Google Cloud

Headers and Data and Bitrates!

And MTUs too! Oh My!

Rick Jones

Disclaimer: In no way, shape, or form should the results presented in this document be construed as defining an SLA, SLI, SLO, or any other TLA. The author's sole intent is to offer helpful examples to facilitate a deeper understanding of the subject matter.

Introduction

There are any number of things which limit the performance of TCP/networking. There can be too-little CPU available, or not enough window to accommodate a desired bitrate and distance. And often people will discuss how a given "NIC" supports a particular bitrate - eg 100 Gbit/s. And they may even speak of "achieving link-rate" through that NIC. But when it comes to the rate at which data can be transferred between applications, there is more to "link-rate" than just the rate at which bits get toggled onto the fibre.

Headers

Many headers are well-known - Ethernet, IP, and TCP headers come to mind. And in the physical world, those will account for the vast majority of the headers out there. Perhaps the occasional VLAN tag or IPSEC header. Cloud networking will include some/many of those, and others. Some of which are known to the general public and some which are not.

Data

Data is just that - data. That which is not headers. The useful information we wish to send from one system to another. Of course, what is data depends on the layer at which one is looking. If one is looking at say the Ethernet/Data-Link layer, everything besides the Ethernet header is "data" - the IP header, and TCP header, and the rest. And when looking at the Network/IP layer, the TCP header is "data." And so on and so forth. Data is useful. Headers are overhead.

Bitrates

Bitrates in the physical world are the rates at which bits can be toggled onto the wire/fibre/air. Usually after any physical layer encoding, which is a topic beyond the scope of this document. As far as the "links" are concerned, bits are bits be they headers or data. We are all familiar with the inexorable march of Ethernet bitrates from 10 Mbit/s up through 100 Gbit/s and counting. Those represent a hard limit to how many bits per second will pass through the port of a NIC.

In a Cloud, where VMs (Virtual Machines) get provisioned with less than “link rate” egress (and in some cases, ingress) caps there is still, ultimately, some piece of physical networking hardware over which those VMs’ traffic must pass. These egress caps tend to be somewhat lower than the hard, physical bitrate limits. But there are times when the egress caps can be the same as the hard, physical bitrate limits.

MTUs Too!

MTUs - Maximum Transmission Units - are the size limits of packets at a given layer in the networking hierarchy. The most commonly discussed MTU is that of IP. For example, the “1500 Byte MTU” often bandied about for Ethernet is actually/also the MTU for IPv4 over Ethernet. The IP MTU includes both the IP header and the IP payload (aka “Data”). Subtracting-off the IP and TCP and … headers from this MTU tells us how many bytes of our data may be carried on the fibre/wire/air before we must pay the cost of another set of headers.

Oh My! Putting it all together

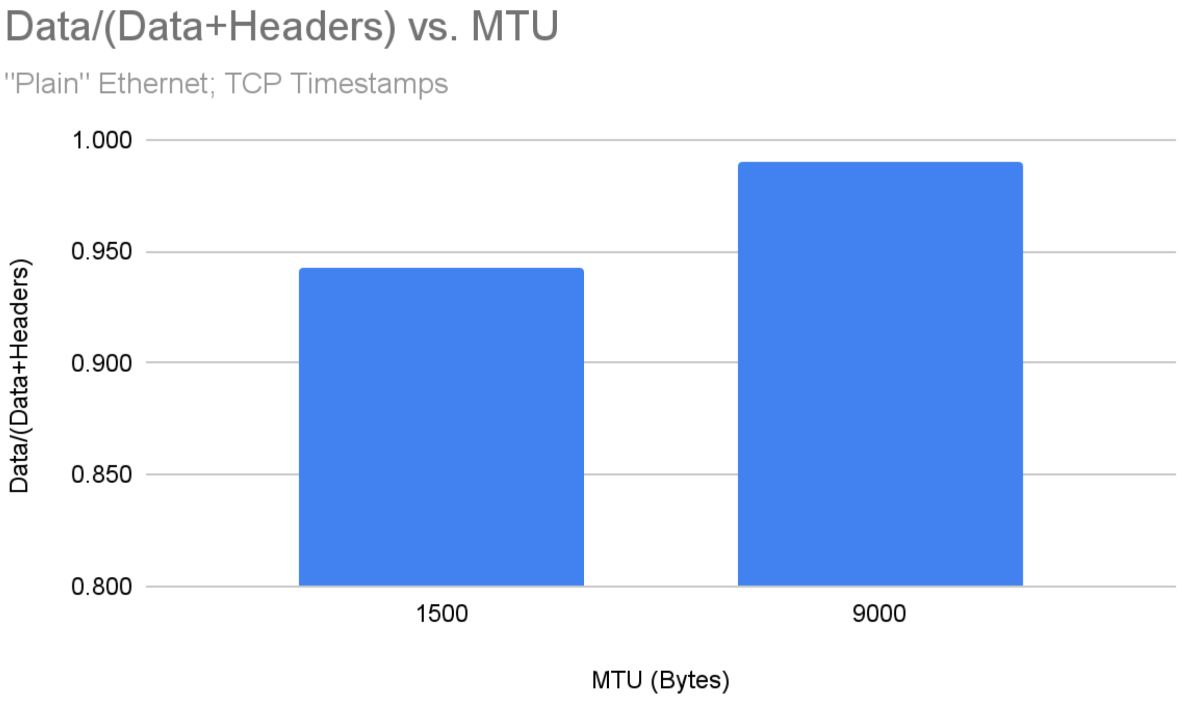
So, what does it all mean about what “link rate” happens to be for a given data transfer? Well, it means we can express the fastest rate at which we can transfer “data” as a function:

$$MaxDataTransferRate \leq \frac{DataPerPacket}{DataPerPacket+HeaderPerPacket} \times PhysicalBitRate$$

DataPerPacket will depend on the MTU of the Link/Interface[1]. By and large, the size of the headers per packet does not depend on the MTU of the Link/Interface, so as the MTU becomes larger, the ratio of *DataPerPacket* to *DataPerPacket+HeaderPerPacket* gets closer and closer to 1 (one). In other words, as the MTU becomes larger, the effect of headers on maximum achievable bulk data transfer rate becomes smaller. In the physical world, almost everything is “Ethernet” and there are two common, or at least one standard and one not-unheard-of MTUs, 1500 and 9000 bytes. And we can then see what those ratios are for a “plain” Ethernet network:[2]

[1] Well, maximum data per packet anyway. Not all packets are full-sized…

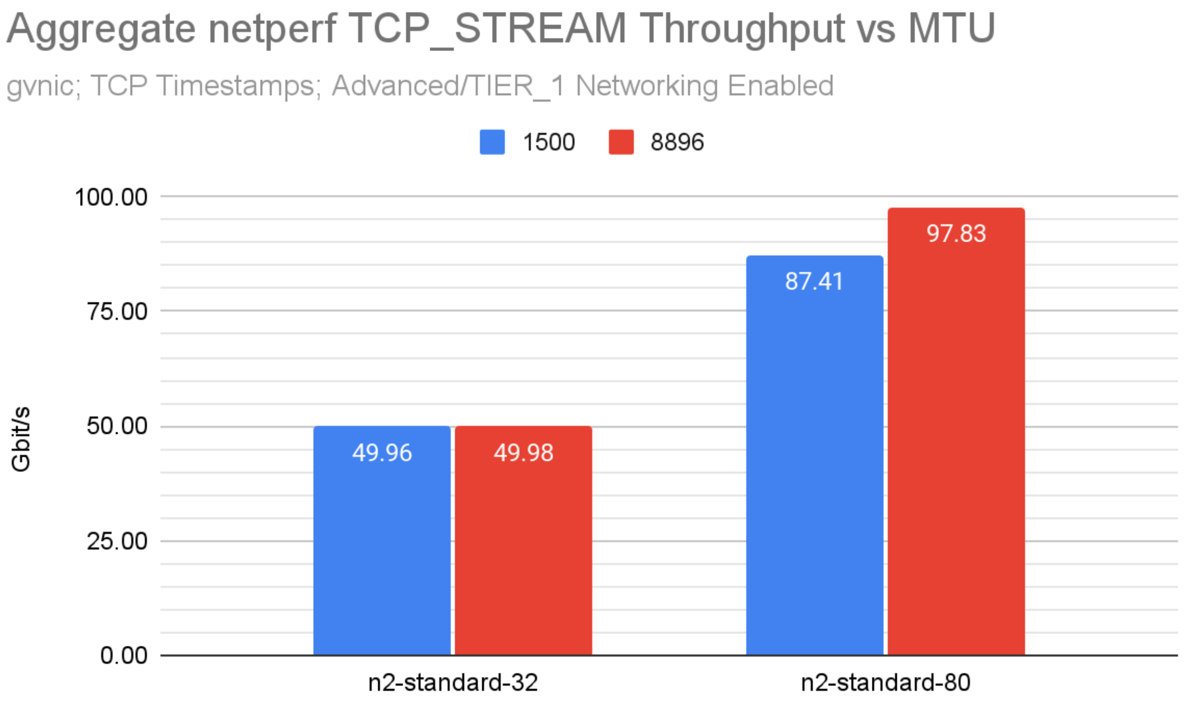
[2] Why TCP Timestamps? Because no one should ever disable TCP Timestamps, and if they do they should/must also disable TCP Window Scaling. TCP Timestamps form the backbone of Protection Against Wrapped Sequence Numbers (PAWS) when Window Scaling is enabled.



So, in terms of what say netperf or iperf3 might report for a bulk TCP transfer on a plain Ethernet network, where we would have 1432 bytes of payload per TCP segment, plus 32 bytes of TCP header, 20 bytes of IPv4 header, 14 bytes of Ethernet header, 7 bytes of Preamble, 1 byte of Start of Frame Delimiter, 4 bytes of Frame Check Sequence, and 12 bytes of Inter-Packet Gap; we would expect to see no more than about 94% of the physical bitrate when the MTU is 1500 bytes. It would be more like 99% with a 9000 byte MTU - the same number of header bytes per packet, but many more data bytes.

But What About Clouds?

Clouds obscure things. They not only add more headers to the mix, they also have per-VM egress caps. But no matter what, Clouds are still bound by physical constraints like everything else. They cannot change the laws of physics. So, let’s take a brief look at aggregate, outbound netperf TCP_STREAM throughput for two VM flavors in Google Cloud (n2-standard-32, n2-standard-80) and MTUs (1500 and 8896 bytes). Both VMs use the “gvnic” vNIC and have Advanced/TIER_1 networking enabled to give them an egress cap of 50 and 100 Gbit/s respectively.



You may be thinking "Wait! I thought you said with a 1500 byte MTU, the most one could get pushing data via TCP would be 94% of link-rate, and an n2-standard-32 has an egress cap of 50 Gbit/s? Why isn't it at 47 Gbit/s instead of almost 50?"

The answer is: An egress cap is not the same thing as a physical link bitrate[3].

The n2-standard-32 VM(s) used in these tests were running on physical hosts with physical NICs with a physical bitrate of 100 Gbit/s. There were enough additional bits-per-second to still be able to send the headers while the data flowed at nearly 50 Gbit/s.

OK, you say, "So why then didn't the n2-standard-80 with its 100 Gbit/s Egress Cap achieve more like 99% of 100 Gbit/s? Wasn't it too on a physical host with a physical NIC with a physical link-rate of 100 Gbit/s?"

The answer is: There is more in Google Cloud packet headers than is dreamt of in a pure Ethernet/IP/TCP philosophy.

[3] With N2 VM Machine types at least. And in Google Cloud, an egress cap is a "Guaranteed not to exceed; not guaranteed to achieve" figure.

While there was “room” in the physical bitrate to allow the n2-standard-32 to get to its 50 Gbit/s Egress Cap at the “user-space level” with the added encapsulation headers used in Google Cloud, there wasn’t “room” to allow the n2-standard-80 to get to its 100 Gbit/s Egress Cap with a 1500 byte MTU. But when we increased the MTU to 8896 bytes, the ratio of data to data+headers improved to the point where we were within a percent or three of the physical limit.

Conclusion

MTUs and header sizes will determine how much useful data may be carried per-packet which will then affect how close to the bitrate of the physical link user-level data transfer will get. And a VM’s egress cap, while kindasorta looking like a NIC bitrate, isn’t the same thing.

Notes

- Aggregate outbound throughput was measured as the highest average bitrate achieved over a two minute interval by parallel netperf TCP_STREAM tests from the VM/Instance Under Test (IUT) to the two load generators/sinks.
- One setup was configured with a VPC MTU of 1500 bytes. The other was configured with a VPC MTU of 8896 bytes.
- The default VPC MTU in Google Cloud is 1460 bytes. That would be 40 fewer bytes of data per packet, so a slightly lower maximum than 1500.
- To obtain the 50 and 100 Gbit/s VM egress caps for the IUTs, they were configured to use the “gvnic” vNIC. Those egress caps are not available to VMs using the “virtio-net” vNIC.
- The tests were conducted in the us-central1 Google Cloud region.
- TCP Segmentation Offload (TSO) remained enabled for all these tests.

Your mileage will vary.

Acknowledgments

The author would like to thank Derek Phanekham for editing and updating this document for public release.