The Early Implications of NVMe/TCP on Ethernet Network Designs

The ratification in November 2018 of the NVMe/TCP standard officially opened the doors for NVMe/TCP to begin to find its way into corporate IT environments. Earlier this week I had the opportunity to listen in on a webinar that SNIA hosted which provided an update on NVMe/TCP's latest developments and its implications for enterprise IT. Here are four key takeaways from that presentation and how these changes will impact corporate data center Ethernet network designs.

First, NVMe/TCP will accelerate the deployment of NVMe in enterprises.

NVMe is already available in networked storage environments using competing protocols such as RDMA which ships as RoCE (RDMA over Converged Ethernet). The challenge is no one (well, very few anyway) use RDMA in any meaningful way in their environment so using RoCE to run NVMe never gained and will likely never gain any momentum.

The availability of NVMe over TCP changes that. Companies already understand TCP, deploy it everywhere, and know how to scale and run it over their existing Ethernet networks. NVMe/TCP will build on this legacy infrastructure and knowledge.

Second, any latency that NVMe/TCP introduces still pales in comparison to existing storage networking protocols.

Running NVMe over TCP does introduces latency versus using

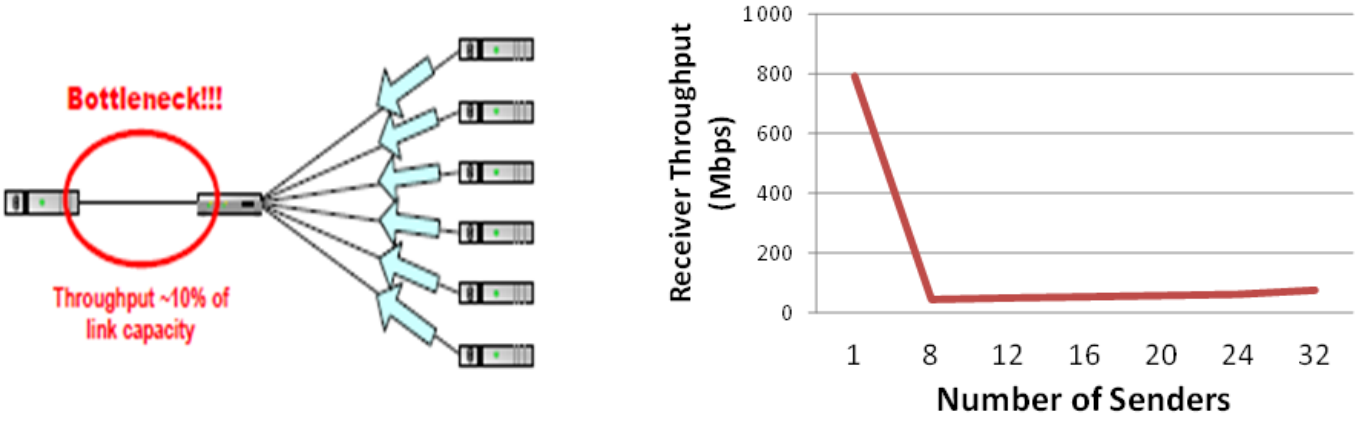
RoCE. However, the latency that TCP introduces is nominal and will likely be measured in microseconds in most circumstances. Most applications will not even detect this level of latency due to the substantial jump in performance that natively running NVMe over TCP will provide versus using existing storage protocols such as iSCSI and FC.

Third, the introduction of NVMe/TCP will require companies implement Ethernet network designs that minimize latency.

Ethernet networks may implement buffering in Ethernet switches to handle periods of peak workloads. Companies will need to modify that network design technique when deploying NVMe/TCP as buffering introduces latency into the network and NVMe is highly latency sensitive. Companies will need to more carefully balance how much buffering they introduce on Ethernet switches.

Fourth, get familiar with the term "incast collapse" on Ethernet networks and how to mitigate it.

NVMe can support up to 64,000 queues. Every queue that NVMe opens up initiates a TCP session. Here is where challenges may eventually surface. Simultaneously opening up multiple queues will result in multiple TCP sessions initiating at the same time. This could, in turn, have all these sessions arrive at a common congestion point in the Ethernet network at the same time. The network remedies this by having all TCP sessions backing off at the same time, or an incast collapse, creating latency in the network.



Source: University of California-Berkeley

Historically this has been a very specialized and rare occurrence in networking due to the low probability that such an event would ever take place. But the introduction of NVMe/TCP into the network makes the possibility of such a event much more likely to occur, especially as more companies deploy NVMe/TCP into their environment.

The Ratification of the NVMe/TCP

Ratification of the NVMe/TCP standard potentially makes every enterprise data center a candidate for storage systems that can deliver dramatically better performance to their work loads. Until the performance demands of every workload in a data center are met instantaneously, some workload requests will queue up behind a bottleneck in the data center infrastructure.

Just as introducing flash memory into enterprise storage systems revealed bottlenecks in storage operating system software and storage protocols, NVMe/TCP-based storage systems will reveal bottlenecks in data center networks. Enterprises seeking to accelerate their applications by implementing NVMe/TCP-based storage systems may discover bottlenecks in their networks that need to be addressed in order to see the full benefits that NVMe/TCP-based storage.

To view this presentation in its entirety, follow this link.