TCP Performance: SACK / BIF

Nalini Elkins
Inside Products, Inc.

Inside Products, Inc.
(831) 659-8360
www.insidethestack.com

Copyright

Agenda

- What is SACK?
- How does ACK work?
- Out of Order / Retransmit
- SACK packets
- How to calculate Bytes in Flight(BIF)
- Why?
- Comparisons

From RFC2018

- RFC2018 defines Selective Acknowledgement (SACK) as follows:

"TCP may experience poor performance when multiple packets are lost from one window of data. With the limited information available from cumulative acknowledgements, a TCP sender can only learn about a single lost packet per round trip time. An aggressive sender could choose to retransmit packets early, but such retransmitted segments may have already been successfully received."

From RFC2018

- So, what is the solution?

- (From RFC)

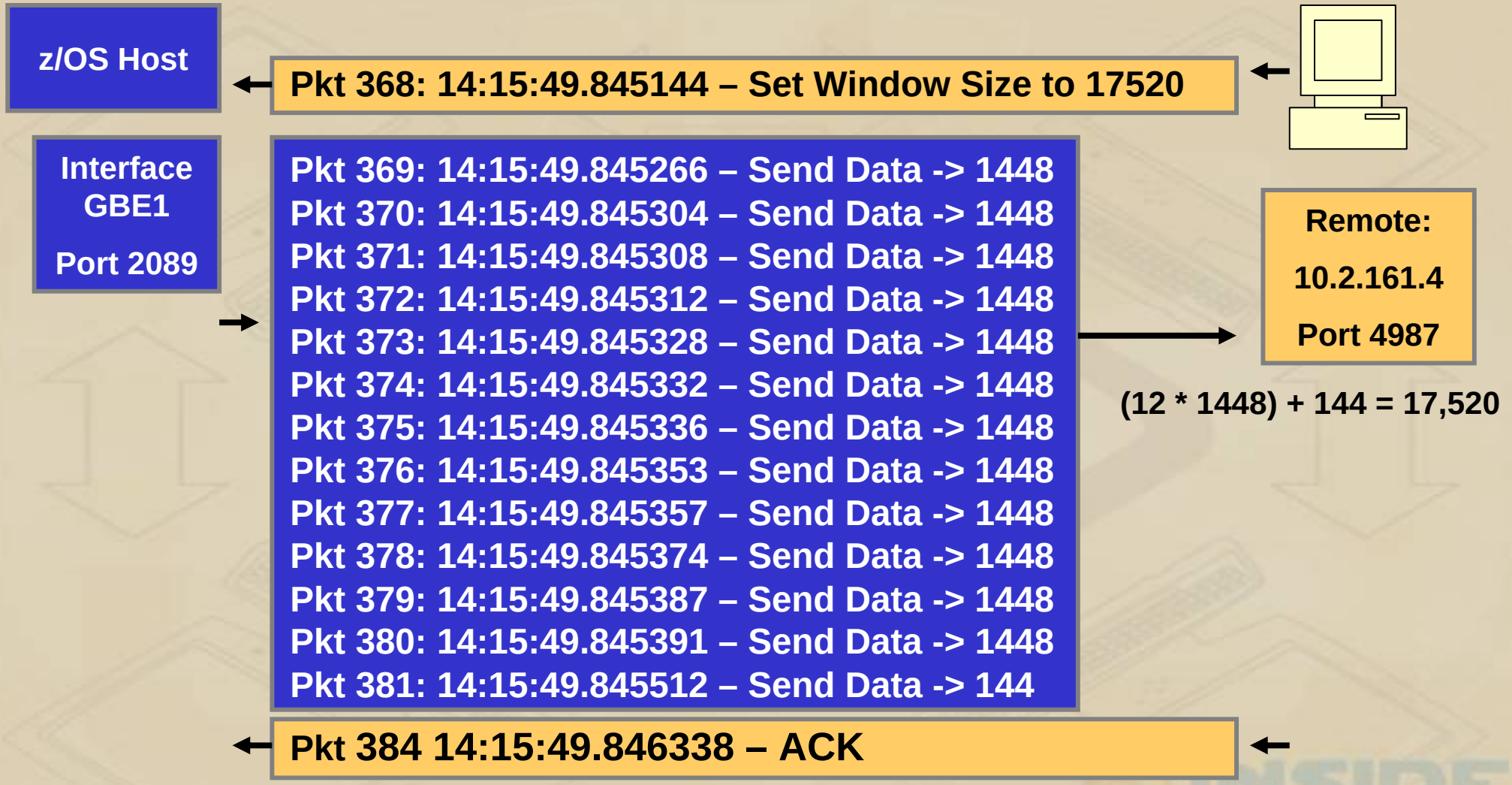
  “A Selective Acknowledgment (SACK) mechanism, combined with a selective repeat retransmission policy, can help to overcome these limitations. The receiving TCP sends back SACK packets to the sender informing the sender of data that has been received. The sender can then retransmit only the missing data segments.”

```
1 HOST2     PACKET   00000001 13:31:51.410583 Packet Trace
  From Interface    : CISCO1              Device: CLAW              Full=48
   Tod Clock        : 2011/10/14 13:31:51.410582
   Sequence #       : 0                   Flags: Pkt Ver2
   Source Port      : 1080                Dest Port: 23
  IpHeader: Version : 4                   Header Length
   Tos              : 00                  QOS: Routine
   Packet Length    : 48                  ID Number: 02
   Fragment         :                     Offset: 0
   TTL              : 125                 Protocol: TCP
   Source           : 192.168.145.7
   Destination      : 10.201.0.2

TCP
   Source Port      : 1080  ()            Destination Port: 23    (telnet)
   Sequence Number  : 3010274480          Ack Number: 0
   Header Length    : 28                  Flags: Syn
   Window Size      : 17520               CheckSum: 0A9C FFFF Urgent Ptr: 0000
    Option          : Max Seg Size Len: 4 MSS: 1448
    Option          : NOP
    Option          : NOP
    Option          : SACK Permitted

 IP Header          : 20
  000000 45000030 026E0000 7D06DEDF C0A89107  0AC900

 Protocol Header    : 28
  000000 04380017 B36D24B0 00000000 70024000  0A9C0000 02040550 01010402
```

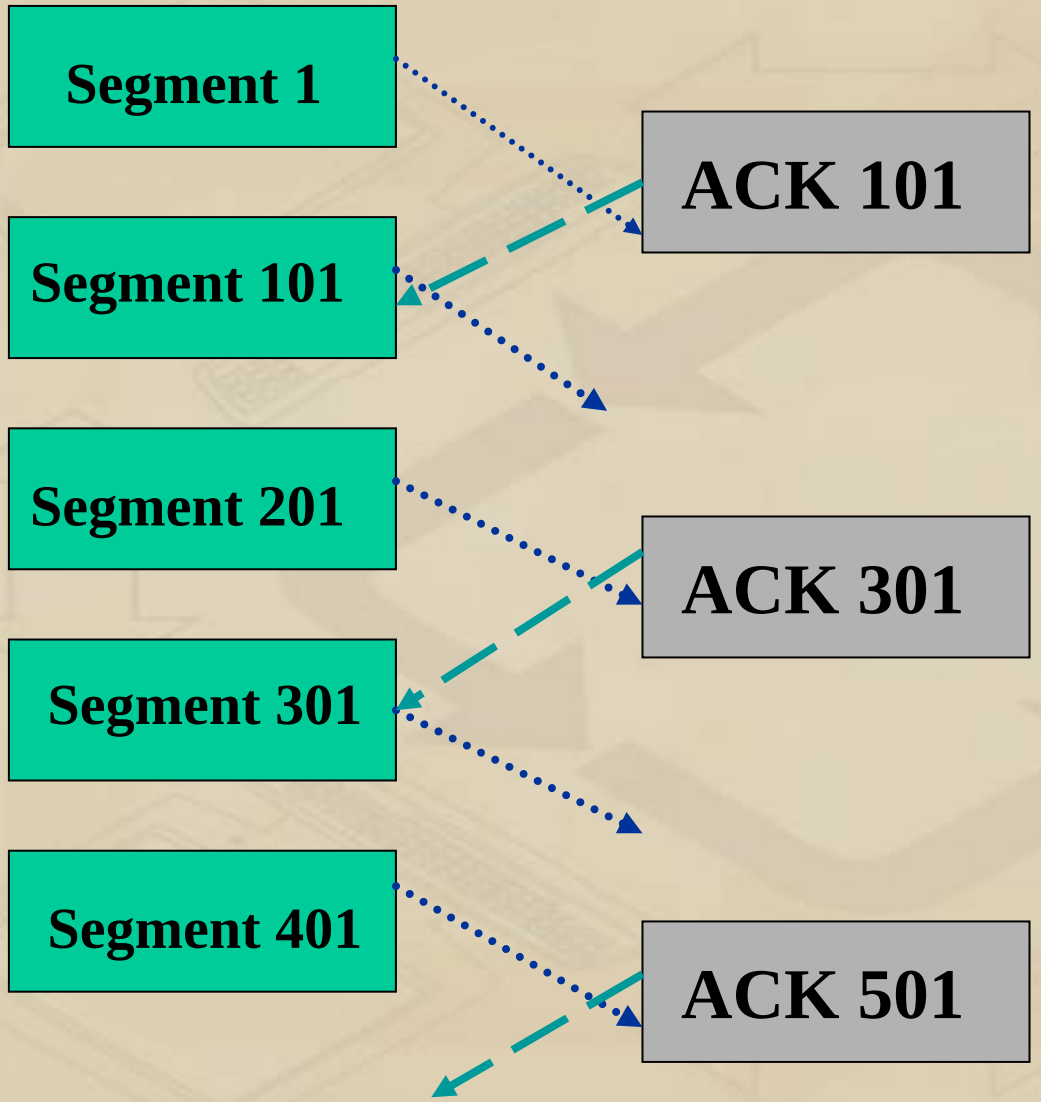
TCP may experience poor performance when multiple packets are lost from one window of data.

-- Will send out 13 packets (12 at 1,448 and 1 with remainder) before waiting for an ACK. (17520 / 1448 = 12)

Packets Sent To Fill Window

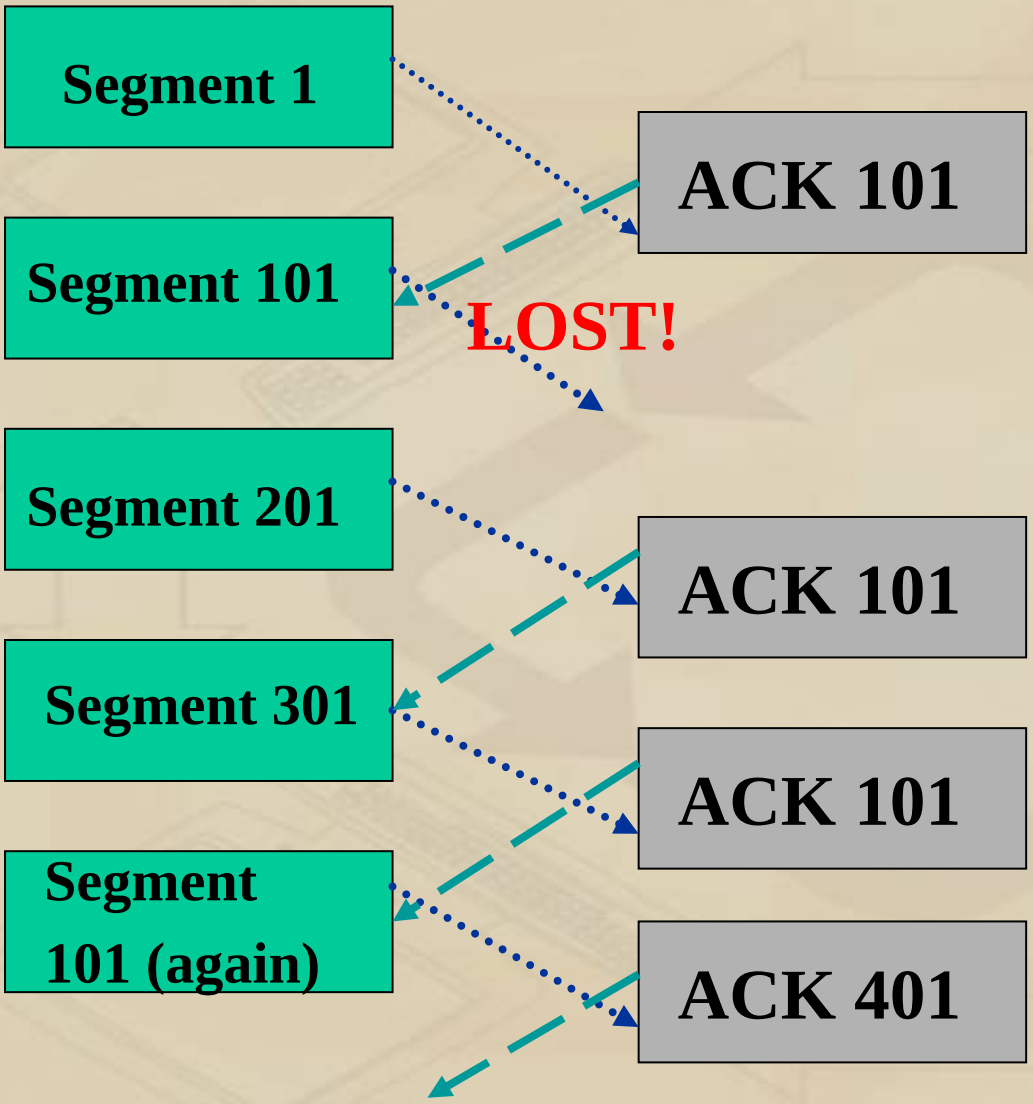


How Does ACK Work?



- *With the limited information available from cumulative acknowledgements, a TCP sender can only learn about a single lost packet per round trip time.*
- Assume each segment has 100 bytes.
- ACK is for all the bytes received and indicates the next byte of data that is expected. This is the 'cumulative ACK' in the RFC above.

Duplicate ACK - Retransmits



- Assume each segment has 100 bytes.
- ACK is for the next byte of data it is waiting for.
- A duplicate ACK is sent when a packet is received and the sequence number indicates that it does not contain the byte you are waiting for.
- After 3 duplicate ACKs, the packet is retransmitted.

Let's Look at Some Packets

Source Address	Source Port	Destination Address	Destination Port	IP ID	Data Length	Sequence	ACK	Expected ACK
10.201.0.2	23	192.168.145.7	1080	81	3	2247247727	3010274481	2247247730
192.168.145.7	1080	10.201.0.2	23	27B	3	3010274481	2247247730	3010274484
10.201.0.2	23	192.168.145.7	1080	FB	3	2247247730	3010274484	2247247733
192.168.145.7	1080	10.201.0.2	23	27C	3	3010274484	2247247733	3010274487
10.201.0.2	23	192.168.145.7	1080	100	6	2247247733	3010274487	2247247739
192.168.145.7	1080	10.201.0.2	23	27D	18	3010274487	2247247739	3010274505

- Sequence Number + Data Length = Expected ACK

Multiple Packets in a Window

Source Address	Source Port	Destination Address	Destination Port	IP ID	Data Length	Sequence	ACK	Expected ACK	Protocol
10.197.4.1	58315	10.173.12.5	4726	5BEE	1260	2167520027	2822526086	2167521287	ACK
10.197.4.1	58315	10.173.12.5	4726	5BEF	1260	2167521287	2822526086	2167522547	ACK
10.197.4.1	58315	10.173.12.5	4726	5BF0	1260	2167522547	2822526086	2167523807	ACK, PSH
10.197.4.1	58315	10.173.12.5	4726	5BF1	1260	2167523807	2822526086	2167525067	ACK
10.197.4.1	58315	10.173.12.5	4726	5BF2	1260	2167525067	2822526086	2167526327	ACK
10.197.4.1	58315	10.173.12.5	4726	5BF3	1260	2167526327	2822526086	2167527587	ACK, PSH

- Sequence Number + Data Length = Next Sequence Number (Expected ACK)

From RFC2018

- RFC2018 defines Selective Acknowledgement (SACK) as follows:

- TCP may experience poor performance when multiple packets are lost from one window of data. ***With the limited information available from cumulative acknowledgements, a TCP sender can only learn about a single lost packet per round trip time. An aggressive sender could choose to retransmit packets early, but such retransmitted segments may have already been successfully received.***

Out of Order

Source Address	Source Port	Destination Address	Destination Port	IP ID	Data Length	Sequence	ACK	Expected ACK	Protocol	Comment
10.197.4.1	58315	10.173.12.5	4726	5BF8	1260	2167531999	2822526086	2167533259	ACK	
10.173.12.5	4726	10.197.4.1	58315	5345	0	2822526086	2167533259	0	ACK	
10.197.4.1	58315	10.173.12.5	4726	5BF9	1260	2167533259	2822526086	2167534519	ACK	
10.197.4.1	58315	10.173.12.5	4726	5BFB	632	2167535779	2822526086	2167536411	ACK, PSH	TCP Out of Order. Expected Seq:2167534519 In Pkt:462

- What was expected in packet #4 is sequence # 2167534519.
- What was actually received is sequence # 2167535779
- The packet containing bytes 2167534519 through 2167535778 is lost (or not yet received).
- Since packet #4 contains a HIGHER sequence number than expected, it is noted as being out of order.

Source Address	Source Port	Destination Address	Destination Port	IP ID	Data Length	Sequence	ACK	Expected ACK	Protocol
10.197.4.1	58315	10.173.12.5	4726	5BF9	1260	2167533259	2822526086	2167534519	ACK
10.197.4.1	58315	10	Out of order segment		632	2167535779	2822526086	2167536411	ACK, PSH
10.173.12.5	4726	10.197.4	ACK for …4519		0	2822526086	2167534519	0	ACK
10.197.4.1	58315	10	Sends Next segment		1260	2167536411	2822526086	2167537671	ACK
10.173.12.5	4726	10.1	Dup ACK for …4519		0	2822526086	2167534519	0	ACK
10.197.4.1	58315	10	Sends Next segment		1260	2167537671	2822526086	2167538931	ACK
10.173.12.5	4726	10.1	Dup ACK for …4519		0	2822526086	2167534519	0	ACK
10.197.4.1	58315	10.173.12.5	4726	5BFE	1260	2167538931	2822526086	2167540191	ACK

Retransmit

Source Address	Source Port	Destination Address	Destination Port	IP ID	Data Length	Sequence	ACK	Expected ACK	Protocol
10.173.12.5	4726	10.197.4.1	58315	534A	0	2822526086	2167534519	0	ACK
10.197.4.1	58315	10.173.12.5	4726	5C01	1260	2167542711	2822526086	2167543971	ACK
10.173.12.5	4726	10.197.4.1	58315	534B	0	2822526086	2167534519	0	ACK
10.197.4.1	58315	10.173.12.5	4726	5C04	1260	2167534519	2822526086	2167535779	ACK

- More duplicate ACKs for …4519, then finally, packet …4519 is retransmitted.

So, now finally… SACK!

- (From RFC2018)

"A Selective Acknowledgment (SACK) mechanism, combined with a selective repeat retransmission policy, can help to overcome these limitations. The receiving TCP sends back SACK packets to the sender informing the sender of data that has been received. The sender can then retransmit only the missing data segments."

SACK will keep track of Out of Order

Source Address	Source Port	Destination Address	Destination Port	IP ID	Data Length	Sequence	ACK	Expected ACK	Protocol	Comment
10.197.4.1	58315	10.173.12.5	4726	5BF9	1260	2167533259	2822526086	2167534519	ACK	
10.197.4.1	58315	10.173.12.5	4726	5BFB	632	2167535779	2822526086	2167536411	ACK, PSH	TCP Out of Order. Expected Seq:2167534519 In Pkt:462
10.173.12.5	4726	10.197.4.1	58315	5346	0	2822526086	2167534519	0	ACK	SACK: SEQ: 2167535779 (8131f8a3) 2167536411 (8131fb1b)

- SACK packets will contain the sequence numbers of the out of order packets received.
- May be multiple blocks (multiple packets received out of order).

The SACK Fields are in TCP Options

- TCP Header:
- Source Port: 4726
- Destination Port: 58315
- Sequence Number: a83c5486
- Acknowledgment Number: 8131f3b7
- TCP Header Length: 32
- Flags: 0x10 (Urgent: Not set. Acknowledgement: Set. Push: Not set. Reset: Not set. Syn: Not set. Fin: Not set.)
- Window size: 64260
- Options:
  - No operation.
  - No operation.
  - SACK
    - SACK Length:8
    - SACK Block #1: 8131f8a3-8131fb1b

This is a breakout of the TCP packet showing the TCP header only.

Multiple Out of Order Packets in SACK

Source Address	Source Port	Destination Address	Destination Port	IP ID	Data Length	Sequence	ACK	Expected ACK	Protocol	Comment
10.173.12.5	4726	10.197.4.1	58315	534A	0	2822526086	2167534519	0	ACK	TCP Dup Ack. Same as packet:465 SACK: SEQ: 2167535779 (8131f8a3) 2167541451 (81320ecb)
10.197.4.1	58315	10.173.12.5	4726	5C01	1260	2167542711	2822526086	2167543971	ACK	TCP Out of Order. Expected Seq:2167541451 In Pkt:472
10.173.12.5	4726	10.197.4.1	58315	534B	0	2822526086	2167534519	0	ACK	TCP Dup Ack. Same as packet:465 SACK: SEQ: 2167542711 (813213b7) 2167543971 (813218a3), 2167535779 (8131f8a3) 2167541451 (81320ecb)

From RFC2581

- RFC2581 is TCP Congestion Control
- FLIGHT SIZE: The amount of data that has been sent but not yet acknowledged.
- Why does this matter?

Example #1

Source Address	Source Port	Destination Address	Destination Port	IP ID	Data Length	Sequence	ACK	Expected ACK	Delta (ss.milli.micro)	Bytes In Flight Sender	Bytes Acked This Packet
208.111.39.67	5902	24.6.68.48	1315	5A3F	4380	3899888273	1990669247	3899892653	0.001.852	4,380	0
208.111.39.67	5902	24.6.68.48	1315	5A42	606	3899892653	1990669247	3899893259	0.000.007	4,986	0
208.111.39.67	5902	24.6.68.48	1315	5A43	8760	3899893259	1990669247	3899902019	0.003.140	13,746	0
24.6.68.48	1315	208.111.39.67	5902	8782	0	1990669247	3899902019	0	0.027.942	0	13,746

- Sender sent packets with bytes 4,380, plus 606 more, and then 8,760 more without waiting very long.
- Then, to get the ACK from the receiver, 27 milliseconds elapsed.

- The idea is to keep the pipe filled.

Example #2

Source Address	Source Port	Destination Address	Destination Port	IP ID	Data Length	Sequence	ACK	Expected ACK	Delta (ss.milli.micro)	Bytes In Flight Sender
10.197.4.1	58315	10.173.12.5	4726	5BEE	1260	2167520027	2822526086	2167521287	0.004.378	1,260
10.197.4.1	58315	10.173.12.5	4726	5BEF	1260	2167521287	2822526086	2167522547	0.000.057	2,520
10.173.12.5	4726	10.197.4.1	58315	5340	0	2822526086	2167522547	0	0.000.031	0
10.197.4.1	58315	10.173.12.5	4726	5BF0	1260	2167522547	2822526086	2167523807	0.000.035	1,260
10.197.4.1	58315	10.173.12.5	4726	5BF1	1260	2167523807	2822526086	2167525067	0.000.579	2,520
10.173.12.5	4726	10.197.4.1	58315	5341	0	2822526086	2167525067	0	0.000.033	0

- In this example, the ACKs are coming very quickly, so even though the bytes in flight are low, there does not appear to be a lot of waiting.

Throughput / BIF

Source Address	Source Port	Connection Time (ss.milli.micro)	Total Packets	Throughput Packets Per Second	Total Data Bytes	Throughput Data Bytes Per Second	Minimum Bytes In Flight	Average Bytes In Flight	Maximum Bytes In Flight
10.19.0.200	21	11.807.996	25 (1.52%)	2	922 (0.1%)	83	0	38	186
10.19.0.200	38571	0.081.236	6 (0.36%)	6	3K (0.44%)	3,812	0	1K	2K
10.19.0.200	56783	0.119.630	473 (28.91%)	473	551K (64.58%)	551,369	0	1K	2K

- We like to look at throughput as well as bytes in flight.
- Complicated area!

Complications

- How fast can you go?
- How congested is the network?

- How fast are the applications?
  - What happens in case of problems?
- How fast can the client react?
  - How many ACKs?

- These all change!

Compare IPv6 and IPv4

Destination Address	IP Protocol	Connection Time (ss.milli.micro)	Total Packets	Throughput Packets Per Second	Total Data Bytes	Throughput Data Bytes Per Second	Minimum Bytes In Flight	Average Bytes In Flight	Maximum Bytes In Flight
208.111.39.67	IPv4	35.327.127	2K (52.01%)	65	11K (0.57%)	324	0	7	28
208.111.39.67	IPv4	0.020.015	5 (0.11%)	5	480 (0.02%)	480	0	96	480
208.111.39.67	IPv4	0.019.998	5 (0.11%)	5	480 (0.02%)	480	0	96	480
208.111.39.67	IPv4	5.371.834	8 (0.18%)	1	6K (0.31%)	1,218	0	761	1K
2607:F740::3F:216:3EFF:FE68:72C0	IPv6	20.940.750	11 (0.25%)	< 0	9K (0.5%)	495	0	908	8K
2607:F740::3F:216:3EFF:FE68:72C0	IPv6	20.792.770	8 (0.18%)	< 0	9K (0.48%)	474	0	1K	8K
2607:F740::3F:216:3EFF:FE68:72C0	IPv6	20.797.733	9 (0.2%)	< 0	1K (0.06%)	60	0	139	579
2607:F740::3F:216:3EFF:FE68:72C0	IPv6	20.795.714	8 (0.18%)	< 0	1K (0.09%)	90	0	234	1K
2607:F740::3F:216:3EFF:FE68:72C0	IPv6	20.793.697	8 (0.18%)	< 0	1K (0.06%)	60	0	156	579
2607:F740::3F:216:3EFF:FE68:72C0	IPv6	20.741.833	8 (0.18%)	< 0	9K (0.5%)	497	0	1K	8K
2607:F740::3F:216:3EFF:FE68:72C0	IPv6	6.440.179	76 (1.73%)	12	42K (2.15%)	7,026	0	1K	7K
2607:F740::3F:216:3EFF:FE68:72C0	IPv6	6.108.390	69 (1.57%)	11	2K (0.12%)	419	0	47	1K

- As integration of IPv6 happens, it is interesting to compare both protocols in terms of throughput and bytes in flight.

Let us help you!

- Inside Products specializes in TCP/IP networks.
- We have many years of expertise on many very large networks.

TCP, EE, SSL, IPv6 Problem Finders

The products for the serious diagnostician :
the Problem Finders allow you to:

- Quickly find problems in diagnostic traces - which can consist of thousands or hundreds of thousands of packets!
- Get automatic recommendations and analysis

- **TCP Problem Finder** turns a trace into English!

–We expect that the amount of time saved is a factor of 8. That is, a trace which might take 8 hours to do manually, can be done in 1 hour with our **Problem Finders.**

Duplicate Acknowledgements

- Duplicate acknowledgements (acks) were found on this session from 10.173.12.5 port:4726.
- The number of duplicate acks received is: 195. The total packets received is: 220. The percent of packets which are duplicate acks is:88.63%.
- Duplicate acks may indicate one of the following conditions:
  - Packet loss,
  - Retransmits,
  - An error indicator, for example from a printer which is out of paper, or
  - A keep-alive indicator.
- Too many duplicate acknowledgments may indicate a situation which needs to be further investigated.
- No duplicate acknowledgements from 10.197.4.1 port:58315 were found in the data flow.

Delayed Acknowledgements

- No delayed acknowledgements from 10.173.12.5 port:4726 were found in the data flow.
- No delayed acknowledgements from 10.197.4.1 port:58315 were found in the data flow.

Out of Order Segments

- No out of order packets from 10.173.12.5 port:4726 were found in the data flow.
- Out of order packets were found on this session from 10.197.4.1 port:58315.
- The number of out of order packets received is: 9. The total packets received is: 232. The p
  packets which are out of order is:3.87%.
- The number of bytes received in out of order packets is: 9,452. The total bytes received is:
  The percent of bytes received in out of order packets is:3.28%.
- Packet 464 was out of order.
- Packet 474 was out of order.
- Packet 507 was out of order.
- Packet 515 was out of order.
- Packet 519 was out of order.
- Packet 537 was out of order.
- Packet 552 was out of order.
- Packet 556 was out of order.
- Packet 928 was out of order.

Selective ACKs

- Selective ACKs were found on this session from 10.173.12.5 port:4726.
- The number of selective ACK packets received is: 200. The total packets received is: 220. The percent of packets which are selective ACKs is:90.9%.
- Packet 465 was a selective ACK.
- Packet 467 was a selective ACK.
- Packet 469 was a selective ACK.
- Packet 471 was a selective ACK

IP Problem Finder will tell you of problems with SACK, out of order, dup acks, and much more!

TCP Classes

- EE Trace Analysis
- Security, IPSec, Policy Agent
- IPv6 (Addressing, Multi-platform)
- TCP Tuning and Performance Analysis
- Trace Analysis and Diagnostics

Contact Us!

- For more information:
  - Products,
  - Consulting,
  - TCP/IP classes

1-831-659-8360

sales@insidethestack.com

- Pricing for TCP Problem Finder:
  - $25K per site per year.
  - Like buying a router!

Australia: Blueline Software

UK : FitzSoftware

BENELUX: Adinsec BvBa

Israel : NESS

France : Query Informatique