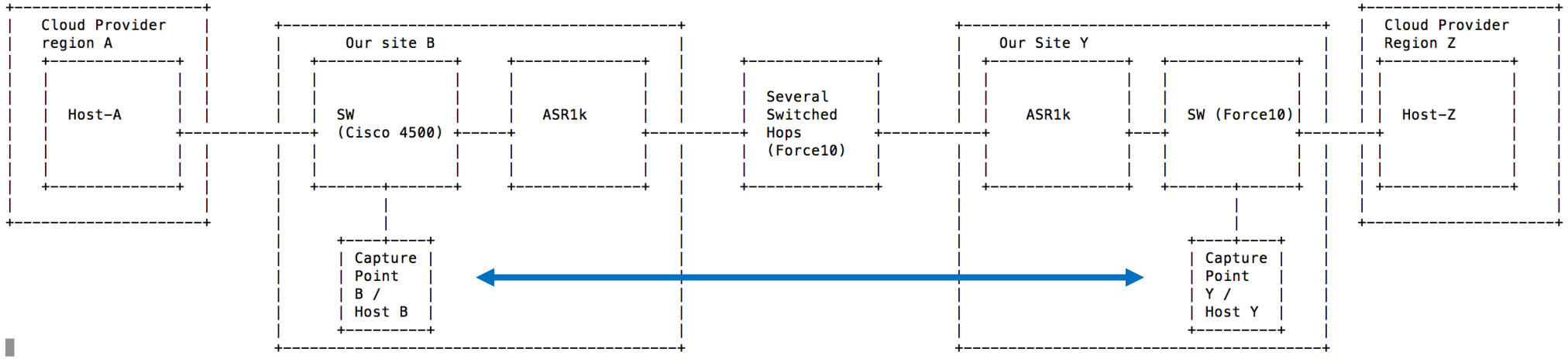
This is a response to a question asked on Tuesday 5th December by “u/thegreattriscuit” in Reddit’s “r/Networking” subreddit.
https://www.reddit.com/r/networking/comments/7hqsqm/im_at_a_loss_here_tcp_performance_issues_across/

The problem was slow file transfer throughput across a long 1 Gbps WAN (“across an ocean”) and caused by packet losses. However, we’re told that the packet loss behaviour was consistent and readily reproducible for this particular iPerf test - but not apparent for other transfers across the same link.

The provided “topology” diagram is below. The flow is from Host-A at the left to Host-Z on the right and the packet losses occur somewhere between the two points (as indicated by the blue line between two capture points). The losses could be within the Cisco 4500, ASR1000, Force10 WAN, ASR1000 or Force10 switch).



A commercial packet analysis tool called NetData Pro was used to perform this analysis.

Summary of Findings

- This is a very good example of relatively small packet losses having a major impact on throughput (due to the TCP “Congestion Avoidance” mechanism).
- The losses always occur at the end of larger packet bursts – which makes the packet loss behaviour look very much like a queueing and buffer overflow in a device along the path (which typically would occur at a point where link speeds decrease, e.g., 1 Gbps down to 250 Mbps).
- However, any queueing would introduce variable network transit times (and round trip times) above 1 ms for packets forced to spend time in a queue.
- No such large variations in trip times are found. The analysis modelled various link speed change values but could find none that matched the observed trip times. All packets appear to traverse the WAN with very little delay (transit times vary by just 0.85 ms).
- A modelling of packet shaping/policing does fit the observed behaviours and so is the most likely cause. The observed small increases in transit times from B to Y occur at times when packets are lost and appear to correlate closely with a small “shaping” queue (perhaps just 80 KB).

Conclusions

The usual simple suspects for slow transfer speeds, such as low bandwidth, not using TCP Window Scaling, small Receive Windows, network congestion, etc. can all be easily eliminated.

If the losses aren’t caused by queue buffer overflows in an intermediate device, some form of shaping/policing is the most likely candidate. As the ASR1000’s are under the control of the orginal poster, the recommendation would be to examine those first. Next would be to look for shaping/policing somewhere in the Force10 WAN.

It is possible, of course, that shaping is done in one device and policing in another device.

The provided PCAP files only contain packets for the one tested flow, so we can’t measure any effects of other traffic competing for the network at any other point in the path. However, the reproducibility of the test results suggests that “random” factors such as that aren’t playing a part.

The modelling done within the tool suggests a shaping/policing PIR of 280 Mbps but with a small buffer (bucket) of 80 KB and a quite low CIR of just 10 Mbps.

There are many assumptions being made by the modeller and so it would be very helpful if the Reddit poster could find any shaping/policing policies and let us know the values. If these settings are somewhere within the Force10 WAN, then these might be more difficult to find.

The Analysis

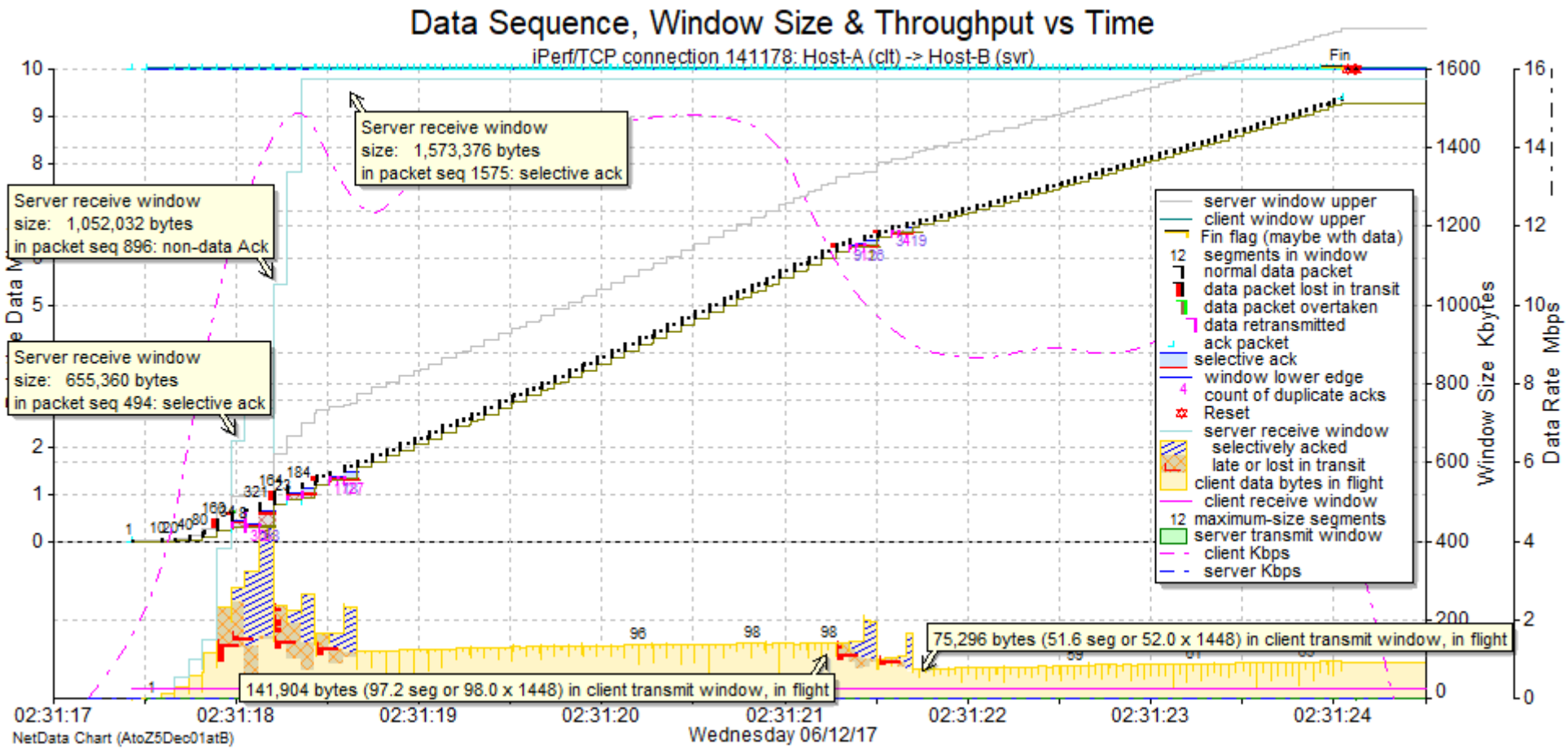
Various detailed charts are provided below, each showing the evidence that supports the conclusions.

The Result (in January, 2018)

The analysis had correctly identified the cause of the packet losses.
The original Reddit poster found policing settings of 300 Mbps with a bucket size of 50 KB somewhere in the Force10 WAN.

Adjusting those settings - and eventually removing them – significantly improved the data throughput back to expected norms.

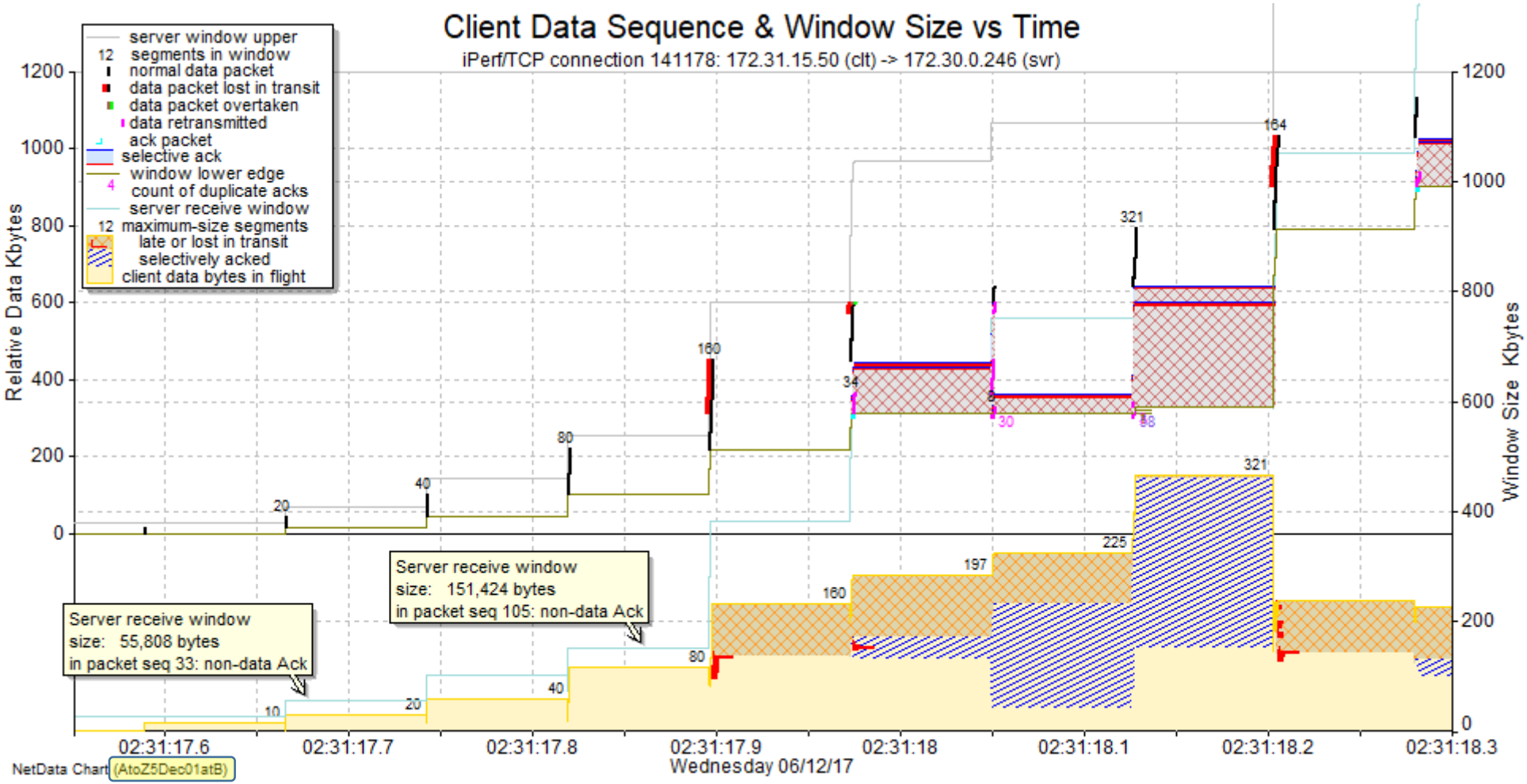
The response from the original Reddit poster is included at the end of the analysis.



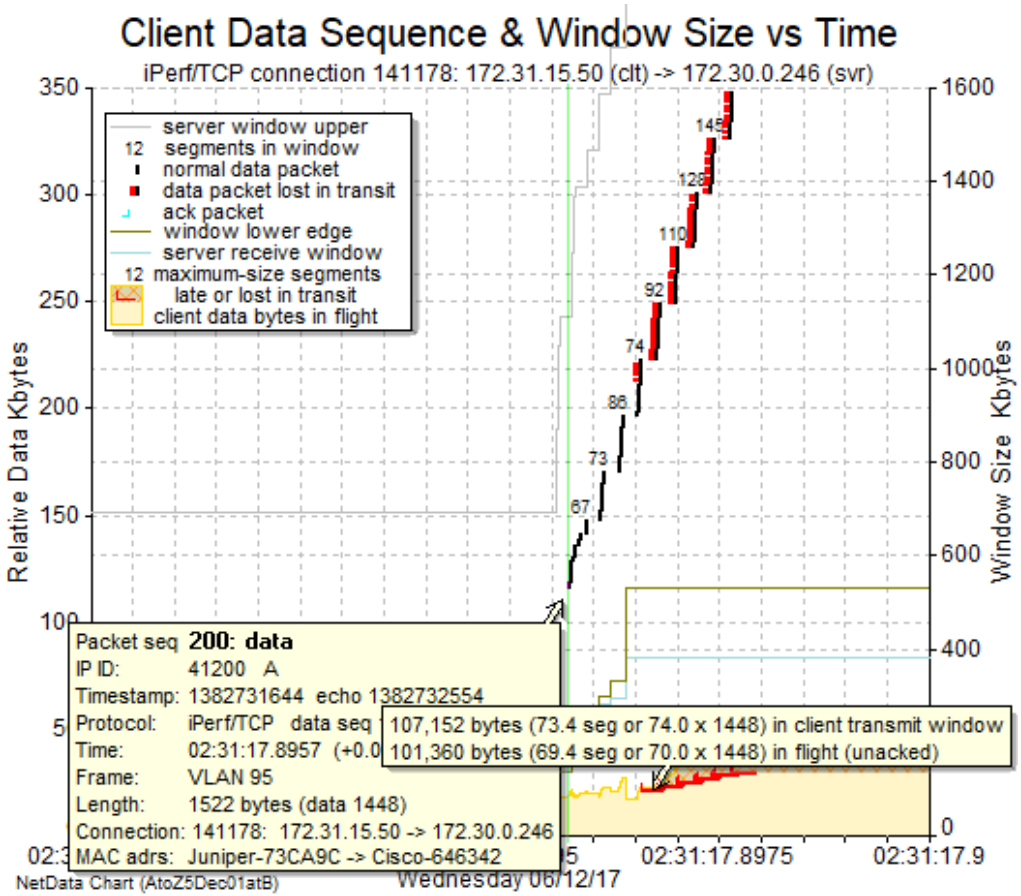
This chart of the whole transfer has many TCP protocol attributes overlaid and correlated with the packets. We see that the server increases its Receive Window (light blue line) well ahead of any packets sent (and Window Scaling is being used). Particularly interesting here is the “in flight” packet data plotted in the bottom section in yellow.

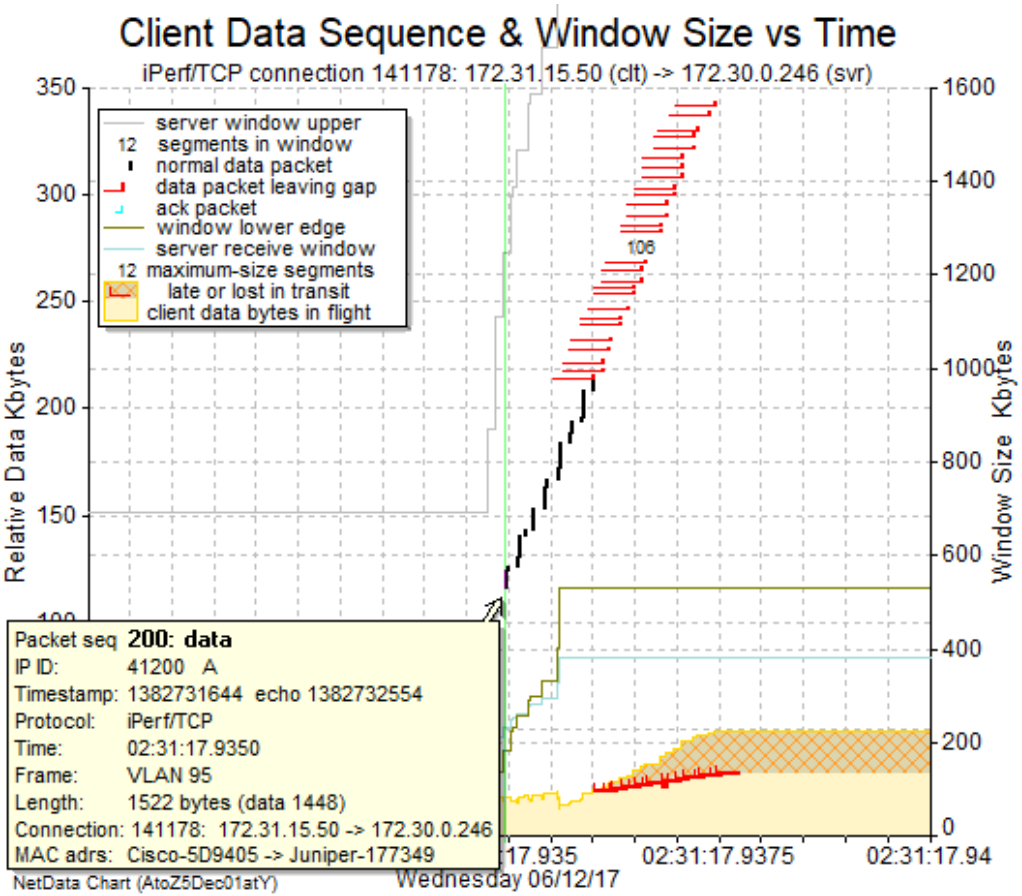
We see the effect of the “Congestion Avoidance” mode of TCP. The sender’s “Slow Start” was 10, 20, 40, 80 then 160 packets per trip but packet losses caused it to halve its transmissions to 84 packets per trip, slowly ramp up to 98 then “halve” again to 52 after more losses, ending the flow with 66 in-flight packets.

The red markers on the congestion-window graph at the bottom of the chart indicate the amount of in-flight data when packets were lost. It looks like losses occur only for larger bursts.



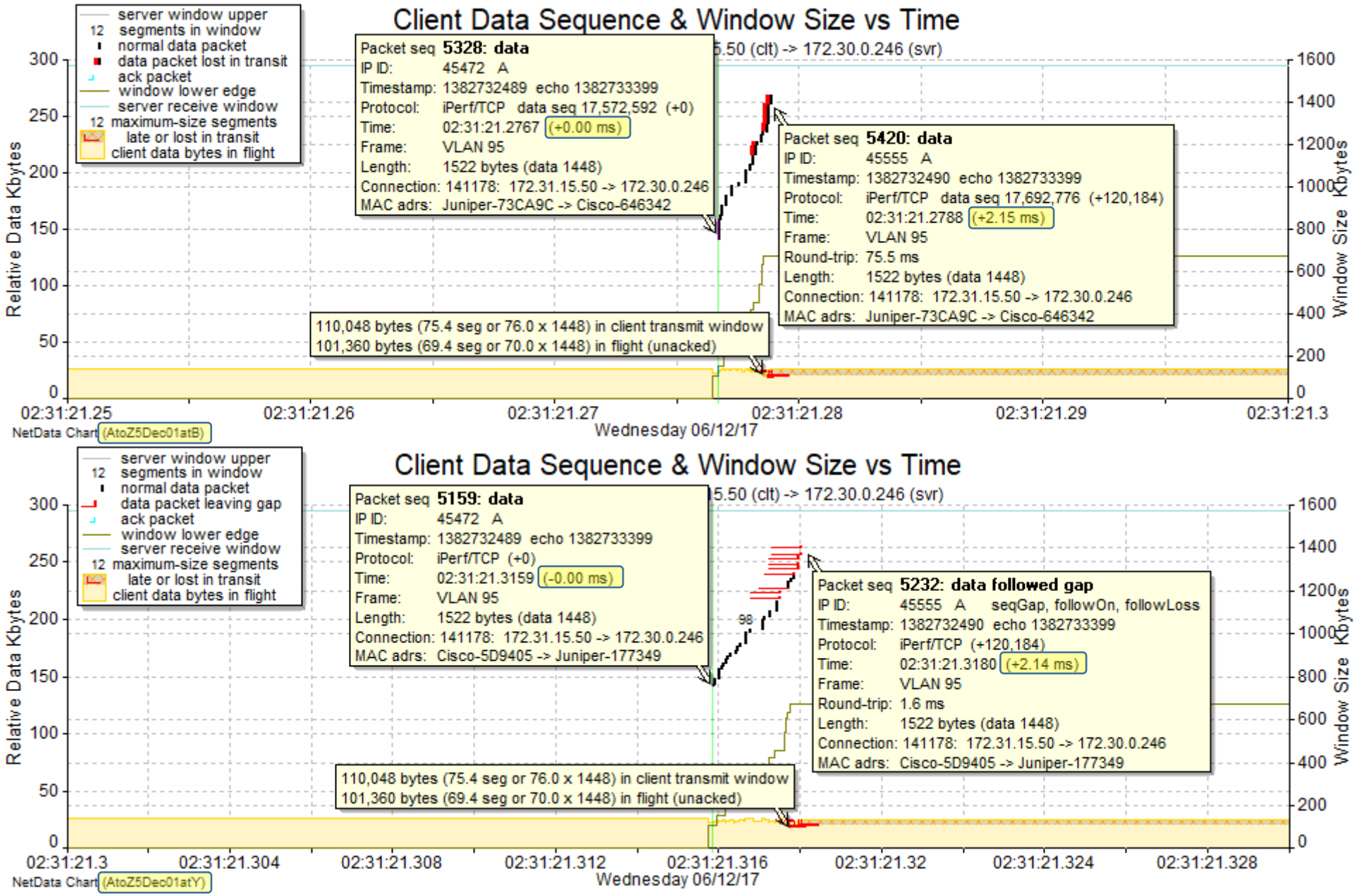
Zoomed-in to the start of the flow as seen at “point B” (note filename at bottom left), we see that packets were first lost in the burst of 160 segments that followed the successful “Slow Start” of 10, 20, 40 and 80 segments per trip (“in flight”). Red markers are attached to the black packet markers when we know (because of ACKs and SACKs) that those packets were subsequently lost. The initial packets in each burst were successful, with losses always occurring some way into each burst. The server’s Receive Window is always larger than the client’s transmission size.



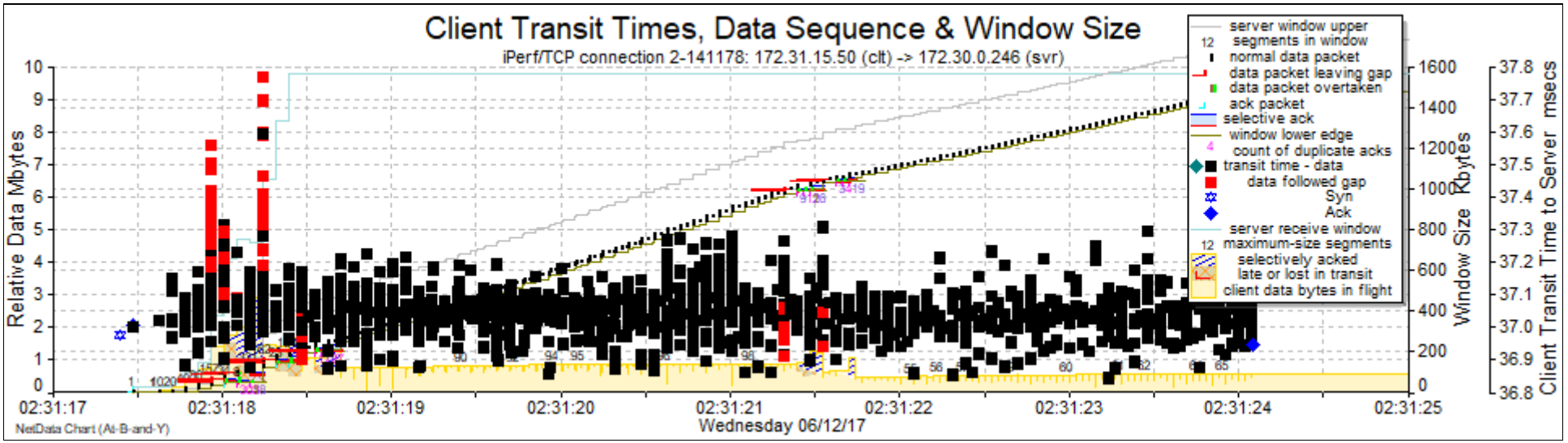


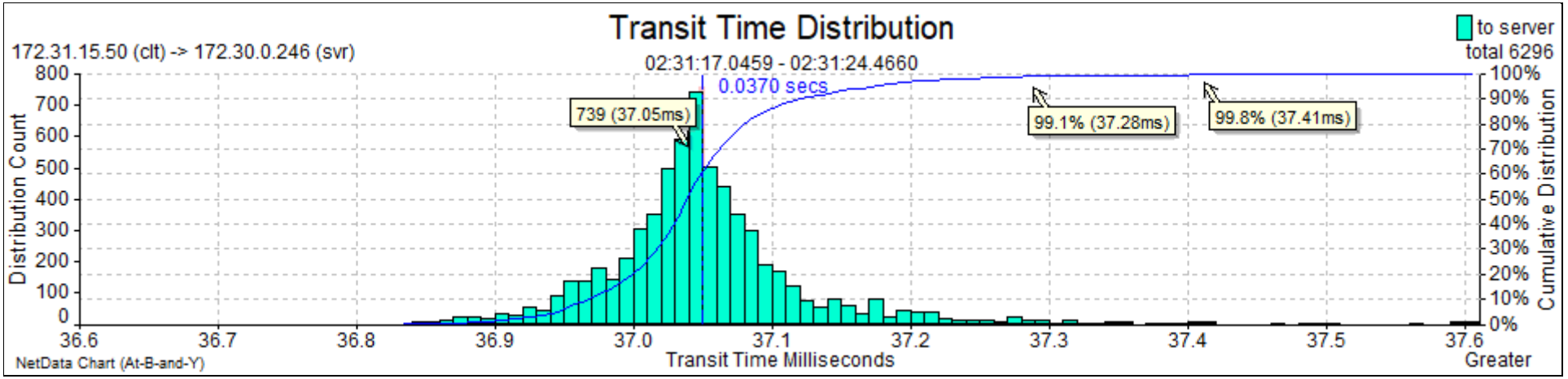
Further zoomed-in to the single burst of 160 segments we observe the first packets were lost when only 70 segments were in flight. Even though we saw the previous burst of 80 segments go through entirely successfully.

The left hand chart is “At B” and the right hand chart is the same 10 ms period “At Y”. Note that the time difference from the first to last packet in the “At B” burst is 1.95 ms whereas “At Y” it is 2.49 ms. The later packets do take half a millisecond longer to traverse the WAN (artefact of “shaping” perhaps?)



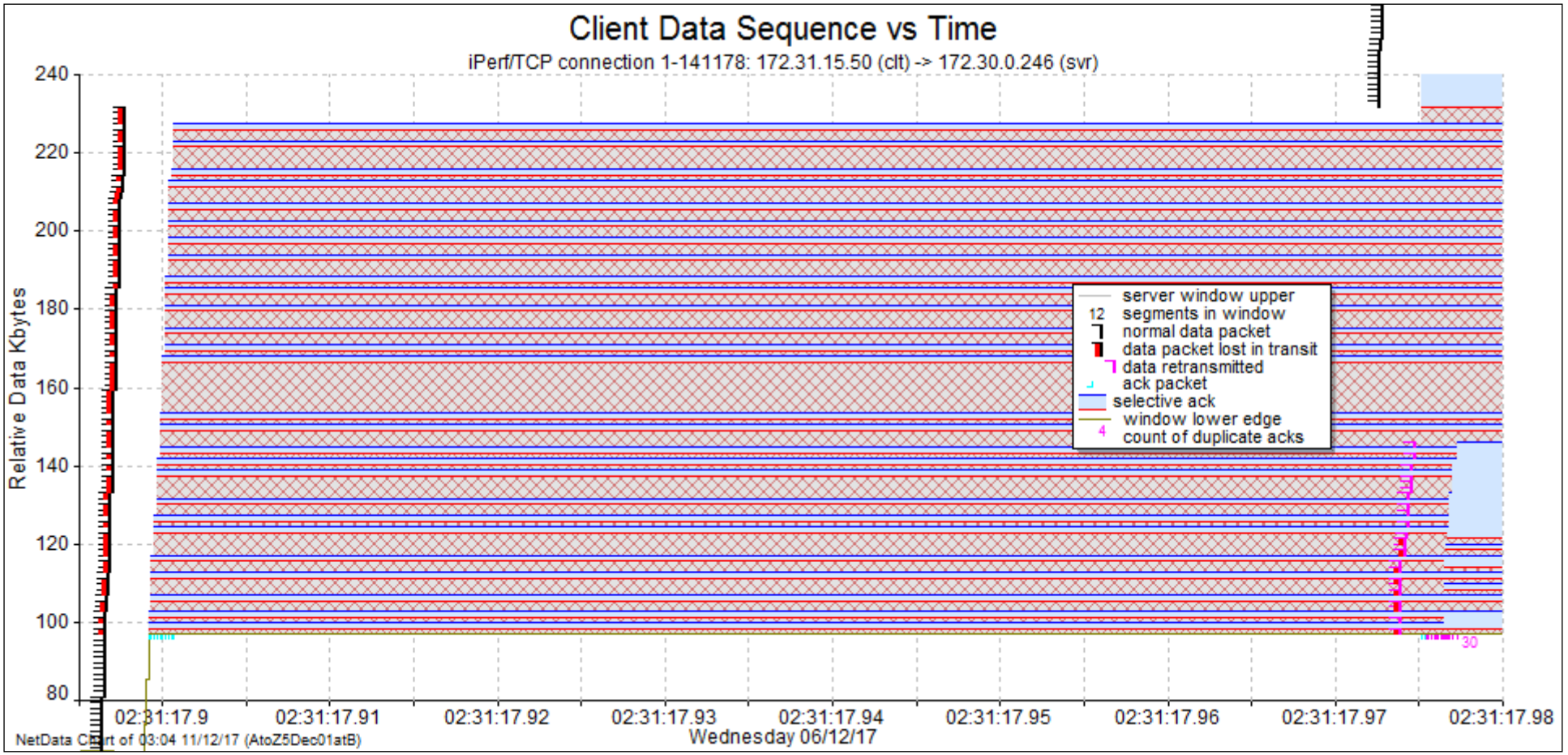
Moving forward in time to the losses in the middle of the flow (after “congestion avoidance” has ramped up to 98 segments per round trip), the next burst also lost its first packet when 70 segments were in flight. Note that the last packet is not delayed as before (2.15 ms and 2.14 ms). Top=”At B”, bottom=”At Y”.





On the top chart, the black squares represent the measured trip times for each packet as measured by comparing the “At B” and “At Y” after adjusting the time stamps. The scale is on the RHS and we see that the variation is mostly between 36.85 and 37.30 ms (0.45 ms). The red squares are packets that followed a gap and it is only for those that we see more significant delays of an additional 0.4 ms (as if the packet loss mechanism also involves a small amount of queueing).

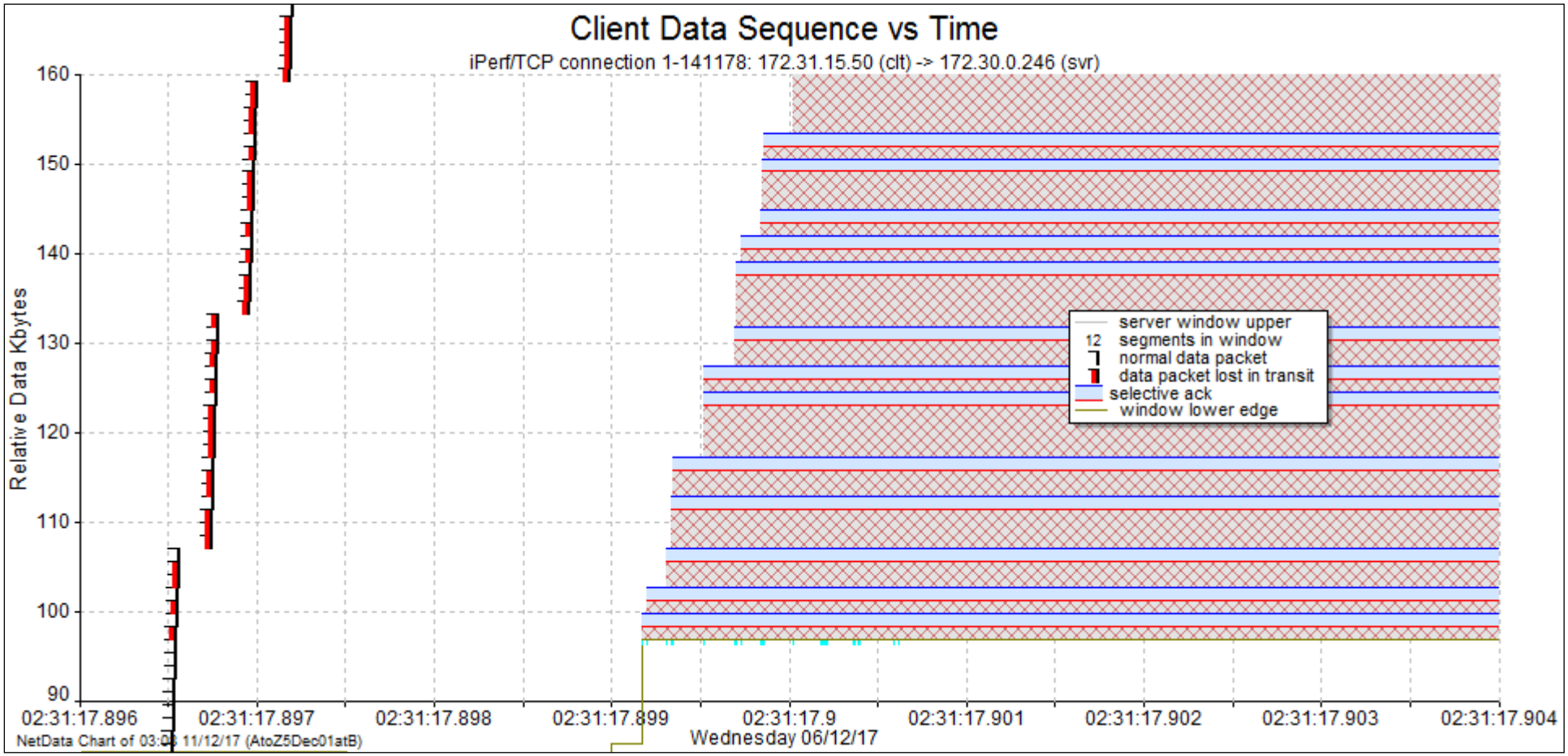
The bottom chart is a frequency distribution histogram of all those transit times. The very small variation in trip times rules out the possibility of any significant queuing delays in the network (such as a transition between links with different speeds) but could be a hint of small scale queuing due to packet shaping.



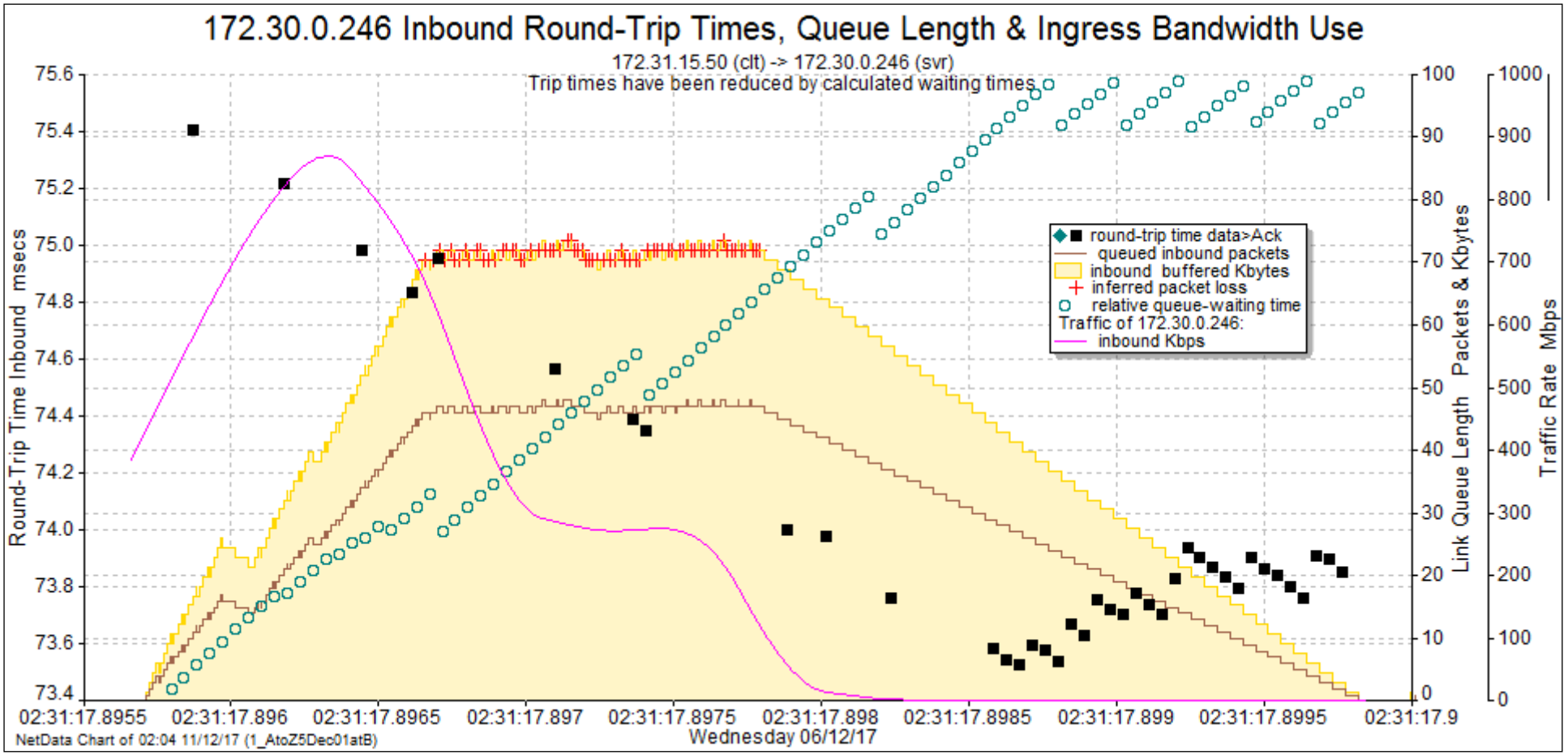
The first burst that lost packets. Once the losses began, roughly 75% of subsequent packets were lost.

On this chart the Selective Ack (SACK) markings are shifted left by 73 ms in order to display them closer to the lost packets. The light blue areas represent packets that have been (selectively) acknowledged and the cross-hatched areas represent packets that were (selectively) not acknowledged. These exactly match the black data packets marked with red.

Observe the purple retransmitted packets at the right, filling in the cross-hatched areas. Many of those retransmissions were again lost in transit after this capture point (“At B”). They have red attached and we see the increase in light blue area for those that weren’t lost.

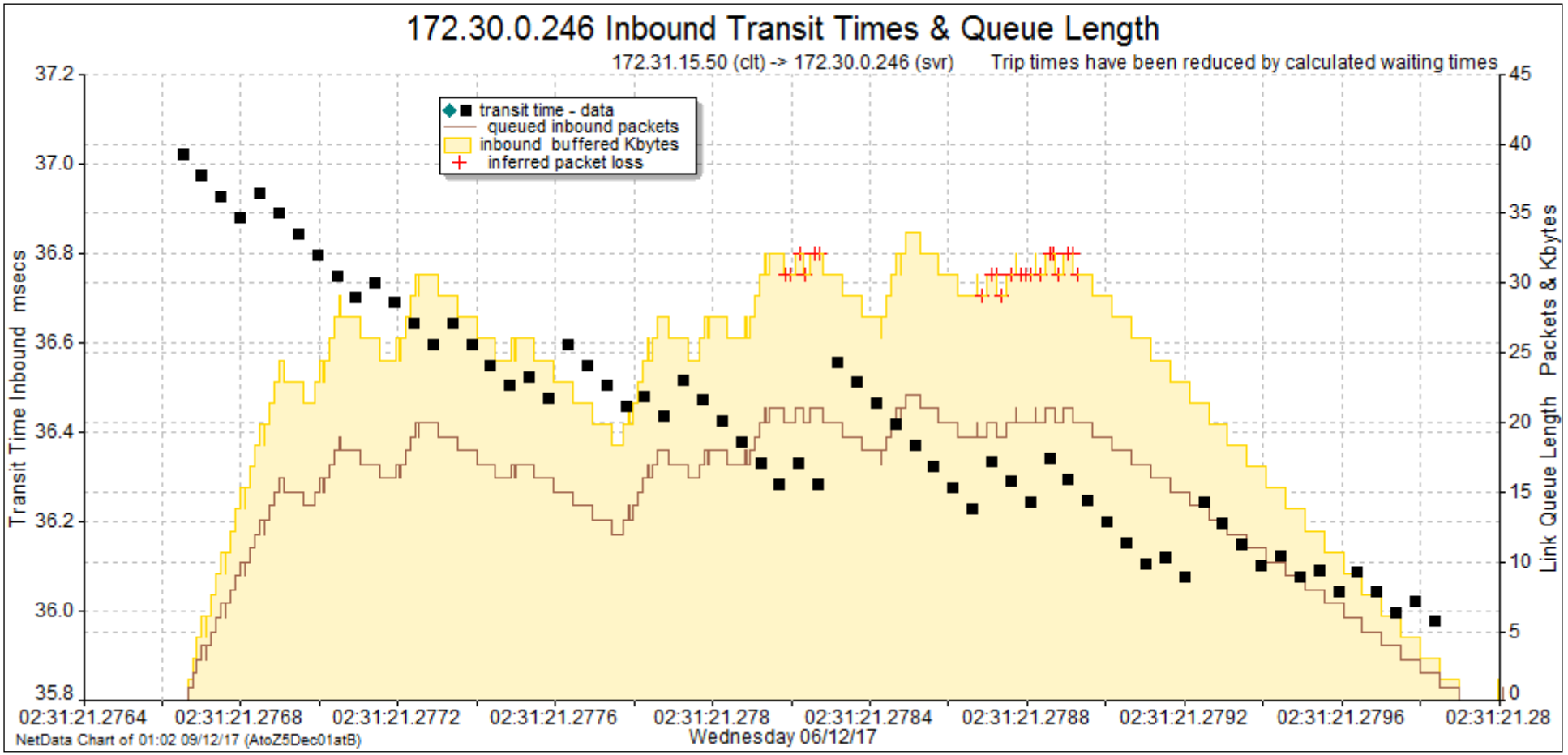


Here we zoom-in to the beginning of the lost packets so that the ratio and positioning of the lost/not-lost is more visible. Given that the ratio is around 3/1, we might expect any queueing or shaping to be taking our 1 Gbps down to somewhere around 250 Mbps.



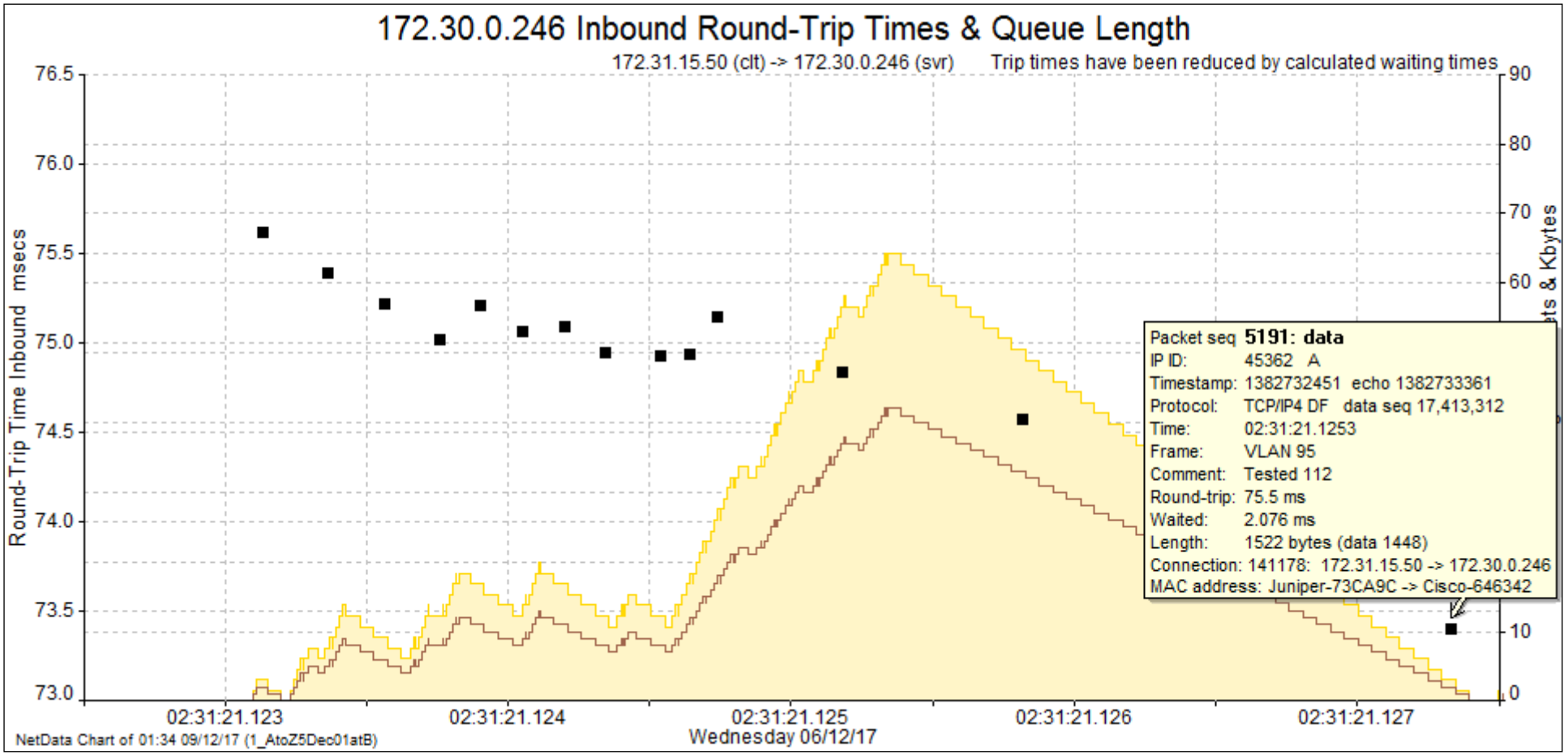
A model of the packet queue between a 1 Gbps link and a 280 Mbps link formed by the first packet burst that suffered packet loss indicated that packets would have been dropped when the queue had 45 packets and occupied 70 KB (RHS scale, red and yellow lines).

However, this model calculates waiting times up to 2.2 ms (LHS scale, height of circle from bottom of chart). Such variations in transit time were not observed in the capture files.

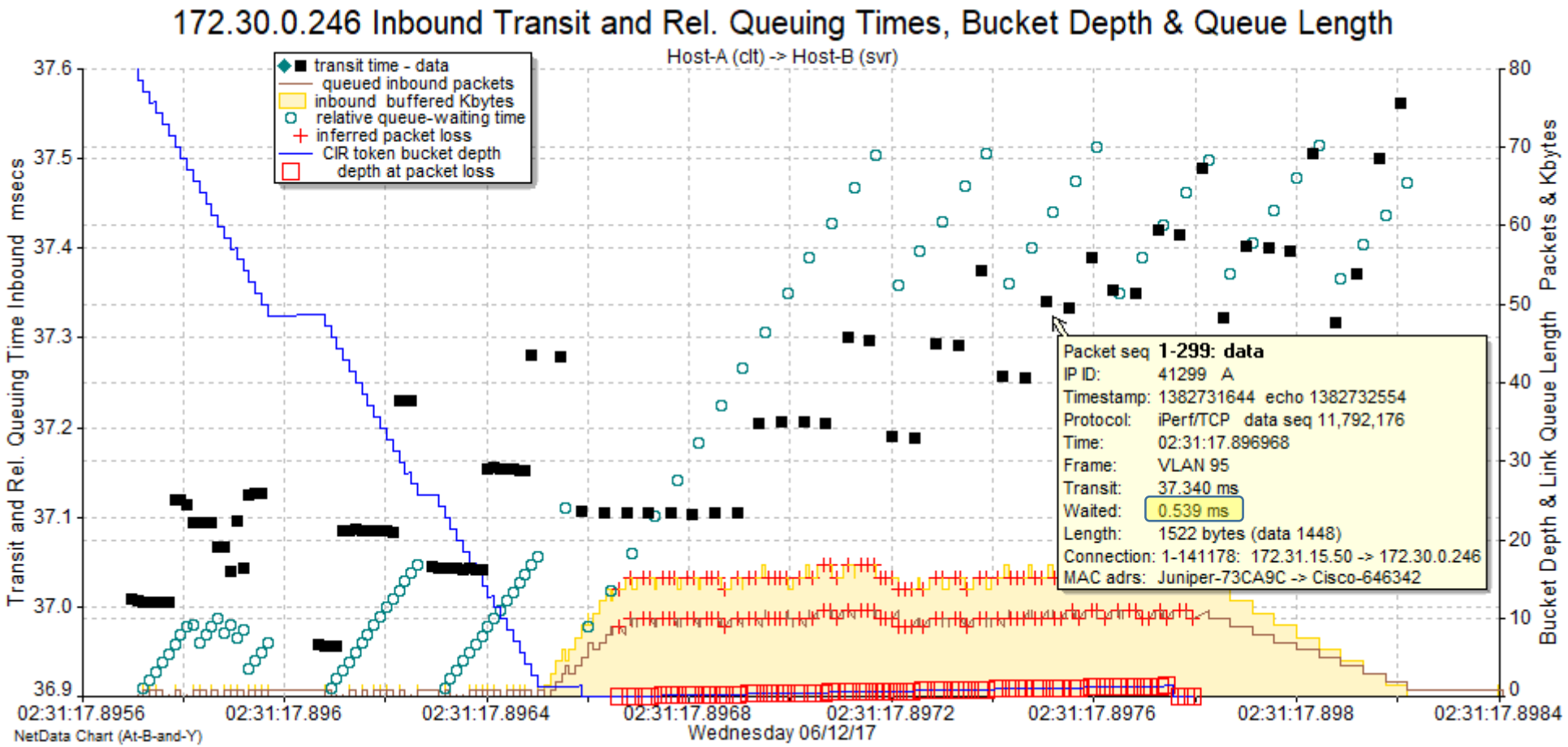


In another model, if packets arrived in a router at 1 Gbps and queued for egress at 250 Mbps, then the lost packets would have been discarded when the queue held 20 packets (30 KB), but some successful packets would have to wait for one millisecond.

Since we observe no increased RTTs above one millisecond, this form of queueing cannot be the source of dropped packets.

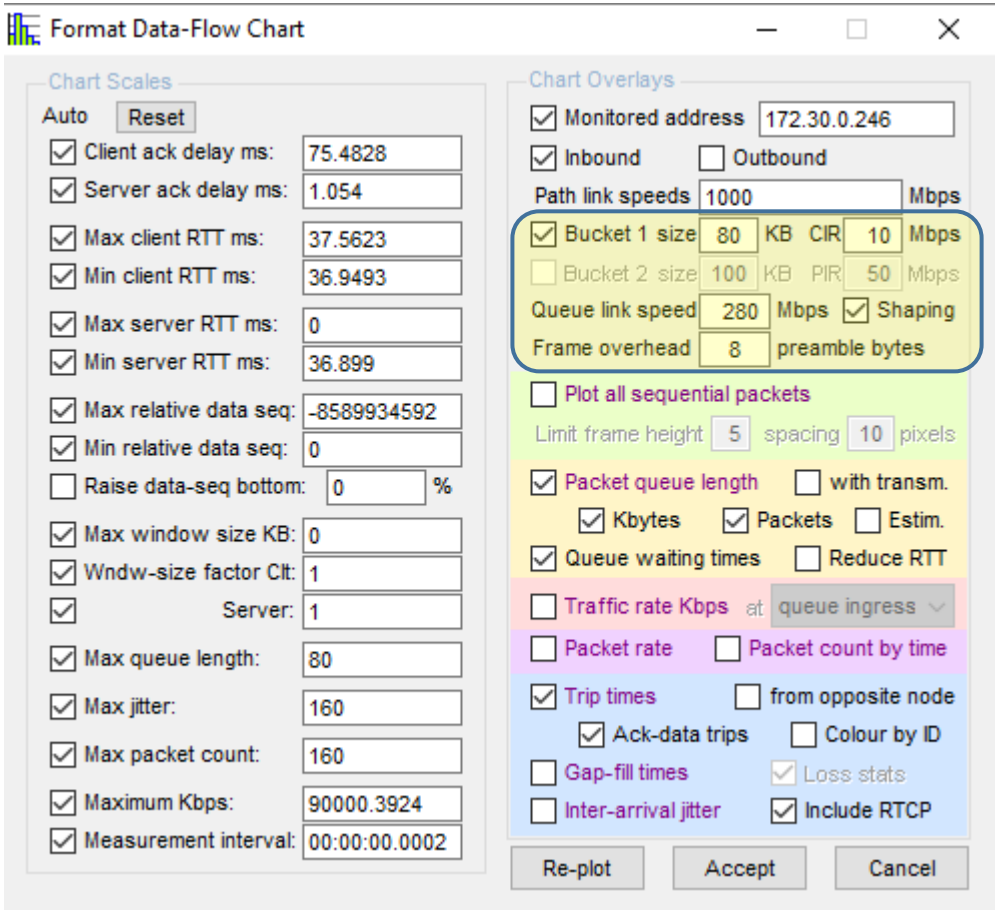


The same queuing model indicates that successful bursts would survive a queue with 40 packets but some packets would wait for more than 2 ms.

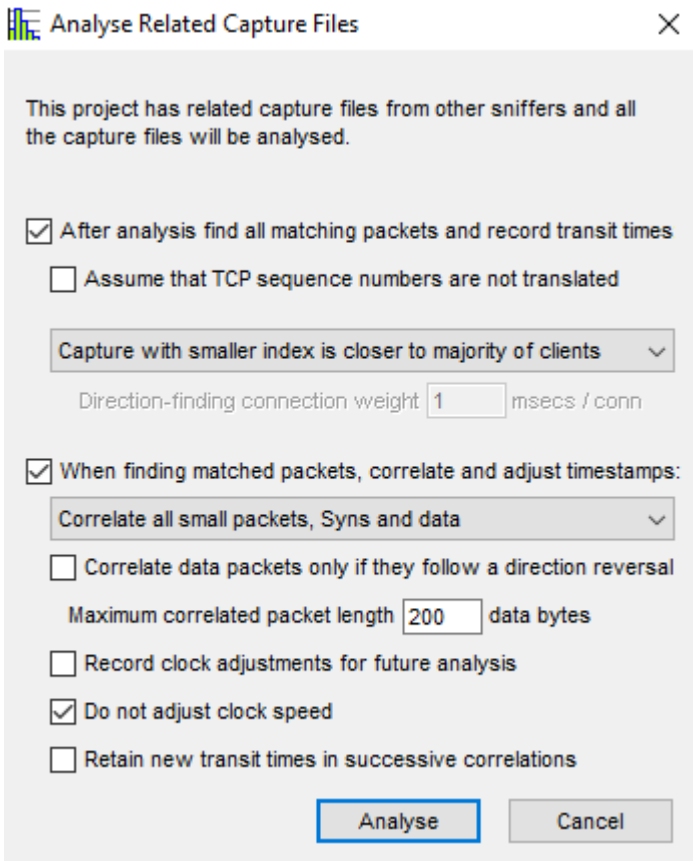


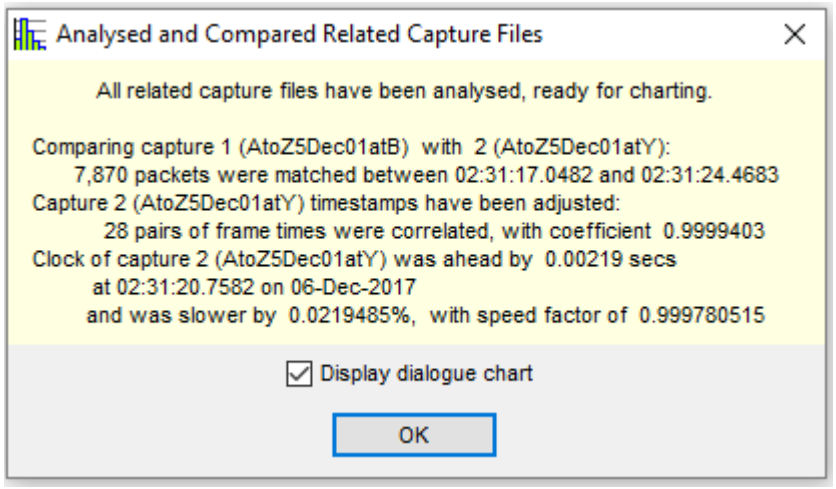
In order to create a chart that produced a viable pattern of behaviours to match the packet capture data, the following parameter values were used:

- Shaping “bucket” size of 80 KB.
- CIR of 10 Mbps.
- Shaped speed of 280 Mbps.



The shaping/policing parameters specified in the final model are highlighted above (this is NetData’s “Format” control window for the packet flow chart).





These are the dialog boxes from within the tool showing how the packets from both captures were loaded, compared and adjusted.

Final Outcome - Response from “r/thegreattriscuit”

(Slightly edited)

Yeah, you guys had it right.

On one of the "several switched hops (Force10)" there was this:

`rate police 300 peak 300`

I'd seen it before, but was doin' the 'ole "thinking about it without thinking about it" thing. "Well, we're not getting anywhere near 300mbps, so surely this isn't the problem", without considering what time interval it might be using to make those decisions.

Part of this is that, for the life of me, I can't find a counter that gets incremented on the Force10 when it drops these packets.

But when you came up with "looks like it's being policed to 280mbps", well... yeah, caused me to take a closer look.

Here's the best part:

```
HOSTNAME(conf-if-te-0/27)#rate police 300 peak 300 ?
<16-200000>             Burst size kilobytes (default = 50)
vlan                    Specify the VLAN(s)
<cr>
```

So the bucket size was 50KB.

When we introduced shapers to smooth things out, throughput jumped right up. Likewise, increasing the burst size on the policer worked fine as well. Ultimately the policer was deemed to be a relic from an obsolete configuration and so we got rid of it entirely.

As for the "crazy" stuff that didn't make sense to me (A to Z impacted more than B to Y, etc...) that all comes down to how bursty the traffic was. Coming from the cloud provider’s beefy (likely 100G) internal network those packets were coming in at nearly exactly line-rate, so they hit that policer hard. Stuff coming off our chintsy linux boxes were spread out a bit more, and so didn't get punished as hard by the policer, and enjoyed higher throughput.

I'm still impressed with how close you got to the correct parameters of the policer. Especially when you consider there's additional overhead on the flows where the policer was applied, you were basically spot on.

Links, References, Supporting Information

Original Reddit post:
https://www.reddit.com/r/networking/comments/7hqsqm/im_at_a_loss_here_tcp_performance_issues_across/

This response to the question within Reddit:
https://www.reddit.com/r/networking/comments/7hqsqm/im_at_a_loss_here_tcp_performance_issues_across/drnkxhy/

Original poster’s final outcome:
https://www.reddit.com/r/networking/comments/7hqsqm/im_at_a_loss_here_tcp_performance_issues_across/dt8ilc7/

Original packet capture files:
https://www.reddit.com/r/networking/comments/7hqsqm/im_at_a_loss_here_tcp_performance_issues_across/dqt6bj6/

AtoZatY: https://packettotal.com/app/analysis?id=61220c94d75e219f064e5e0f83b7f1a9
AtoZatB: https://packettotal.com/app/analysis?id=35c2310d44275f986213f4ba275722b6

Very good blog article, “Understanding Single-Rate and Dual-Rate Traffic Policing”.
http://blog.ine.com/2011/05/22/understanding-single-rate-and-dual-rate-traffic-policing

Cisco “Shaping and Policing Overview”
https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_plcshp/configuration/xe-16/qos-plcshp-xe-16-book/qos-plcshp-oview.html

This document as a PDF:
http://www.networkdetective.com.au/PDFs/Reddit-At-A-Loss.pdf

Author Profile

Phil Storey is a freelance troubleshooter and performance analyst for both networks and applications. He has more than 35 years’ experience in IT and communications.

Phil’s early years were as a technical expert for US companies NCR and AT&T and included connecting Unix systems into mainframe SNA networks, developing database applications and connecting PCs to the fledgling Internet (all with their attendant problem solving and troubleshooting components). Phil’s later career, as a Network and Infrastructure Architect at a major Australian bank, was less technically detailed, but far broader ranging.

A hidden passion was unleashed in 2009 when he was introduced to the art of packet capture and analysis – moving his career back to the world of extremely detailed technical knowledge.

After a stint at a major telco using NetScout, Phil’s analysis tool of choice is now NetData Pro from the Australian company, Measure IT Pty Ltd, because it allows him to very quickly get to the bottom of very tough and very complex problems or application/network behaviours.

Phil holds a Bachelor of Computer Science degree from the University of Sydney (as well as a Postgraduate Diploma in Applied Finance and Investment).

Contact

If you are “at a loss” with tough performance problems, if you want facts (not guesses) and if you want real answers (not finger pointing), or just to discuss anything - please find me at:

Phil Storey
Website: www.NetworkDetective.com.au
LinkedIn: au.linkedin.com/in/philipstorey3
Twitter: @PhilStorey24
YouTube: www.youtube.com/c/NetworkDetective