SharkFest'17 US

Validating Your Packet Capture:
How to be sure you've captured correct & complete data for analysis

Dupes, Drops, and Misses, Oh My!

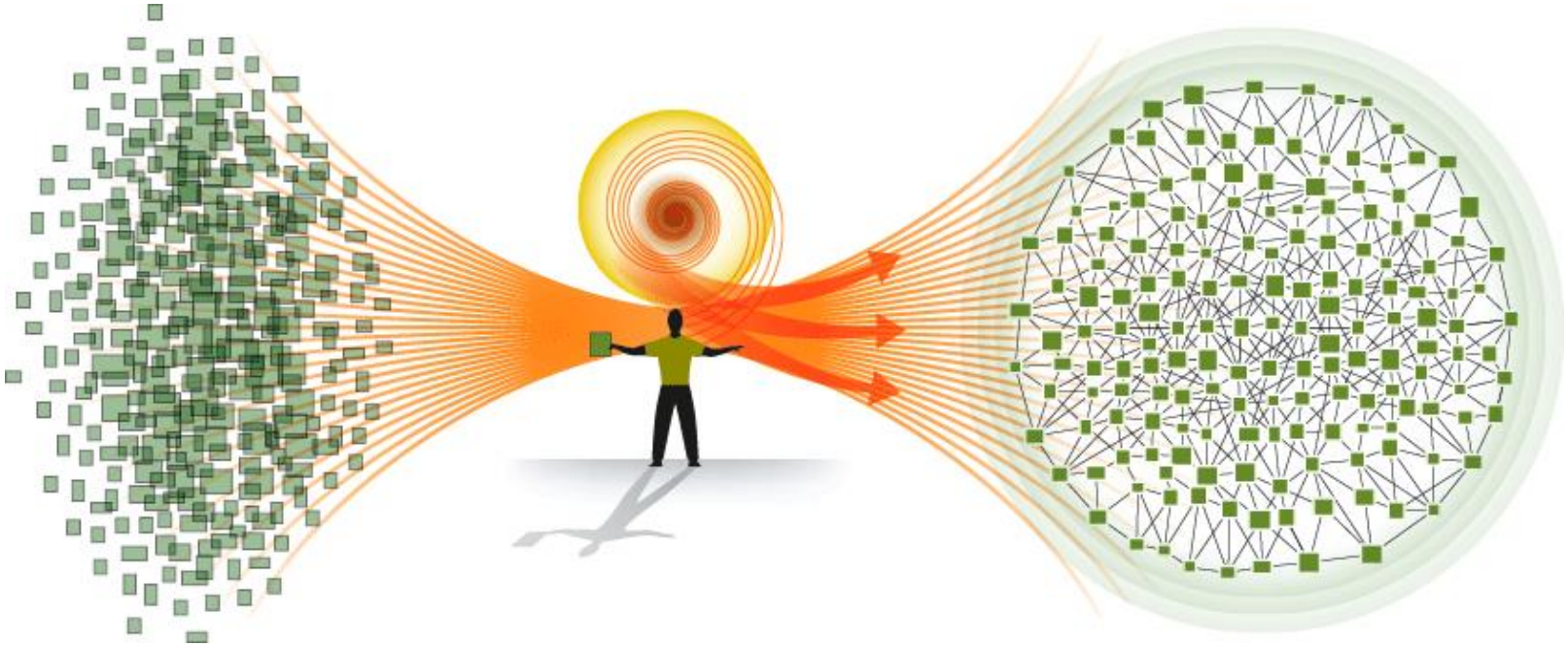
*New title; same product

J. Scott Haugdahl and Mike Canney
Blue Cross Blue Shield of MN and Viavi Solutions

A Cool Visual

Dupes or drops here…

…can throw us over here

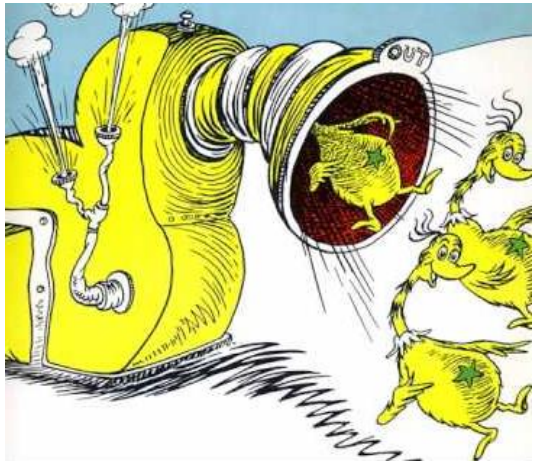


*Source: Fast Company, then Google raw image search

Oh the things you can find if you don’t stay behind!

~ Dr. Seuss

Why Duplicate Packets?



in·ten·tion·al
/inˈten(t)SH(ə)n(ə)l/

adjective

done on purpose; deliberate.

Capture "above and below" switches, internal firewalls, etc. in a single trace to check for presence, latency, drops

Often more convenient than multiple traces and provides a single timestamp source when capturing from visibility fabrics/packet brokers into one analyzer

un·in·ten·tion·al
/ˌəninˈten(t)SH(ə)n(ə)l/

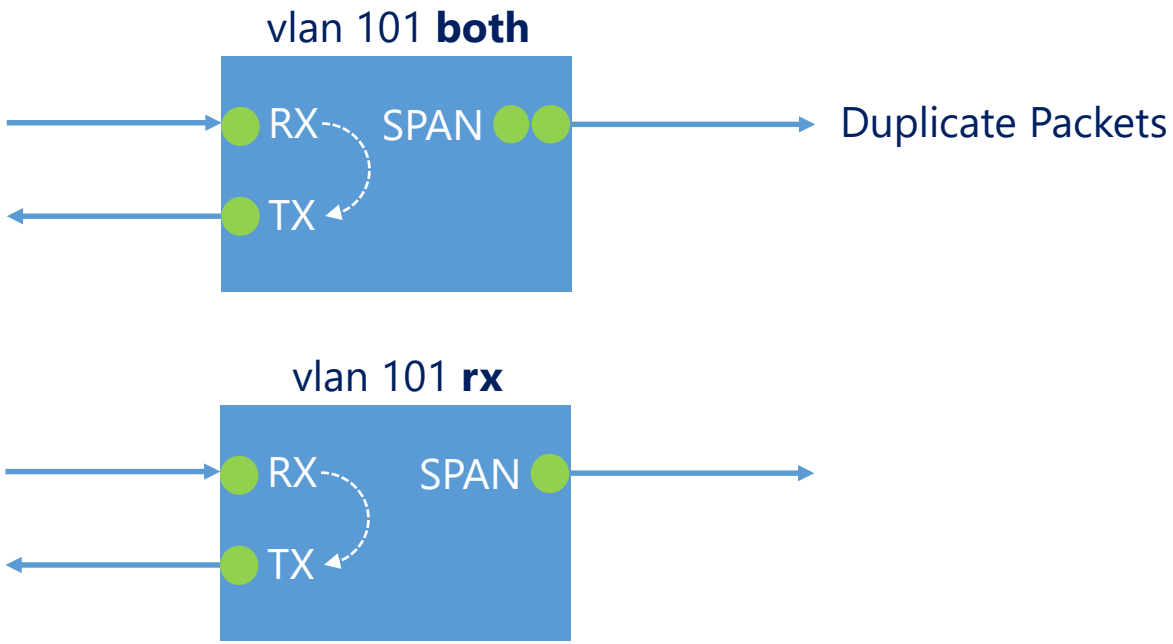
adjective

not done on purpose.

Multi-tier, multi-path, multi-tap capture across the enterprise

Cisco "misconfigured" SPAN on VLANS or port channels

The VLAN SPAN Conundrum



*Solution: Pick VLAN in one direction & specific port(s) for the other**

```
monitor session 1 source vlan 101 rx
monitor session 1 destination interface Gi2/4
```

*Excellent article on all things SPAN – VACLs, VLANs, Virtual VLANS, RSPANS, redundant topologies, etc.
http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Data_Center/ServerFarmSec_2-1/ServSecDC/7_VACL.html

Deduplicating Packets

de·du·pli·ca·tion

/dē ˌd(y)o͞oplə ˈkāSH(ə)n/

noun

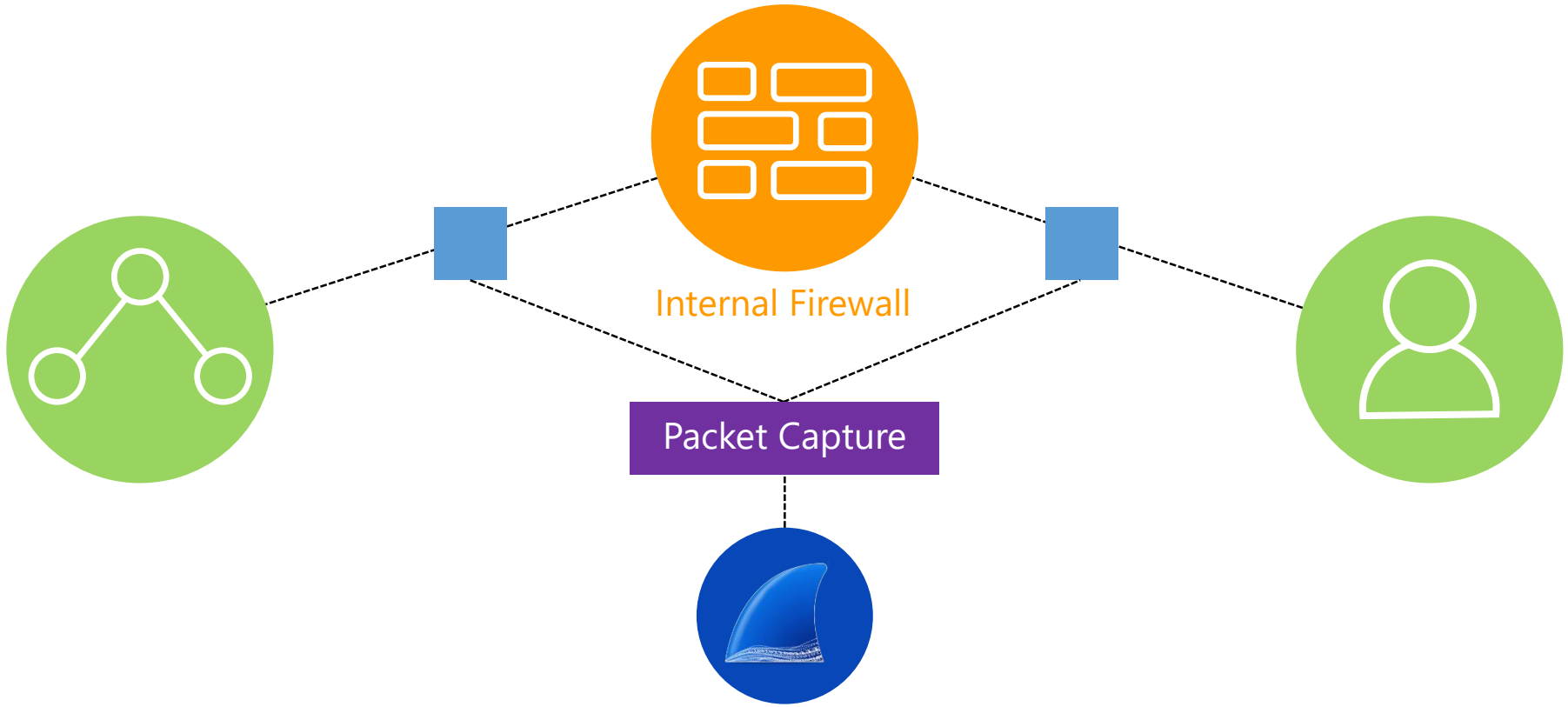
the elimination of duplicate or redundant information, especially in computer data.
"deduplication removes the repetitive information before storing it"

editcap, python scripts*, many analyzers, packet broker dedupers

Or in some cases, use Wireshark creatively!

* http://blogs.cisco.com/security/span-packet-duplication-problem-and-solution

Duplicate Packets: A Simplified Capture Scenario



Hmm… It looks like we have a problem!

tcp.stream eq 0

No.	Length	Delta Time	Source	Destination	Protocol	Info
1	70	0.000000000	10.7.100.55	10.1.101.38	TCP	49255 → 389 [SYN] Seq=958252935 Win=8192 Len=0 MSS=1428 WS=256 SACK_PERM=1
2	70	0.000160000	10.7.100.55	10.1.101.38	TCP	[TCP Out-Of-Order] 49255 → 389 [SYN] Seq=958252935 Win=8192 Len=0 MSS=1428 WS=256 SACK_PERM=1
3	70	0.000376000	10.1.101.38	10.7.100.55	TCP	389 → 49255 [SYN, ACK] Seq=3388464272 Ack=958252936 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=
4	70	0.000020000	10.1.101.38	10.7.100.55	TCP	[TCP Out-Of-Order] 389 → 49255 [SYN, ACK] Seq=3388464272 Ack=958252936 Win=8192 Len=0 MSS=146
5	64	0.000221000	10.7.100.55	10.1.101.38	TCP	49255 → 389 [ACK] Seq=958252936 Ack=3388464273 Win=65536 Len=0
6	60	0.000020000	10.7.100.55	10.1.101.38	TCP	[TCP Dup ACK 5#1] 49255 → 389 [ACK] Seq=958252936 Ack=3388464273 Win=65536 Len=0
7	1486	0.000445000	10.7.100.55	10.1.101.38	TCP	[TCP segment of a reassembled PDU]
8	356	0.000001000	10.7.100.55	10.1.101.38	LDAP	bindRequest(21) "<ROOT>" sasl
9	1486	0.000012000	10.7.100.55	10.1.101.38	TCP	[TCP Out-Of-Order] 49255 → 389 [ACK] Seq=958252936 Ack=3388464273 Win=65536 Len=1428
10	356	0.000001000	10.7.100.55	10.1.101.38	TCP	[TCP Retransmission] 49255 → 389 [PSH, ACK] Seq=958254364 Ack=3388464273 Win=65536 Len=298
11	64	0.000353000	10.1.101.38	10.7.100.55	TCP	389 → 49255 [ACK] Seq=3388464273 Ack=958254662 Win=131328 Len=0
12	60	0.000019000	10.1.101.38	10.7.100.55	TCP	[TCP Dup ACK 11#1] 389 → 49255 [ACK] Seq=3388464273 Ack=958254662 Win=131328 Len=0
13	268	0.001705000	10.1.101.38	10.7.100.55	LDAP	bindResponse(21) success
14	268	0.000016000	10.1.101.38	10.7.100.55	TCP	[TCP Retransmission] 389 → 49255 [PSH, ACK] Seq=3388464273 Ack=958254662 Win=131328 Len=210
15	339	0.000465000	10.7.100.55	10.1.101.38	LDAP	SASL GSS-API Integrity:
16	339	0.000015000	10.7.100.55	10.1.101.38	TCP	[TCP Retransmission] 49255 → 389 [PSH, ACK] Seq=958254662 Ack=3388464483 Win=65280 Len=281
17	1486	0.007729000	10.1.101.38	10.7.100.55	TCP	[TCP segment of a reassembled PDU]
18	1486	0.000001000	10.1.101.38	10.7.100.55	TCP	[TCP segment of a reassembled PDU]
19	1486	0.000001000	10.1.101.38	10.7.100.55	TCP	[TCP segment of a reassembled PDU]
20	1486	0.000002000	10.1.101.38	10.7.100.55	TCP	[TCP segment of a reassembled PDU]
21	1486	0.000011000	10.1.101.38	10.7.100.55	TCP	[TCP Out-Of-Order] 389 → 49255 [ACK] Seq=3388464483 Ack=958254943 Win=131072 Len=1428
22	1486	0.000002000	10.1.101.38	10.7.100.55	TCP	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
23	1486	0.000001000	10.1.101.38	10.7.100.55	TCP	[TCP Out-Of-Order] [TCP segment of a reassembled PDU]
24	1486	0.000003000	10.1.101.38	10.7.100.55	TCP	[TCP Retransmission] 389 → 49255 [ACK] Seq=3388468767 Ack=958254943 Win=131072 Len=1428

A Closer Look

Packets #8 and #10 appear identical with the same packet length, IP addresses, TCP ports, and TCP sequence numbers.

No.	Length	Source	Destination	Protocol	Info
8	356	10.7.100.55	10.1.101.38	LDAP	bindRequest(21) "<ROOT>" sasl
9	1486	10.7.100.55	10.1.101.38	TCP	[TCP Out-Of-Order] 49255 → 389 [ACK] Seq=958252936 Ack=3388464273 Win=65536 Len=1428
10	356	10.7.100.55	10.1.101.38	TCP	[TCP Retransmission] 49255 → 389 [PSH, ACK] Seq=958254364 Ack=3388464273 Win=65536 Len=298

```
▷ Frame 8: 356 bytes on wire (2848 bits), 356 bytes captured (2848 bits)
▷ Ethernet II, Src: Cisco_09:14:41 (d8:67:d9:09:14:41), Dst: CheckPoi_40:6c:1c (00:1c:7f:40:6c:1c)
▷ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 3299
▷ Internet Protocol Version 4, Src: 10.7.100.55 (10.7.100.55), Dst: 10.1.101.38 (10.1.101.38)
▷ Transmission Control Protocol, Src Port: 49255, Dst Port: 389, Seq: 958254364, Ack: 3388464273, Len: 298
```

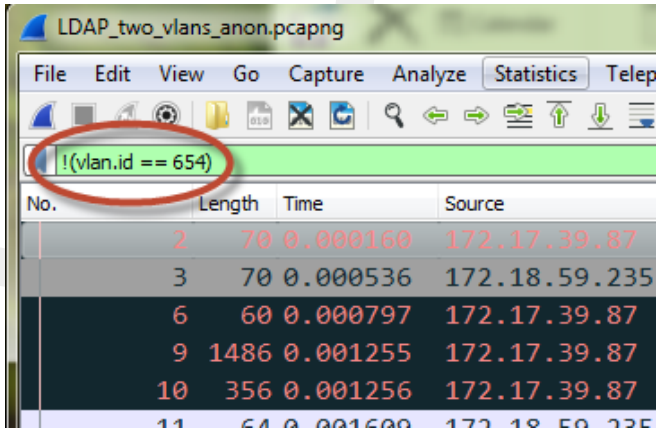
```
▷ Frame 10: 356 bytes on wire (2848 bits), 356 bytes captured (2848 bits)
▷ Ethernet II, Src: CheckPoi_40:6c:1c (00:1c:7f:40:6c:1c), Dst: Cisco_9f:fc:b6 (00:00:0c:9f:fc:b6)
▷ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 3254
▷ Internet Protocol Version 4, Src: 10.7.100.55 (10.7.100.55), Dst: 10.1.101.38 (10.1.101.38)
▷ Transmission Control Protocol, Src Port: 49255, Dst Port: 389, Seq: 958254364, Ack: 3388464273, Len: 298
```

Thus Wireshark flags packet #10 as a retransmission. BUT the packets were captured on two different segments as evidenced by multiple hints. Can you spot them?

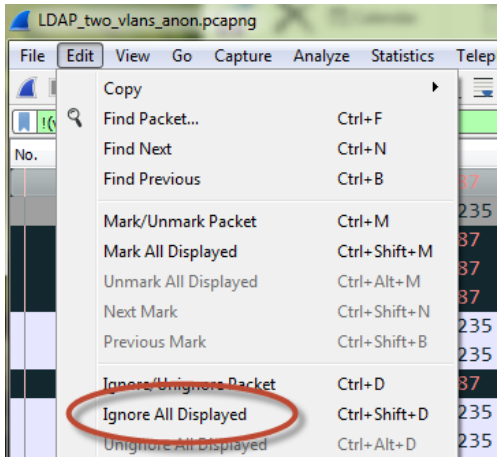
Using Wireshark to “dedupe”

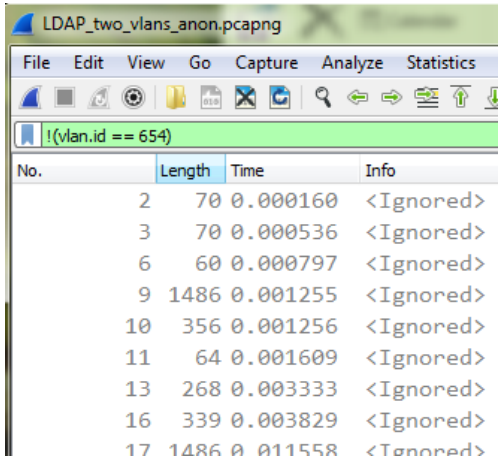
Let’s tell Wireshark to ignore packets not in our VLAN (could we also use MAC addresses)?

1

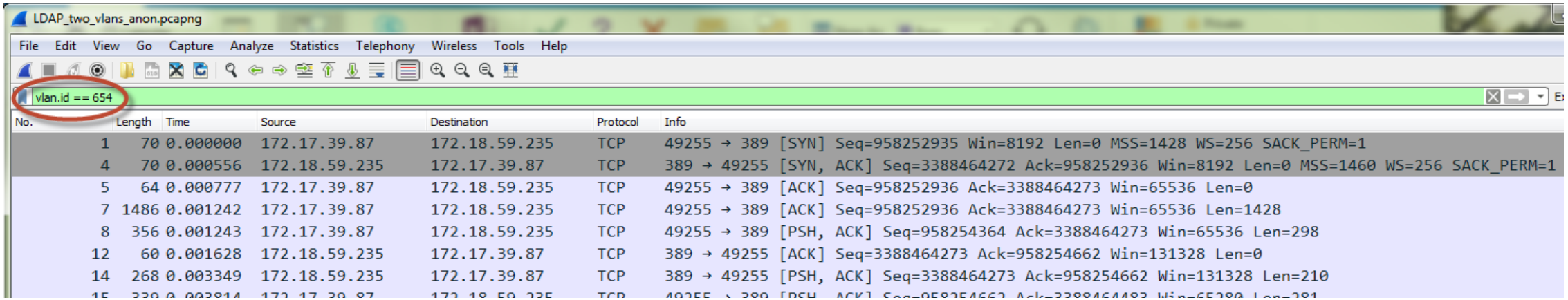


2





3

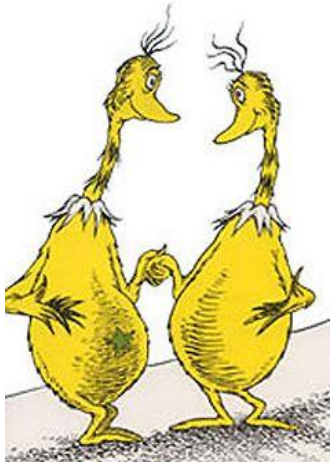


Could Also Export and Read Back In

tcp.stream eq 0

No.	Length	Delta Time	Source	Destination	Protocol	Info
1	70	0.000000	10.7.100.55	10.1.101.38	TCP	49255 → 389 [SYN] Seq=958252935 Win=8192 Len=0 MSS=1428 WS=256 SACK_PERM=1
2	70	0.000536	10.1.101.38	10.7.100.55	TCP	389 → 49255 [SYN, ACK] Seq=3388464272 Ack=958252936 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
3	64	0.000241	10.7.100.55	10.1.101.38	TCP	49255 → 389 [ACK] Seq=958252936 Ack=3388464273 Win=65536 Len=0
4	1486	0.000465	10.7.100.55	10.1.101.38	TCP	[TCP segment of a reassembled PDU]
5	356	0.000001	10.7.100.55	10.1.101.38	LDAP	bindRequest(21) "<ROOT>" sasl
6	64	0.000366	10.1.101.38	10.7.100.55	TCP	389 → 49255 [ACK] Seq=3388464273 Ack=958254662 Win=131328 Len=0
7	268	0.001724	10.1.101.38	10.7.100.55	LDAP	bindResponse(21) success
8	339	0.000481	10.7.100.55	10.1.101.38	LDAP	SASL GSS-API Integrity:
9	1486	0.007744	10.1.101.38	10.7.100.55	TCP	[TCP segment of a reassembled PDU]
10	1486	0.000001	10.1.101.38	10.7.100.55	TCP	[TCP segment of a reassembled PDU]
11	1486	0.000001	10.1.101.38	10.7.100.55	TCP	[TCP segment of a reassembled PDU]
12	1486	0.000002	10.1.101.38	10.7.100.55	TCP	[TCP segment of a reassembled PDU]

Much better!



If you keep your eyes open enough, oh, the stuff you will learn!

~ Dr. Suess

Why Dropped Packets?

These are packets that are present yet not captured nor pre-filtered.

- ☑ Overrun capture platform
- ☑ Aggregating TAP
- ☑ Inside Packet Broker/Visibility Fabric
- ☑ Oversubscribed mirror/SPAN port
- ☑ Error packets not forwarded to mirror/SPAN ports
- ☑ ?

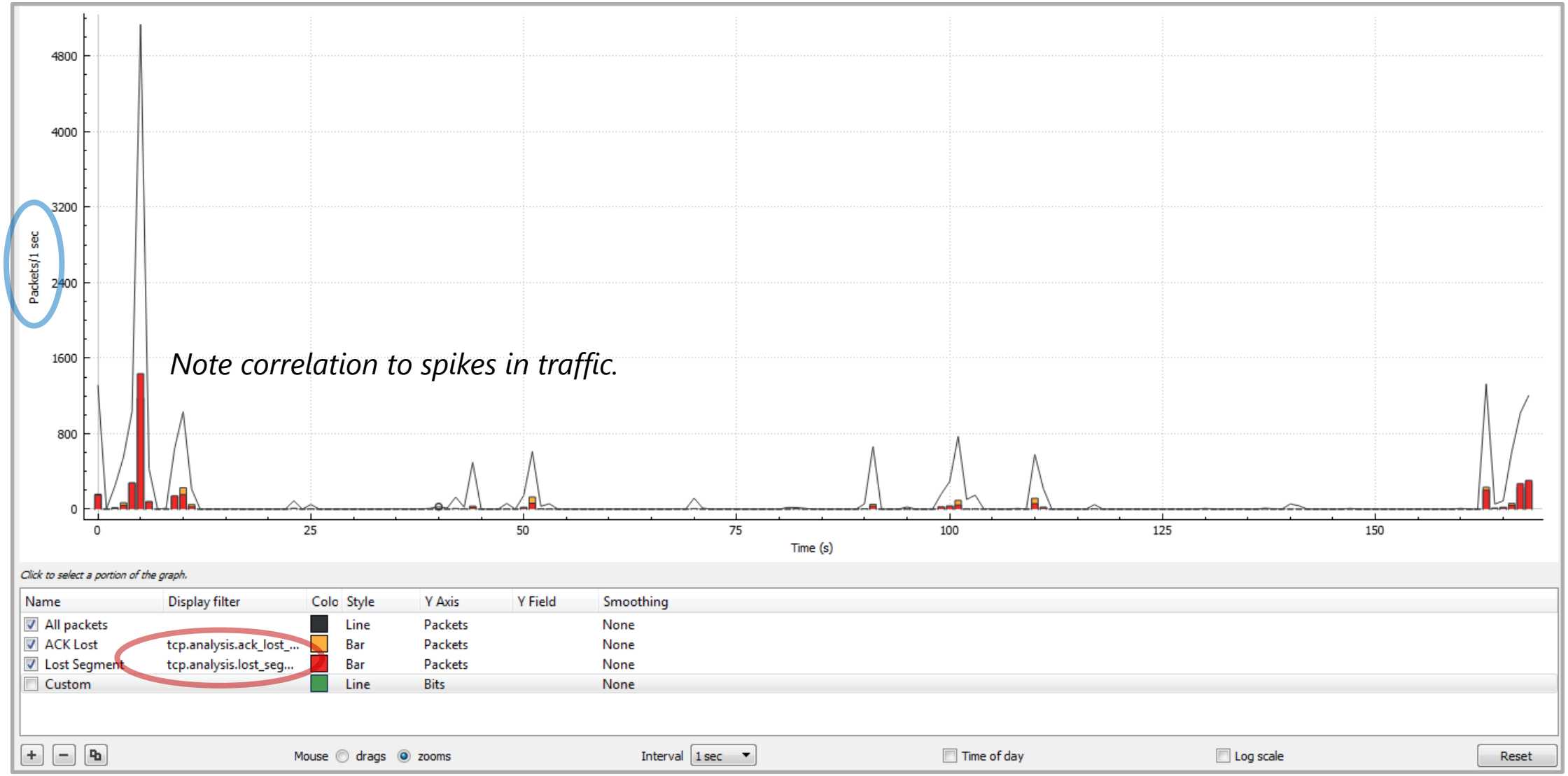
Packet Drops: At the SPAN

"Bar code" is not good!

*Look for **TCP ACKed unseen segment** and **TCP Previous segment not captured***

No.	Length	Time	Source	Destination	Protocol	Info
3982	107	5.382891	10.8.56.28	10.8.16.8	TDS	[TCP Previous segment not captured] Remote Procedu…
3983	1522	5.382905	10.8.16.8	10.8.56.28	TCP	[TCP Previous segment not captured] [TCP segment o…
3984	1522	5.382907	10.8.16.8	10.8.56.28	TCP	[TCP segment of a reassembled PDU]
3985	1522	5.382910	10.8.16.8	10.8.56.28	TCP	[TCP segment of a reassembled PDU]
3986	68	5.382991	10.8.56.28	10.8.16.8	TCP	[TCP ACKed unseen segment] 63815 → 1433 [ACK] Seq=…
3987	101	5.382997	10.8.16.8	10.8.56.28	TDS	Response
3988	1522	5.383042	10.8.16.8	10.8.56.28	TCP	[TCP Previous segment not captured] [TCP segment o…
3989	1522	5.383044	10.8.16.8	10.8.56.28	TCP	[TCP segment of a reassembled PDU]
3990	1522	5.383049	10.8.16.8	10.8.56.28	TCP	[TCP segment of a reassembled PDU]
3991	68	5.383187	10.8.56.28	10.8.16.8	TCP	[TCP ACKed unseen segment] 63815 → 1433 [ACK] Seq=…
3992	68	5.383207	10.8.56.28	10.8.16.8	TCP	[TCP ACKed unseen segment] 63815 → 1433 [ACK] Seq=…
3993	68	5.383225	10.8.56.28	10.8.16.8	TCP	[TCP ACKed unseen segment] 63815 → 1433 [ACK] Seq=…
3994	88	5.383308	10.8.16.8	10.8.56.28	TCP	[TCP ACKed unseen segment] [TCP segment of a reass…
3995	1276	5.383356	10.8.16.8	10.8.56.28	TDS	[TCP Previous segment not captured] Unknown Packet…
3996	1522	5.383390	10.8.16.8	10.8.56.28	TCP	[TCP segment of a reassembled PDU]
3997	1522	5.383392	10.8.16.8	10.8.56.28	TCP	[TCP segment of a reassembled PDU]
3998	1522	5.383395	10.8.16.8	10.8.56.28	TCP	[TCP segment of a reassembled PDU]
3999	68	5.383493	10.8.56.28	10.8.16.8	TCP	[TCP ACKed unseen segment] 63815 → 1433 [ACK] Seq=…

SPAN Drops: A Graphical View



SPAN Drops: How Bad?

1. Let's **isolate a stream**

(ip.addr eq 10.8.56.28 and ip.addr eq 10.8.16.8) and (tcp.port eq 63815 and tcp.port eq 1433)

No.	Length	Time	Source	Destination	Protocol	Info
479	68	2.420468	10.8.56.28	10.8.16.8	TCP	63815 → 1433 [ACK] Seq=4112872093 Ack=3272284726 Win=4102 Len=0
527	191	2.445173	10.8.56.28	10.8.16.8	TDS	Remote Procedure Call
528	456	2.445767	10.8.16.8	10.8.56.28	TDS	Response
529	223	2.446481	10.8.16.8	10.8.56.28	TDS	[TCP ACKed unseen segment] Response
530	125	2.446577	10.8.56.28	10.8.16.8	TDS	[TCP Previous segment not captured] Remote Procedure Call
531	223	2.447135	10.8.16.8	10.8.56.28	TDS	[TCP ACKed unseen segment] Response
532	107	2.447239	10.8.56.28	10.8.16.8	TDS	Remote Procedure Call
533	88	2.447374	10.8.16.8	10.8.56.28	TDS	Response
605	68	2.467352	10.8.56.28	10.8.16.8	TCP	63815 → 1433 [ACK] Seq=4112872402 Ack=3272285468 Win=4104 Len=0

2. Checking the **expert stats**, we see lots of drops relative to packets captured for this stream

Severity	Summary	Group	Protocol	Count
Warning	ACKed segment that wasn't captured (common at capture...	Sequence	TCP	137
Warning	Previous segment not captured (common at capture start)	Sequence	TCP	159

Display filter: "(ip.addr eq 10.8.16.8 and ip.addr eq 10.8.56.28) and (tcp.port eq 1433 and tcp.port eq 63815)"

☑ Limit to Display Filter ☑ Group by summary Search: Show...

Make sure this is checked

Packets: 20000 · Displayed: 625 (3.1%) · Load time: 0:0.358

3. Per **TCP sequence numbers**, we should have 2,227,300 bytes of payload

```
Transmission Control Protocol, Src Port: 1433, Dst Port: 63815, Seq: 3274559693, Ack: 4112881420, Len: 35
Transmission Control Protocol, Src Port: 1433, Dst Port: 63815, Seq: 3272282394, Ack: 4112871186, Len: 394
```

Last Seq # - First Seq # + 1 = 2,277,300 bytes of TCP payload
(or show relative sequence numbers in Wireshark)

4. Yet our **TCP conversation stats** show 485k bytes or a ***loss of nearly 80%***

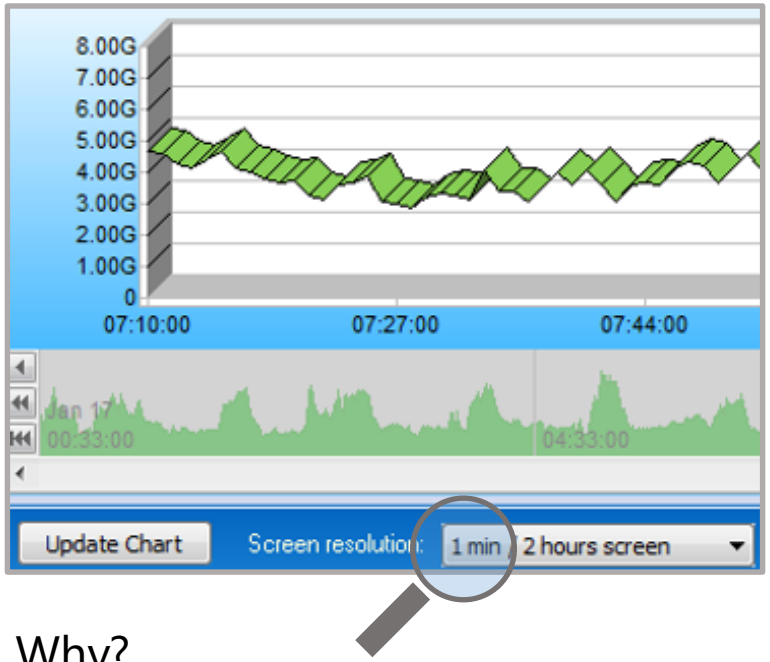
Ethernet · 2 | IPv4 · 1 | IPv6 | TCP · 1 | UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
10.8.56.28	63815	10.8.16.8	1433	623	509 k	219	23 k	404	485 k	0.004588	168.2151

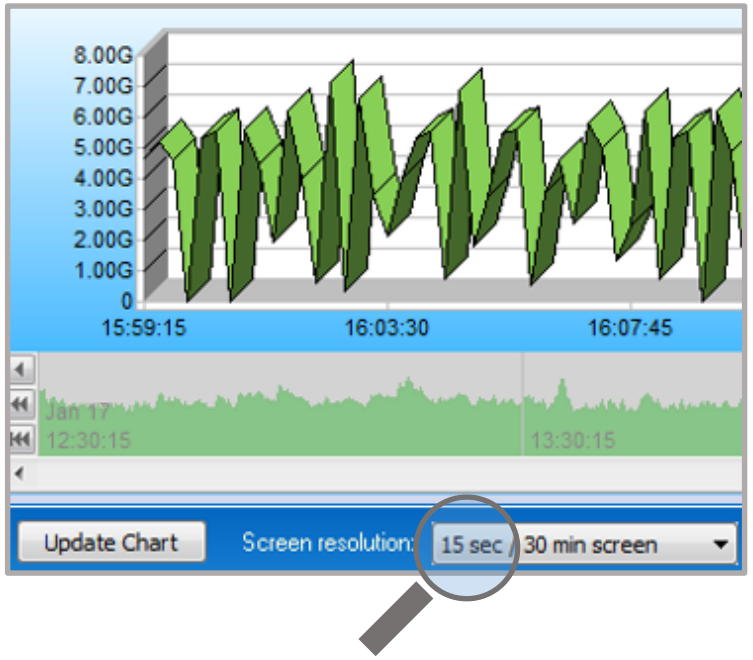
Packet Drops: At the Data Center Analyzer

Real-time Dashboard – Continuous Packet Capture

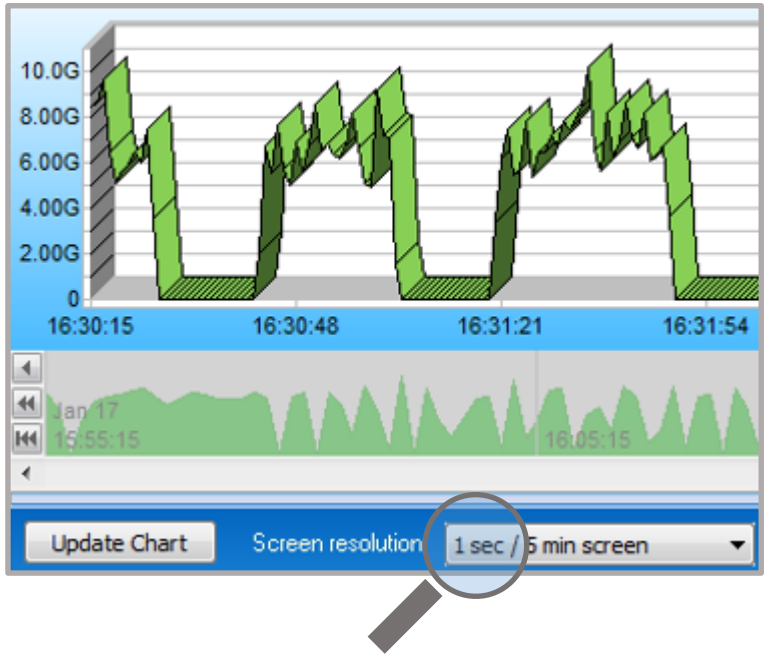
Looking Good…



Wait, something's going on!



…Loss is catastrophic



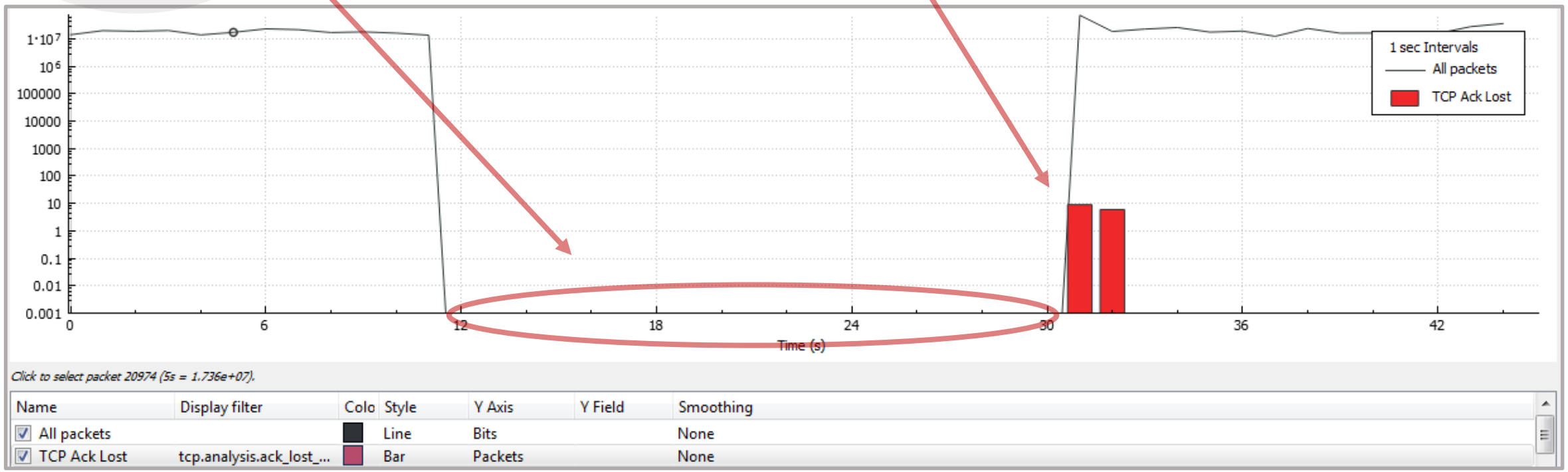
Why?

A packet FIFO (system RAM) acts as an elastic buffer between the NIC and disk until full in which case the *entire* FIFO must be written to disk before continuing!

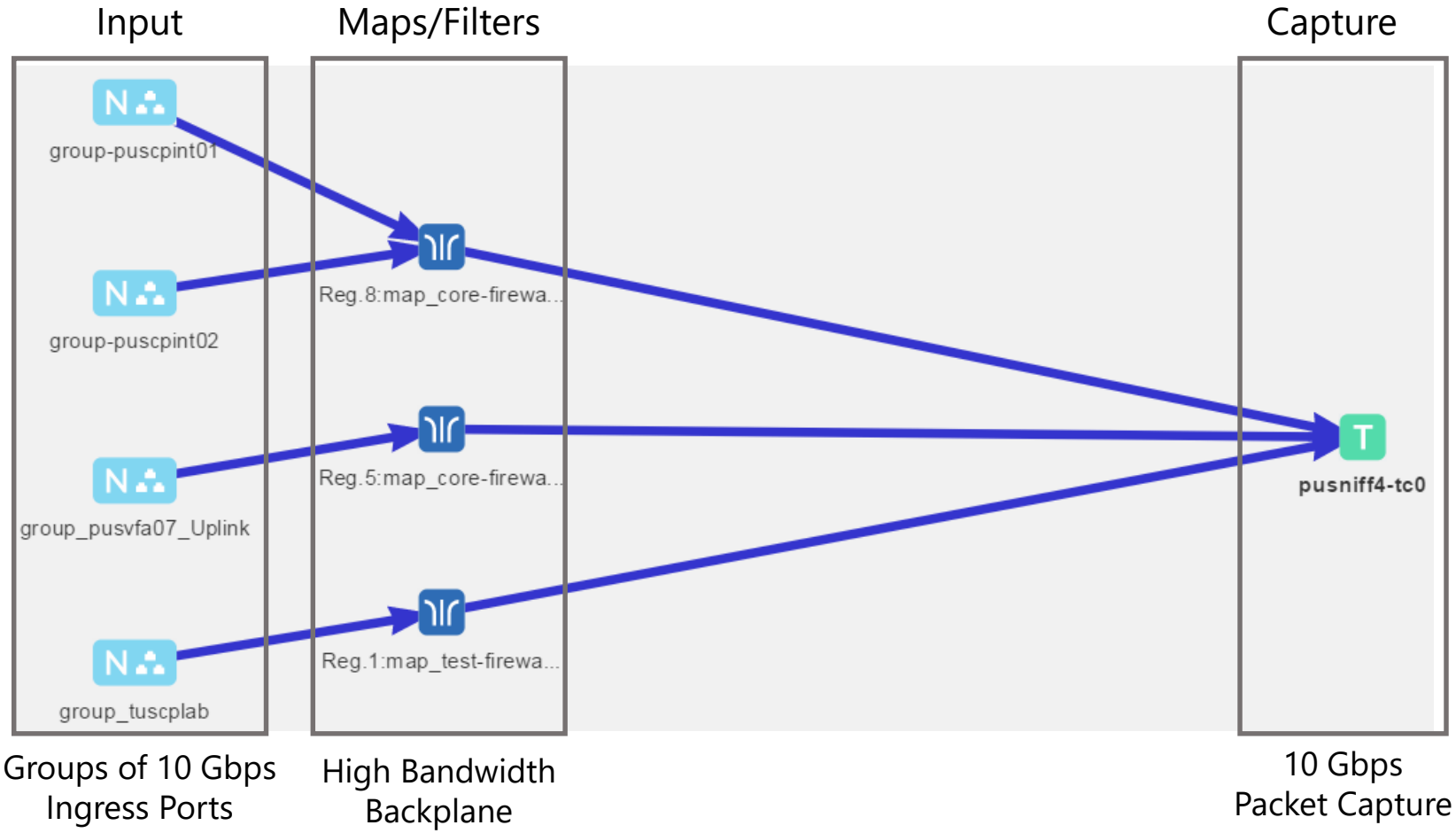
OK! OK! BAD!

Packet Drops: Catastrophic!

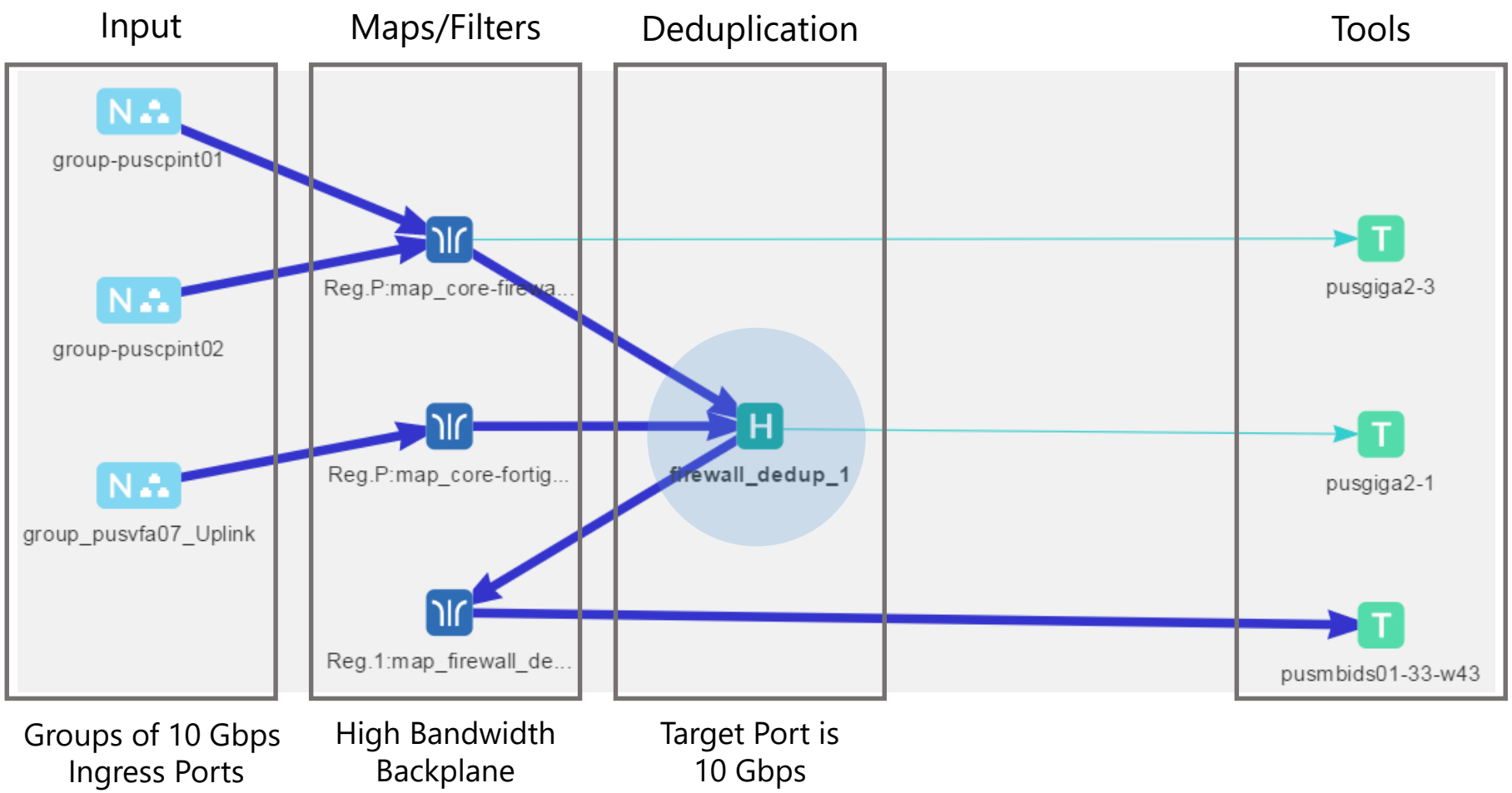
41896	185	11.617517	192.168.175.134	192.168.165.169	TCP	59910 → 1446 [PSH, ACK] Seq=42087 Ack=1622843 Win=4106 Len=127
41897	269	11.617785	192.168.165.169	192.168.175.134	TCP	1446 → 59910 [PSH, ACK] Seq=1622843 Ack=42214 Win=4104 Len=211
41898	209	11.617797	192.168.175.134	192.168.165.169	TCP	60022 → 1446 [PSH, ACK] Seq=2115021 Ack=14154461 Win=4101 Len=151
41899	185	31.699939	192.168.175.134	192.168.165.169	TCP	[TCP ACKed unseen segment] [TCP Previous segment not captured] 60022 → 1446 [PSH,…
41900	1072	31.700331	192.168.165.169	192.168.175.134	TCP	[TCP ACKed unseen segment] [TCP Previous segment not captured] 1446 → 60022 [PSH,…
41901	187	31.700903	192.168.175.134	192.168.165.169	TCP	[TCP ACKed unseen segment] 60022 → 1446 [PSH, ACK] Seq=3909212 Ack=26394957 Win=4…
41902	1518	31.702085	192.168.165.169	192.168.175.134	TCP	1446 → 60022 [ACK] Seq=26394957 Ack=3909341 Win=4105 Len=1460
41903	1518	31.702088	192.168.165.169	192.168.175.134	TCP	1446 → 60022 [ACK] Seq=26396417 Ack=3909341 Win=4105 Len=1460



Packet Drops: Simple Packet Broker Case



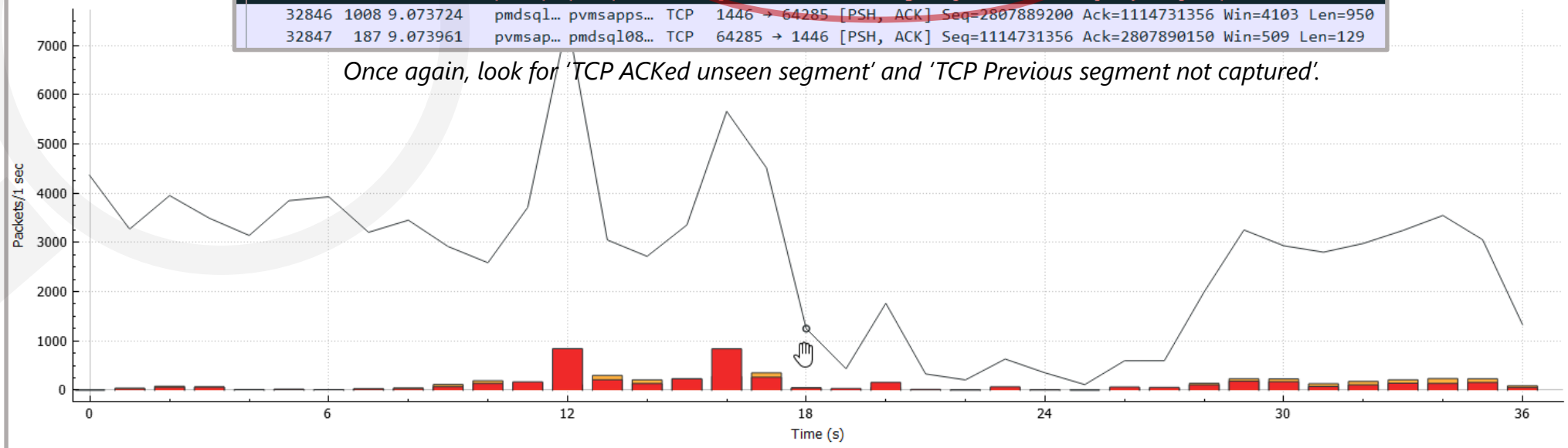
Packet Drops: Complex Packet Broker Case



Packet Drops: Complex Packet Broker Case

No.	Length	Time	Source	Destination	Protocol	Info
32839	64	9.069860	pvmsap…	pmdsql08…	TCP	64285 → 1446 [ACK] Seq=1114730776 Ack=2807887223 Win=513 Len=0
32840	209	9.070161	pvmsap…	pmdsql08…	TCP	64285 → 1446 [PSH, ACK] Seq=1114730776 Ack=2807887223 Win=513 Len=151
32841	717	9.070789	pmdsql…	pvmsapps…	TCP	1446 → 64285 [PSH, ACK] Seq=2807887223 Ack=1114730927 Win=4105 Len=659
32842	209	9.071210	pvmsap…	pmdsql08…	TCP	64285 → 1446 [PSH, ACK] Seq=1114730927 Ack=2807887882 Win=510 Len=151
32843	209	9.071985	pvmsap…	pmdsql08…	TCP	[TCP ACKed unseen segment] 64285 → 1446 [PSH, ACK] Seq=1114731078 Ack=2
32844	717	9.072369	pmdsql…	pvmsapps…	TCP	[TCP Previous segment not captured] 1446 → 64285 [PSH, ACK] Seq=2807888
32845	185	9.073204	pvmsap…	pmdsql08…	TCP	[TCP ACKed unseen segment] 64285 → 1446 [PSH, ACK] Seq=1114731229 Ack=2
32846	1008	9.073724	pmdsql…	pvmsapps…	TCP	1446 → 64285 [PSH, ACK] Seq=2807889200 Ack=1114731356 Win=4103 Len=950
32847	187	9.073961	pvmsap…	pmdsql08…	TCP	64285 → 1446 [PSH, ACK] Seq=1114731356 Ack=2807890150 Win=509 Len=129

Once again, look for 'TCP ACKed unseen segment' and 'TCP Previous segment not captured'.



Click to select packet 69872 (18s = 1247).

Name	Display filter	Colo	Style	Y Axis	Y Field	Smoothing
☑ All packets			Line	Packets		None
☑ ACK Lost	tcp.analysis.ack_lost_segment		Bar	Packets		None
☑ Lost Segment	tcp.analysis.lost_segment		Bar	Packets		None

+ − Mouse ◉ drags ○ zooms Interval 1 sec ☐ Time of day ☐ Log scale Reset

Thank You!