	21502	1813	RADIUS	Accounting-Response(5) (id=118
10	0.673942	2.129507	172.30.16.147	69.252.208.133	1813	21504	RADIUS	Accounting-Request(4) (id=118,
11	0.000012	2.129519	172.30.16.147	69.252.208.133	1813	21504	RADIUS	Accounting-Request(4) (id=118,
12	0.000015	2.129534	172.30.16.147	69.252.208.133	1813	21504	RADIUS	Accounting-Request(4) (id=118,
13	0.011410	2.140944	69.252.208.133	172.30.16.147	21504	1813	RADIUS	Accounting-Response(5) (id=118
14	0.000028	2.140972	69.252.208.133	172.30.16.147	21504	1813	RADIUS	Accounting-Response(5) (id=118
15	1.354180	3.495152	172.30.16.147	69.252.208.133	1813	21502	RADIUS	Accounting-Request(4) (id=118,
16	0.000017	3.495169	172.30.16.147	69.252.208.133	1813	21502	RADIUS	Accounting-Request(4) (id=118,
17	0.000004	3.495173	172.30.16.147	69.252.208.133	1813	21502	RADIUS	Accounting-Request(4) (id=118,
18	0.001219	3.496392	69.252.208.133	172.30.16.147	21502	1813	RADIUS	Accounting-Response(5) (id=118
19	1.693790	5.190182	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
20	0.000013	5.190195	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
21	0.000004	5.190199	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
22	0.000813	5.191012	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118
23	0.892666	6.083678	172.30.16.147	69.252.208.133	1813	21503	RADIUS	Accounting-Request(4) (id=118,
24	0.000015	6.083693	172.30.16.147	69.252.208.133	1813	21503	RADIUS	Accounting-Request(4) (id=118,
25	0.000017	6.083710	172.30.16.147	69.252.208.133	1813	21503	RADIUS	Accounting-Request(4) (id=118,
26	0.000820	6.084530	69.252.208.133	172.30.16.147	21503	1813	RADIUS	Accounting-Response(5) (id=118
27	0.000052	6.084582	69.252.208.133	172.30.16.147	21503	1813	RADIUS	Accounting-Response(5) (id=118

Technical Indicator

Number of packets in each session.

Technique

Use Columns to Visualize Sessions

Case Study: Filter to Single Session

No.	del.t	rel.t	Destination	Source	dst.port	src.port	Protocol	Info
1	0.000000	0.000000	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
2	0.000013	0.000013	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
3	0.000817	0.000830	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118
4	2.015653	2.016483	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
5	0.000012	2.016495	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
6	0.000637	2.017132	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118
7	0.000028	2.017160	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118
8	4.499100	6.516260	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
9	0.000027	6.516287	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
10	0.000715	6.517002	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118
11	2.027109	8.544111	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
12	0.000014	8.544125	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
13	0.001001	8.545126	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118
14	6.732012	15.277138	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
15	0.000007	15.277145	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
16	0.000786	15.277931	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118
17	2.035259	17.313190	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
18	0.000002	17.313192	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
19	0.001446	17.314638	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118
20	6.735149	24.049787	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
21	0.000016	24.049803	172.30.16.147	69.252.208.133	1813	21501	RADIUS	Accounting-Request(4) (id=118,
22	0.000738	24.050541	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118
23	0.000032	24.050573	69.252.208.133	172.30.16.147	21501	1813	RADIUS	Accounting-Response(5) (id=118

Notice the 2 second delays manifest themselves after packets are filtered down to a single session!

Case Study: Visualize Packet Flow

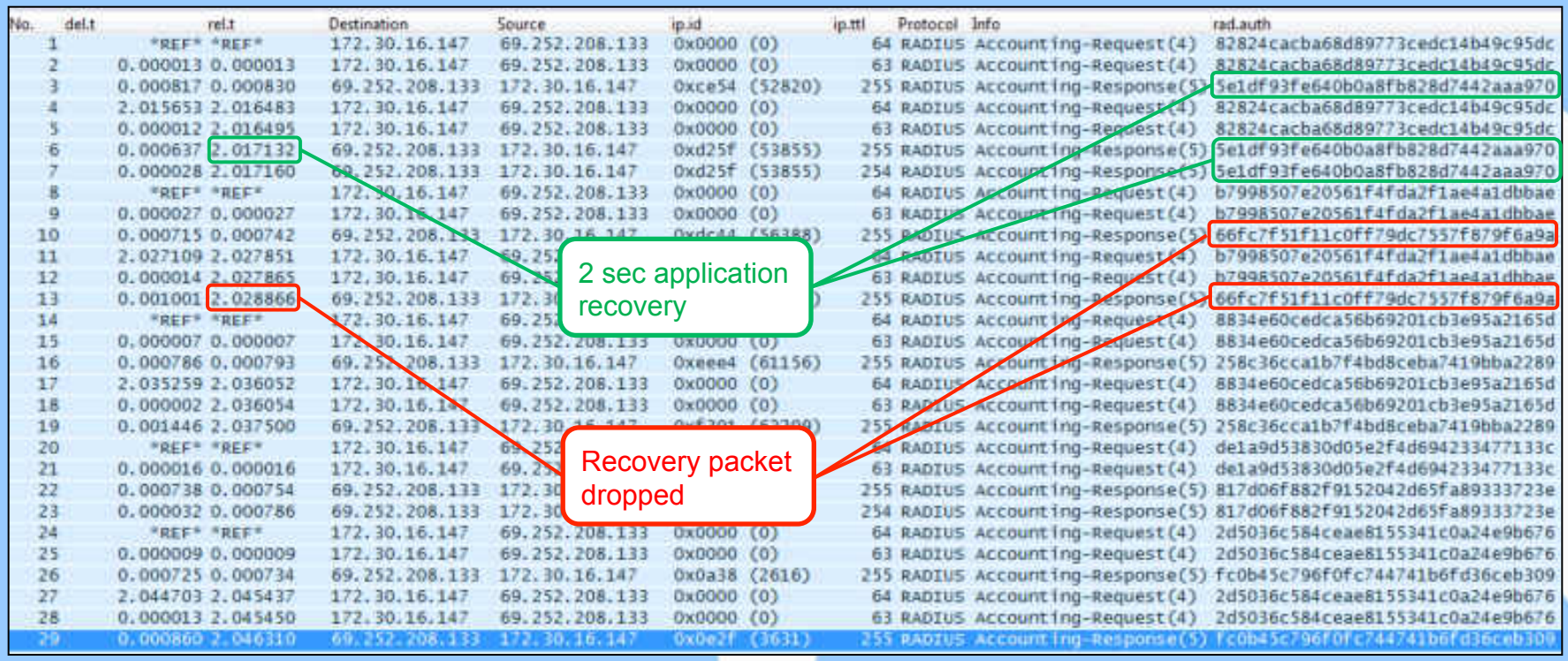
No.	del.t	rel.t	Destination	Source	ip.id	ip.ttl	Protocol	Info
1	0.000000	0.000000	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)
2	0.000013	0.000013	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)
3	0.000817	0.000830	69.252.208.133	172.30.16.147	0xce54 (52820)	255	RADIUS	Accounting-Response(5)
4	2.015653	2.016483	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)
5	0.000012	2.016495	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)
6	0.000637	2.017132	69.252.208.133	172.30.16.147	0xd25f (53855)	255	RADIUS	Accounting-Response(5)
7	0.000028	2.017160	69.252.208.133	172.30.16.147	0xd25f (53855)	254	RADIUS	Accounting-Response(5)
8	4.499100	6.516260	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)
9	0.000027	6.516287	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)
10	0.000715	6.517002	69.252.208.133	172.30.16.147	0xdc44 (56388)	255	RADIUS	Accounting-Response(5)
11	2.027109	8.544111	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)
12	0.000014	8.544125	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)
13	0.001001	8.545126	69.252.208.133	172.30.16.147	0xe0f6 (57590)	255	RADIUS	Accounting-Response(5)
14	6.732012	15.277138	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)
15	0.000007	15.277145	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)
16	0.000786	15.277931	69.252.208.133	172.30.16.147	0xeee4 (61156)	255	RADIUS	Accounting-Response(5)
17	2.035259	17.313190	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)
18	0.000002	17.313192	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)
19	0.001446	17.314638	69.252.208.133	172.30.16.147	0xf301 (62209)	255	RADIUS	Accounting-Response(5)
20	6.735149	24.049787	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)
21	0.000016	24.049803	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)
22	0.000738	24.050541	69.252.208.133	172.30.16.147	0x0043 (67)	255	RADIUS	Accounting-Response(5)
23	0.000032	24.050573	69.252.208.133	172.30.16.147	0x0043 (67)	254	RADIUS	Accounting-Response(5)
24	4.592564	28.643137	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)
25	0.000009	28.643146	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)
26	0.000725	28.643871	69.252.208.133	172.30.16.147	0x0a38 (2616)	255	RADIUS	Accounting-Response(5)
27	2.044703	30.688574	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)
28	0.000013	30.688587	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)
29	0.000860	30.689447	69.252.208.133	172.30.16.147	0x0e2f (3631)	255	RADIUS	Accounting-Response(5)

Technique

Use IP ID and TTL to track packet flow through a router

TTLs allow us to see packet loss inside of the router. IPID=56388 is never shown with TTL=254

Case Study: Correlating for Visibility



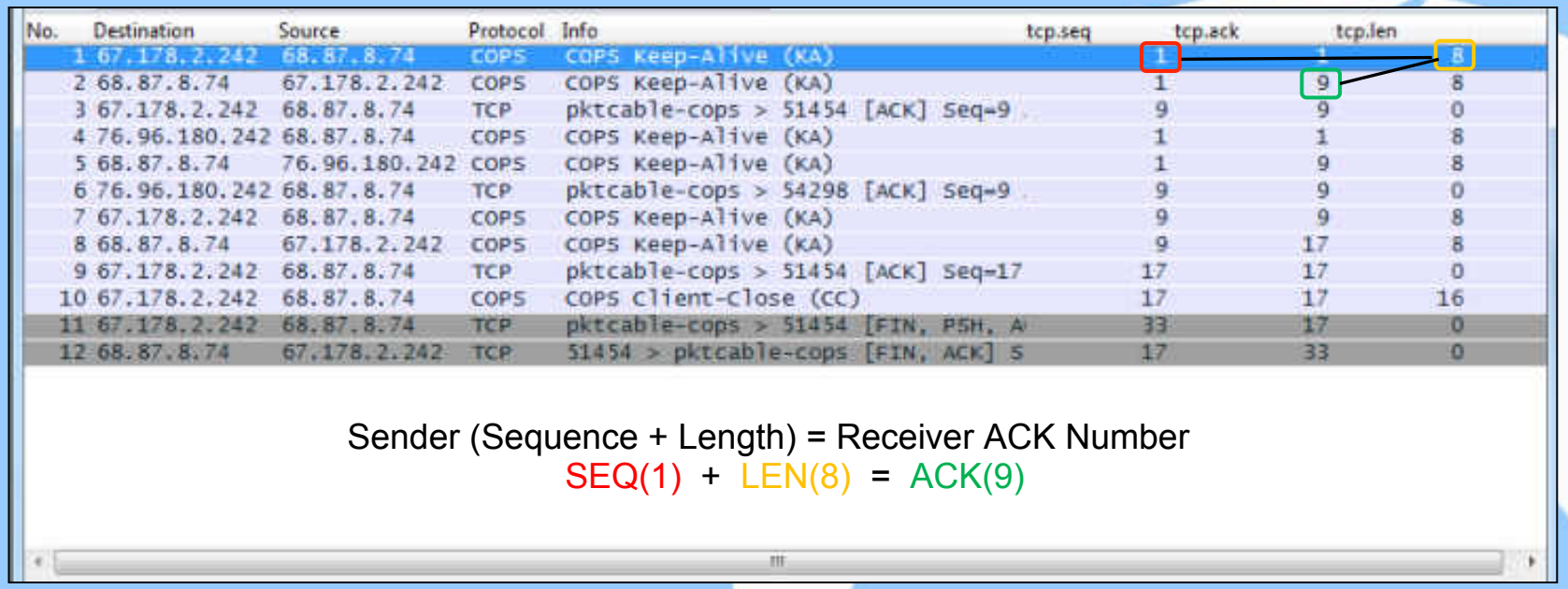
No.	del.t	rel.t	Destination	Source	ip.id	ip.ttl	Protocol	Info	rad.auth
1		*REF*	*REF*	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)	82824cacba68d89773cedc14b49c95dc
2	0.000013	0.000013	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)	82824cacba68d89773cedc14b49c95dc
3	0.000817	0.000830	69.252.208.133	172.30.16.147	0xce54 (52820)	255	RADIUS	Accounting-Response(5)	5e1df93fe640b0a8fb828d7442aaa970
4	2.015653	2.016483	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)	82824cacba68d89773cedc14b49c95dc
5	0.000012	2.016495	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)	82824cacba68d89773cedc14b49c95dc
6	0.000637	2.017132	69.252.208.133	172.30.16.147	0xd25f (53855)	255	RADIUS	Accounting-Response(5)	5e1df93fe640b0a8fb828d7442aaa970
7	0.000028	2.017160	69.252.208.133	172.30.16.147	0xd25f (53855)	254	RADIUS	Accounting-Response(5)	5e1df93fe640b0a8fb828d7442aaa970
8		*REF*	*REF*	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)	b7998507e20561f4fda2f1ae4a1dbbae
9	0.000027	0.000027	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)	b7998507e20561f4fda2f1ae4a1dbbae
10	0.000715	0.000742	69.252.208.133	172.30.16.147	0xdc44 (56388)	255	RADIUS	Accounting-Response(5)	66fc7f51f11c0ff79dc7557f879f6a9a
11	2.027109	2.027851	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)	b7998507e20561f4fda2f1ae4a1dbbae
12	0.000014	2.027865	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)	b7998507e20561f4fda2f1ae4a1dbbae
13	0.001001	2.028866	69.252.208.133	172.30.16.147	0xdc44 (56388)	255	RADIUS	Accounting-Response(5)	66fc7f51f11c0ff79dc7557f879f6a9a
14		*REF*	*REF*	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)	8834e60cedca56b69201cb3e95a2165d
15	0.000007	0.000007	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)	8834e60cedca56b69201cb3e95a2165d
16	0.000786	0.000793	69.252.208.133	172.30.16.147	0xeee4 (61156)	255	RADIUS	Accounting-Response(5)	258c36cca1b7f4bd8ceba7419bba2289
17	2.035259	2.036052	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)	8834e60cedca56b69201cb3e95a2165d
18	0.000002	2.036054	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)	8834e60cedca56b69201cb3e95a2165d
19	0.001446	2.037500	69.252.208.133	172.30.16.147	0xf301 (62209)	255	RADIUS	Accounting-Response(5)	258c36cca1b7f4bd8ceba7419bba2289
20		*REF*	*REF*	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)	de1a9d53830d05e2f4d694233477133c
21	0.000016	0.000016	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)	de1a9d53830d05e2f4d694233477133c
22	0.000738	0.000754	69.252.208.133	172.30.16.147	0x0a38 (2616)	255	RADIUS	Accounting-Response(5)	817d06f882f9152042d65fa89333723e
23	0.000032	0.000786	69.252.208.133	172.30.16.147	0x0a38 (2616)	254	RADIUS	Accounting-Response(5)	817d06f882f9152042d65fa89333723e
24		*REF*	*REF*	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)	2d5036c584ceae8155341c0a24e9b676
25	0.000009	0.000009	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)	2d5036c584ceae8155341c0a24e9b676
26	0.000725	0.000734	69.252.208.133	172.30.16.147	0x0a38 (2616)	255	RADIUS	Accounting-Response(5)	fc0b45c796f0fc744741b6fd36ceb309
27	2.044703	2.045437	172.30.16.147	69.252.208.133	0x0000 (0)	64	RADIUS	Accounting-Request(4)	2d5036c584ceae8155341c0a24e9b676
28	0.000013	2.045450	172.30.16.147	69.252.208.133	0x0000 (0)	63	RADIUS	Accounting-Request(4)	2d5036c584ceae8155341c0a24e9b676
29	0.000860	2.046310	69.252.208.133	172.30.16.147	0x0e2f (3631)	255	RADIUS	Accounting-Response(5)	fc0b45c796f0fc744741b6fd36ceb309

Technique

Correlation of data from different columns in Wireshark allows us to visualize the packet loss inside the router and the attempts by the application to recover from it.

Useful Visualizations

TCP Sequence and Acknowledgements



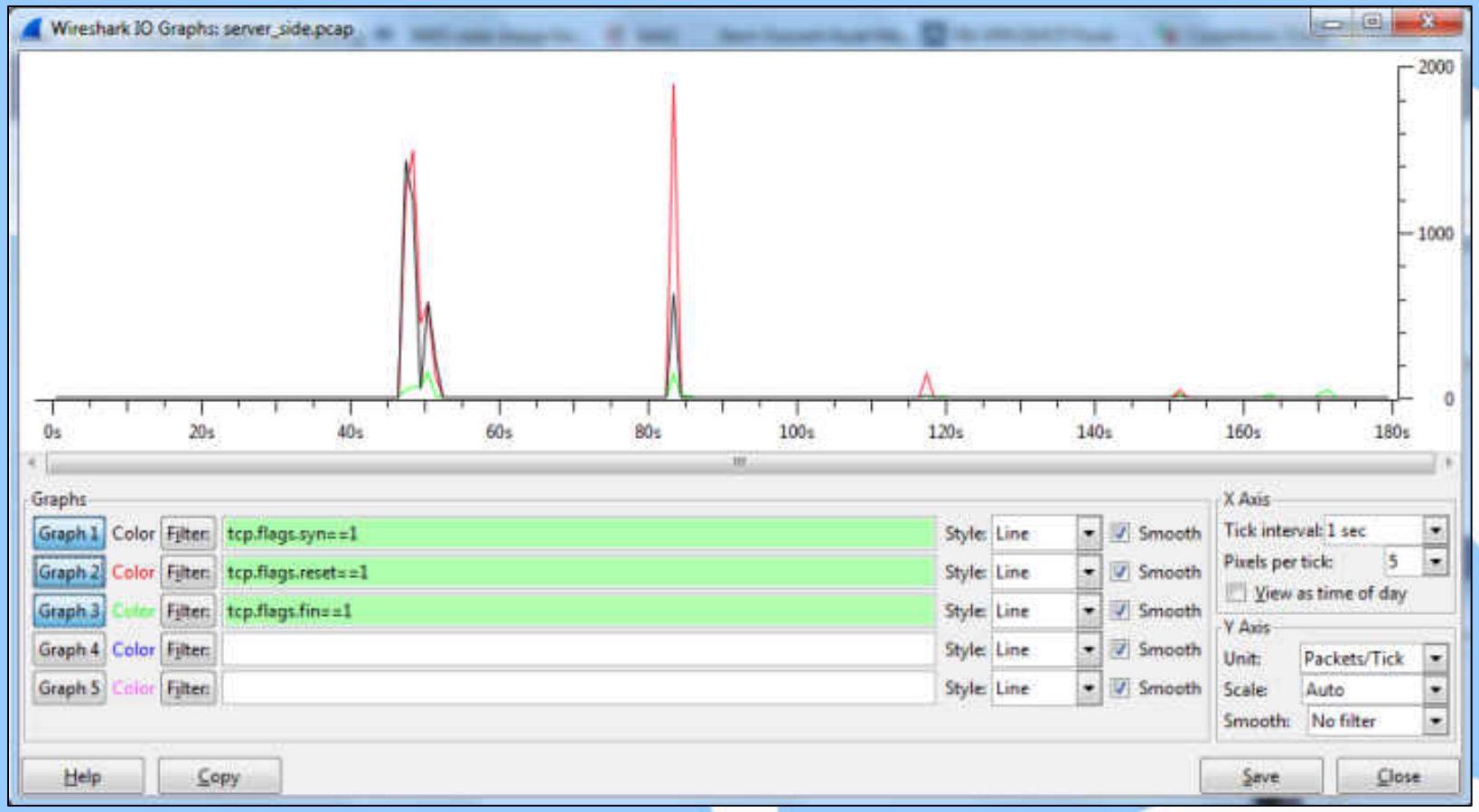
No.	Destination	Source	Protocol	Info	tcp.seq	tcp.ack	tcp.len
1	67.178.2.242	68.87.8.74	COPS	COPS Keep-Alive (KA)	1	1	8
2	68.87.8.74	67.178.2.242	COPS	COPS Keep-Alive (KA)	1	9	8
3	67.178.2.242	68.87.8.74	TCP	pktcable-cops > 51454 [ACK] Seq=9	9	9	0
4	76.96.180.242	68.87.8.74	COPS	COPS Keep-Alive (KA)	1	1	8
5	68.87.8.74	76.96.180.242	COPS	COPS Keep-Alive (KA)	1	9	8
6	76.96.180.242	68.87.8.74	TCP	pktcable-cops > 54298 [ACK] Seq=9	9	9	0
7	67.178.2.242	68.87.8.74	COPS	COPS Keep-Alive (KA)	9	9	8
8	68.87.8.74	67.178.2.242	COPS	COPS Keep-Alive (KA)	9	17	8
9	67.178.2.242	68.87.8.74	TCP	pktcable-cops > 51454 [ACK] Seq=17	17	17	0
10	67.178.2.242	68.87.8.74	COPS	COPS Client-Close (CC)	17	17	16
11	67.178.2.242	68.87.8.74	TCP	pktcable-cops > 51454 [FIN, PSH, A	33	17	0
12	68.87.8.74	67.178.2.242	TCP	51454 > pktcable-cops [FIN, ACK] S	17	33	0

Sender (Sequence + Length) = Receiver ACK Number

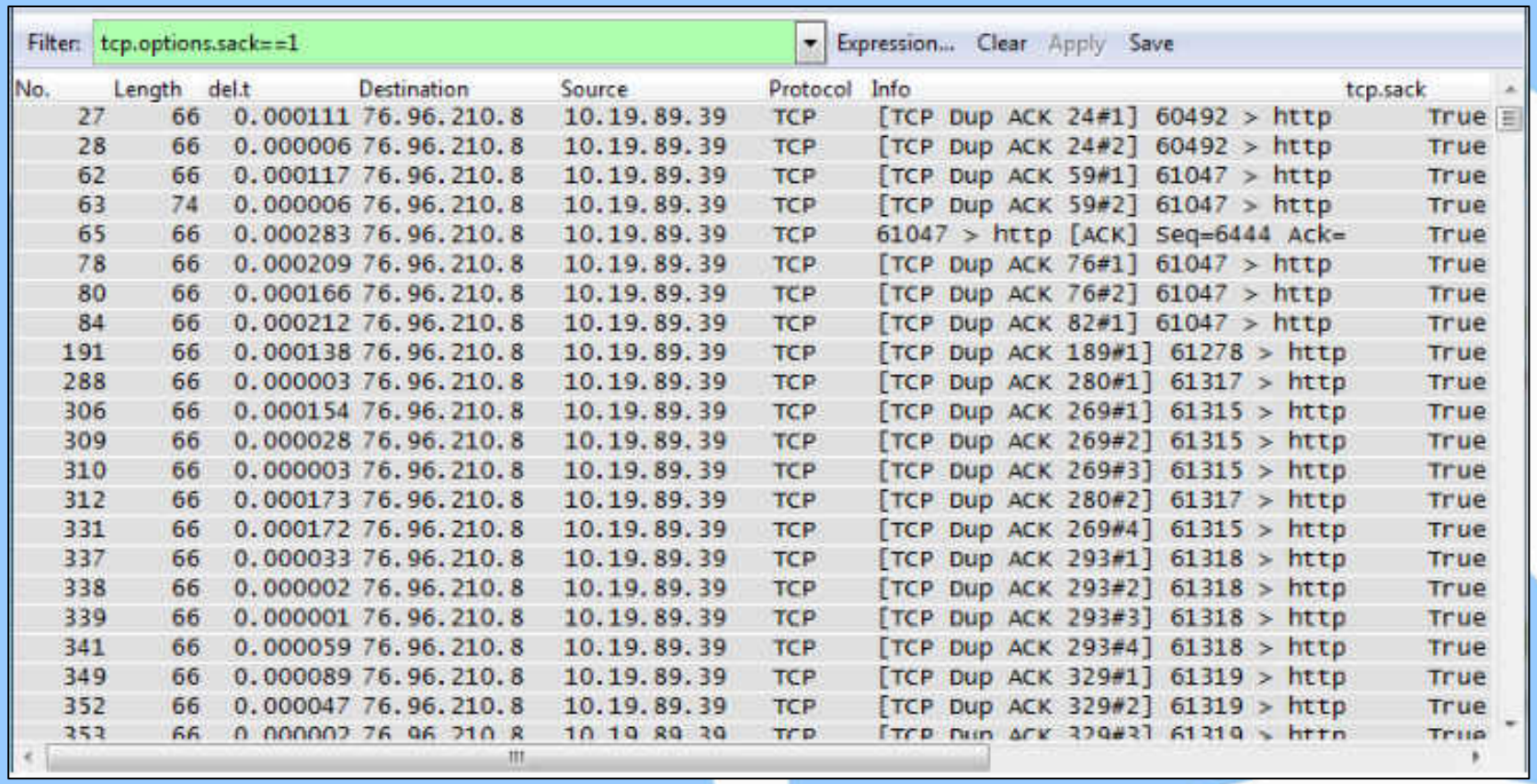
SEQ(1) + LEN(8) = ACK(9)

TCP sequence, acknowledgement, and length fields are invaluable at proving a packet arrived at a destination.

TCP Session Visualization



TCP Selective Acknowledgements

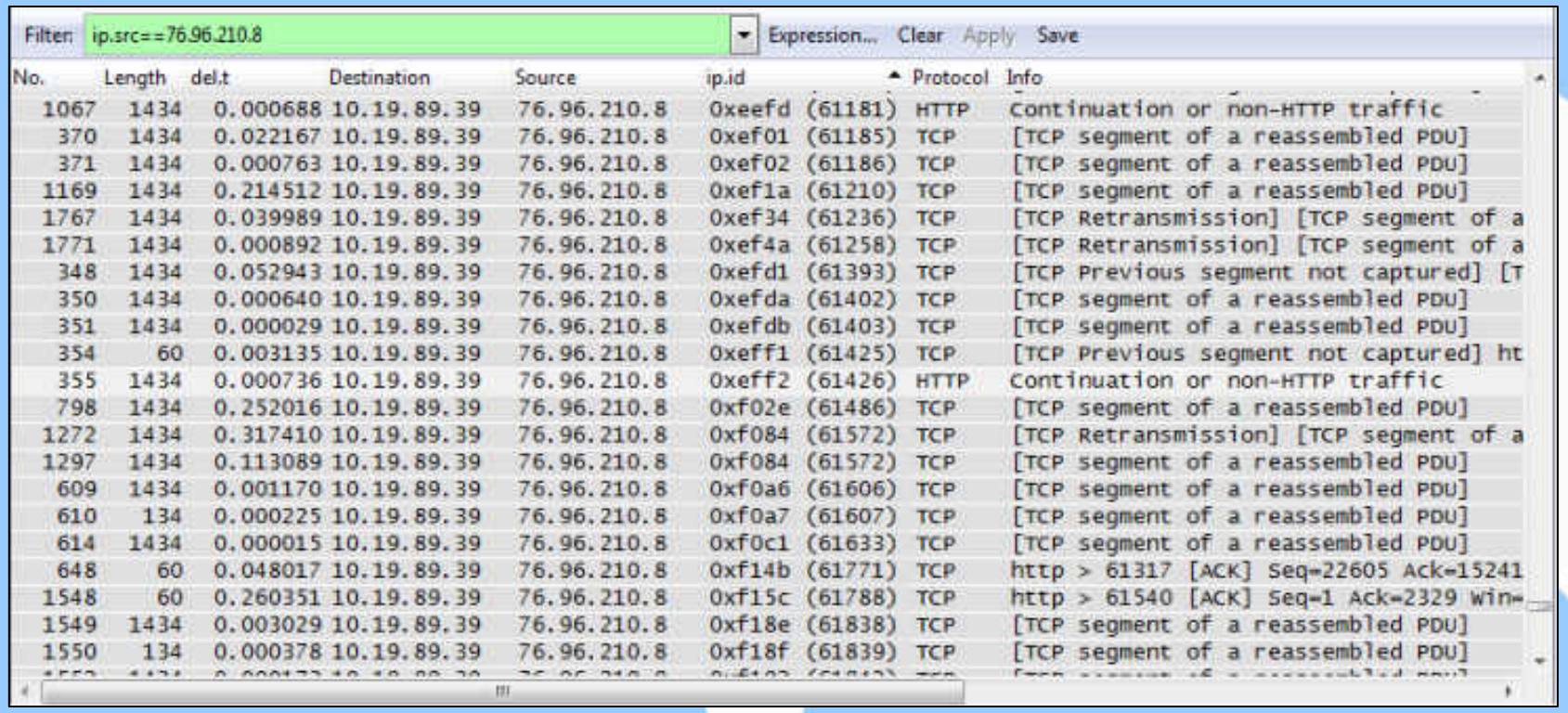


Filter: tcp.options.sack==1 Expression... Clear Apply Save

No.	Length	del.t	Destination	Source	Protocol	Info	tcp.sack
27	66	0.000111	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 24#1] 60492 > http	True
28	66	0.000006	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 24#2] 60492 > http	True
62	66	0.000117	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 59#1] 61047 > http	True
63	74	0.000006	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 59#2] 61047 > http	True
65	66	0.000283	76.96.210.8	10.19.89.39	TCP	61047 > http [ACK] Seq=6444 Ack=	True
78	66	0.000209	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 76#1] 61047 > http	True
80	66	0.000166	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 76#2] 61047 > http	True
84	66	0.000212	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 82#1] 61047 > http	True
191	66	0.000138	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 189#1] 61278 > http	True
288	66	0.000003	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 280#1] 61317 > http	True
306	66	0.000154	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 269#1] 61315 > http	True
309	66	0.000028	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 269#2] 61315 > http	True
310	66	0.000003	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 269#3] 61315 > http	True
312	66	0.000173	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 280#2] 61317 > http	True
331	66	0.000172	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 269#4] 61315 > http	True
337	66	0.000033	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 293#1] 61318 > http	True
338	66	0.000002	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 293#2] 61318 > http	True
339	66	0.000001	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 293#3] 61318 > http	True
341	66	0.000059	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 293#4] 61318 > http	True
349	66	0.000089	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 329#1] 61319 > http	True
352	66	0.000047	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 329#2] 61319 > http	True
353	66	0.000002	76.96.210.8	10.19.89.39	TCP	[TCP Dup ACK 329#3] 61319 > http	True

Filtering on TCP Selective Acknowledgement packets allows us to see the manifestation of unidirectional packet loss

IP Identification Field

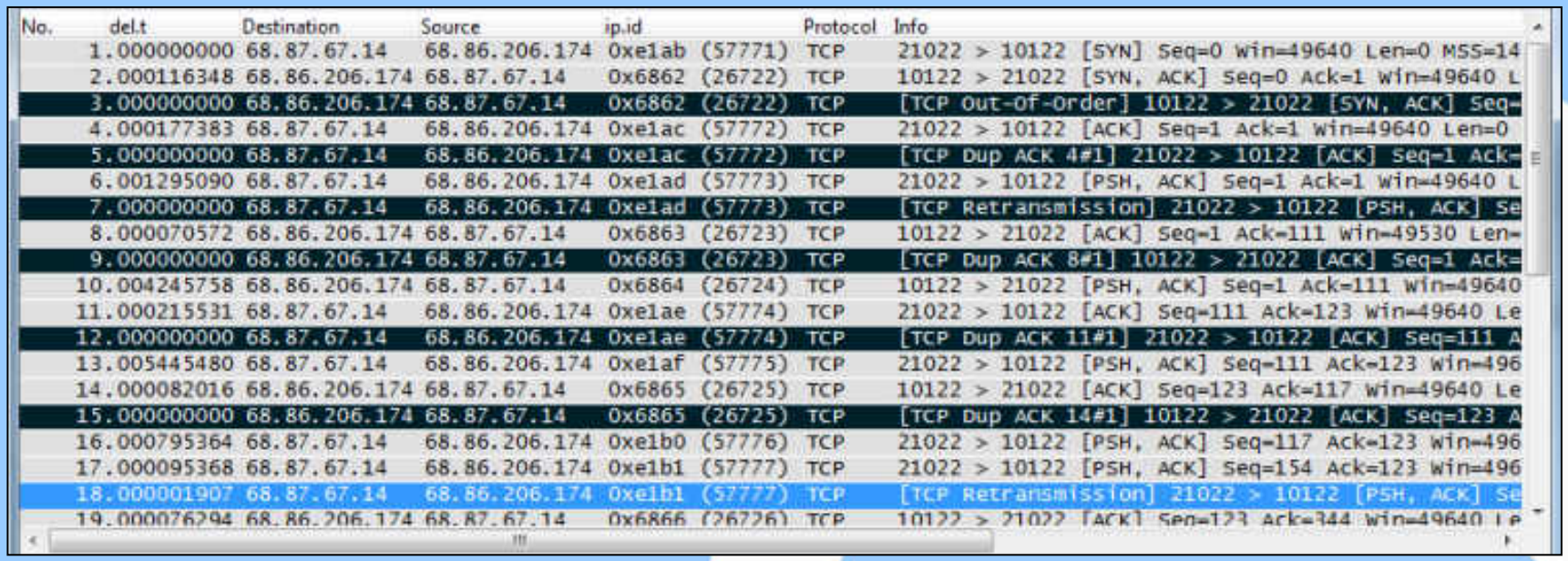


Filter: ip.src==76.96.210.8 Expression... Clear Apply Save

No.	Length	del.t	Destination	Source	ip.id	Protocol	Info
1067	1434	0.000688	10.19.89.39	76.96.210.8	0xeefd (61181)	HTTP	Continuation or non-HTTP traffic
370	1434	0.022167	10.19.89.39	76.96.210.8	0xef01 (61185)	TCP	[TCP segment of a reassembled PDU]
371	1434	0.000763	10.19.89.39	76.96.210.8	0xef02 (61186)	TCP	[TCP segment of a reassembled PDU]
1169	1434	0.214512	10.19.89.39	76.96.210.8	0xef1a (61210)	TCP	[TCP segment of a reassembled PDU]
1767	1434	0.039989	10.19.89.39	76.96.210.8	0xef34 (61236)	TCP	[TCP Retransmission] [TCP segment of a
1771	1434	0.000892	10.19.89.39	76.96.210.8	0xef4a (61258)	TCP	[TCP Retransmission] [TCP segment of a
348	1434	0.052943	10.19.89.39	76.96.210.8	0xefd1 (61393)	TCP	[TCP Previous segment not captured] [T
350	1434	0.000640	10.19.89.39	76.96.210.8	0xefda (61402)	TCP	[TCP segment of a reassembled PDU]
351	1434	0.000029	10.19.89.39	76.96.210.8	0xefdb (61403)	TCP	[TCP segment of a reassembled PDU]
354	60	0.003135	10.19.89.39	76.96.210.8	0xeff1 (61425)	TCP	[TCP Previous segment not captured] ht
355	1434	0.000736	10.19.89.39	76.96.210.8	0xeff2 (61426)	HTTP	Continuation or non-HTTP traffic
798	1434	0.252016	10.19.89.39	76.96.210.8	0xf02e (61486)	TCP	[TCP segment of a reassembled PDU]
1272	1434	0.317410	10.19.89.39	76.96.210.8	0xf084 (61572)	TCP	[TCP Retransmission] [TCP segment of a
1297	1434	0.113089	10.19.89.39	76.96.210.8	0xf084 (61572)	TCP	[TCP segment of a reassembled PDU]
609	1434	0.001170	10.19.89.39	76.96.210.8	0xf0a6 (61606)	TCP	[TCP segment of a reassembled PDU]
610	134	0.000225	10.19.89.39	76.96.210.8	0xf0a7 (61607)	TCP	[TCP segment of a reassembled PDU]
614	1434	0.000015	10.19.89.39	76.96.210.8	0xf0c1 (61633)	TCP	[TCP segment of a reassembled PDU]
648	60	0.048017	10.19.89.39	76.96.210.8	0xf14b (61771)	TCP	http > 61317 [ACK] Seq=22605 Ack=15241
1548	60	0.260351	10.19.89.39	76.96.210.8	0xf15c (61788)	TCP	http > 61540 [ACK] Seq=1 Ack=2329 Win=
1549	1434	0.003029	10.19.89.39	76.96.210.8	0xf18e (61838)	TCP	[TCP segment of a reassembled PDU]
1550	134	0.000378	10.19.89.39	76.96.210.8	0xf18f (61839)	TCP	[TCP segment of a reassembled PDU]

Filtering on a single direction and sorting by the IP ID field allows us to visualize unidirectional packet loss.

Validation using IP Identification



No.	delt	Destination	Source	ip.id	Protocol	Info
1	0.000000000	68.87.67.14	68.86.206.174	0xe1ab (57771)	TCP	21022 > 10122 [SYN] Seq=0 Win=49640 Len=0 MSS=14
2	0.000116348	68.86.206.174	68.87.67.14	0x6862 (26722)	TCP	10122 > 21022 [SYN, ACK] Seq=0 Ack=1 Win=49640 L
3	0.000000000	68.86.206.174	68.87.67.14	0x6862 (26722)	TCP	[TCP Out-Of-Order] 10122 > 21022 [SYN, ACK] Seq=
4	0.000177383	68.87.67.14	68.86.206.174	0xe1ac (57772)	TCP	21022 > 10122 [ACK] Seq=1 Ack=1 Win=49640 Len=0
5	0.000000000	68.87.67.14	68.86.206.174	0xe1ac (57772)	TCP	[TCP Dup ACK 4#1] 21022 > 10122 [ACK] Seq=1 Ack=
6	0.001295090	68.87.67.14	68.86.206.174	0xe1ad (57773)	TCP	21022 > 10122 [PSH, ACK] Seq=1 Ack=1 Win=49640 L
7	0.000000000	68.87.67.14	68.86.206.174	0xe1ad (57773)	TCP	[TCP Retransmission] 21022 > 10122 [PSH, ACK] Se
8	0.000070572	68.86.206.174	68.87.67.14	0x6863 (26723)	TCP	10122 > 21022 [ACK] Seq=1 Ack=111 Win=49530 Len=
9	0.000000000	68.86.206.174	68.87.67.14	0x6863 (26723)	TCP	[TCP Dup ACK 8#1] 10122 > 21022 [ACK] Seq=1 Ack=
10	0.004245758	68.86.206.174	68.87.67.14	0x6864 (26724)	TCP	10122 > 21022 [PSH, ACK] Seq=1 Ack=111 Win=49640
11	0.000215531	68.87.67.14	68.86.206.174	0xe1ae (57774)	TCP	21022 > 10122 [ACK] Seq=111 Ack=123 Win=49640 Le
12	0.000000000	68.87.67.14	68.86.206.174	0xe1ae (57774)	TCP	[TCP Dup ACK 11#1] 21022 > 10122 [ACK] Seq=111 A
13	0.005445480	68.87.67.14	68.86.206.174	0xe1af (57775)	TCP	21022 > 10122 [PSH, ACK] Seq=111 Ack=123 Win=496
14	0.000082016	68.86.206.174	68.87.67.14	0x6865 (26725)	TCP	10122 > 21022 [ACK] Seq=123 Ack=117 Win=49640 Le
15	0.000000000	68.86.206.174	68.87.67.14	0x6865 (26725)	TCP	[TCP Dup ACK 14#1] 10122 > 21022 [ACK] Seq=123 A
16	0.000795364	68.87.67.14	68.86.206.174	0xe1b0 (57776)	TCP	21022 > 10122 [PSH, ACK] Seq=117 Ack=123 Win=496
17	0.000095368	68.87.67.14	68.86.206.174	0xe1b1 (57777)	TCP	21022 > 10122 [PSH, ACK] Seq=154 Ack=123 Win=496
18	0.000001907	68.87.67.14	68.86.206.174	0xe1b1 (57777)	TCP	[TCP Retransmission] 21022 > 10122 [PSH, ACK] Se
19	0.000076294	68.86.206.174	68.87.67.14	0x6866 (26726)	TCP	10122 > 21022 [ACK] Seq=123 Ack=344 Win=49640 Le

Wireshark is confused by duplicate packets and thinks there are DUP ACKs and Retransmissions occurring. IP ID field allows us to see the duplicate IP packets.

Questions?