Thoughts on Data Center Virtualization via SDN, TCP Optimization

Posted by Sue Hares on September 25, 2013 at 9:30am View Blog

VMware's NSX deployment with STT (Stateless Transport Protocol for network Virtualization)requires that one relook at TCP as a requirement for end-to-end Software Defined Networks (SDNs).

VMWare Network Virtualization Proposals

VMWare utilizes Stateless Transport Protocol for network Virtualization (STT) which has TCPlike frames to provide Distributed Edge Overlays (DEO) for an over-the-top (OTT) virtual network for multi-tenant topologies. The DEOs are being standardized by IETF's nv03 working group to support multi-tenant data center connections to other data centers (public cloud or private cloud) or to users via the Internet or private VPNs. The distributed edge overlays (DEOs) connect data centers that run either layer 2 or layer 3 protocols. DEOs have the following four components: encapsulation formats, traffic engineering (TE) calculations for OTT tunnels, logical views, and management systems. The IETF nv03 working group has seen DEO proposals include: NVGRE, VXLAN, STT, and modified L2 VPN protocols.

Figure 1 NVGRE Encapsulation

Encapsulation formats used in these SDN protocols include: layer 2 (L2) in L3 (NVGRE), L2 in layer 4 (L4) UDP (VXLAN), and STT's encapsulation of L2 in pseudo-TCP. Figure 1 shows how NVGRE encapsulates Layer 2 in GRE tunnels. Figure 2 shows how VxLAN encapsulates L2 in UDP packets. Figure 3 shows how STT encapsulates the L2/L3 packet in TCP.

Traffic engineering in NVGRE and VXLAN requires multicast support in the IP layer. STT utilizes SDN engineering not specified in the draft. The VXLAN, NVGRE, and L2VPN proposals have a logical view of an L2 learning domain. The logical domain that STT takes is a pseudo-TCP end-to-end approach.

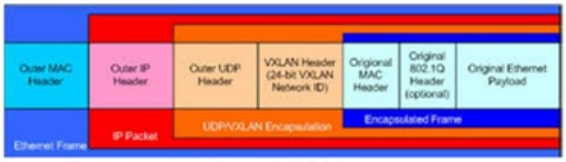

Figure 2 VXLAN

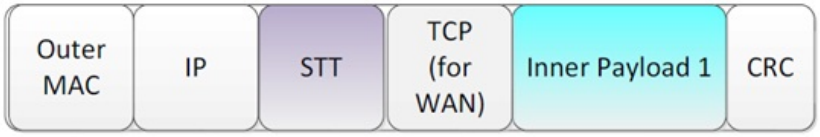

Figure 3 - STT encapsulation

Why did the STT Authors Take This TCP Level Approach?

In their STT draft, Davie and Gross stated the STT design goals were to utilize the hardware based TCP transport offload (TLO) and efficient large TCP packet receive operation (LRO); support Equal Cost Multipath (ECMP) at the transport layer per TCP flow; and enable unique traffic engineering based on context specific flows for transport data based on Metadata and TCP connection. These design goals point toward another step away from the initial Open Flow approach of hardware switches that Nicira's CEO Martin Casado promoted and then recanted from in favor of virtual switches controlled by Open Flow SDN. TCP is the first end to end level protocol. By trial and effort Nicira (the chief component based SDN proponent) and VMware (the chief L2 over L3 virtualization proponent), have come to virtual TCP as the focal point for end-to-end SDN.

However, why did Davie and Gross choose to do a pseudo-TCP instead of TCP?

End-to-End SDN
End-to-End SDN is SDN that stretches from application to application or application to user. It encompasses all of the IT products and services as a Virtual IT world empowered by SDN.

For decades, the end-to-end IT world has been transitioning to the Virtual IT world with the aid of software defined replacements for server computing power, storage devices, and networking devices (routers and switches).

Pseudo TCP for STT Versus Real TCP Optimization

One of the stated concerns of STT authors was the TCP control block size within the STT data center environment. However, what about the WAN environments in the DEO uses cases? Let's compare the use of TCP within STT with the IETF's TCP with Quick Start (RFC 4792) with TCP packet de-duplication plus an SDN traffic engineering controller (centralized or distributed) and include one loss packet in the stream. To provide STT the best benefit, we'll ignore the any IP layer tunnels (GRE or gif) tunnels or the virtual LANs (VLANs) from IP layer technologies and focus on the TCP scaling for a DEO.

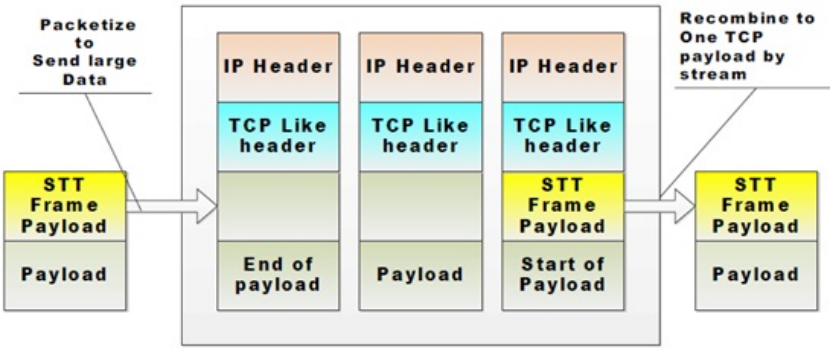

Figure 4 – STT packing process

Without packet loss, STT itself does not suffer any issues that TCP has. STT can chop a large application packet's data that normally goes over UDP into STT. STT will send it out as a large TCP Segment (even though it is not one). If no packet is dropped, then everything is great. If one packet is dropped out of the UDP Stream, you have problems. When STT hands the pieces up to the Large Receive Operation (LRO), the STT packet reassembly will occur and there will be one packet lost. The application will need to retransmit the stream.

How often will the STT protocol drop packets mid-stream due to errors? In a data center with short distances and high quality lines, the time between drops may be large. In DEO environment, the WAN environment that connects Data Center to Data Center Environment or the Data Center to SMB offices will likely have more delay and more drops. At this point, STT could require that TCP runs on top of the STT protocol to carry TCP or UDP data (STT + TCP). These two TCP headers would be required in WAN environment with any delay or drops.

TCP's Problems STT Will Encounter on WAN

TCP is a byte oriented protocol in a packet world with a slow-start mechanism to protect the network. Groups of TCP packets must be reassembled into byte streams via hardware assists (TCO or LRO) or by removing duplicate packets being sent. Up to 50% of web traffic may be duplicate packets.

The TCP slow start mechanisms try to handle congestion by building up the window gradually. The packet sending rate for a TCP connection is increased one maximum segment size (MSS) for each TCP ACK in slow start. If TCP congestion control detects a lost packet, it cuts the rate by 50%. Due to this cut, end-to-end TCP is throttled by the slowest network link in the end-toend path. Any two applications exchanging data via TCP or via a HTTP session running over TCP may hit these congestion throttle backs where data slows by 50%. These two application entities can be an application running in a data center on a virtual system or a user on a mobile phone or a large well connected server.

There is a distance bias in normal TCP congestion control algorithms. Short distance sessions may have packet loss or drop the connection, but they recover quickly. Long distance sessions impacted by packet loss recover slowly.

Solution to TCP's Problem: Shorten TCP's Distance and Up the TCP Rate

The solution to the TCP problem is "shortening the distance" by tuning the network paths at the TCP layer for each session flow. This tuning needs to be done on a continual basis since network congestions may change mid-flow. The end-to-end SDN controller can provide the traffic engineering logic to provide optimal TCP rates based on the traffic engineering done at layer 2 through layer 7. The Quick Start protocol for TCP and IP (RFC4782) provides a standard mechanism to provide this feedback to TCP and IP. Quick start will allow the IP or the TCP to go to a maximum rate immediately starting at the end host. If you added a "router sets rate" option, and the network that runs over WAN links implements TCP Quick start (as shown in figure 6), the long distance expensive TCP WAN links can utilize their full bandwidth.

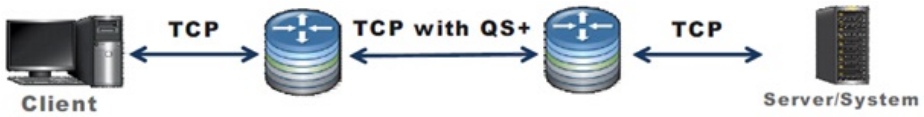

Figure 5 TCP Quick Start+ used on WAN Links

The result change of TCP's performance (Quick start plus "router sets rate") can be seen in the graph below. The diagram on the left in figure 6 is normal TCP running over a WAN link (from figure 5 above) where the loss of packets causes the TCP saw-tooth pattern of increasing and decreasing window sizes. The diagram on the right in figure is a TCP which implements TCP Quick start guided by end-to-end SDN traffic engineering per TCP flow. These are actual result for a TCP Run on a WAN link.

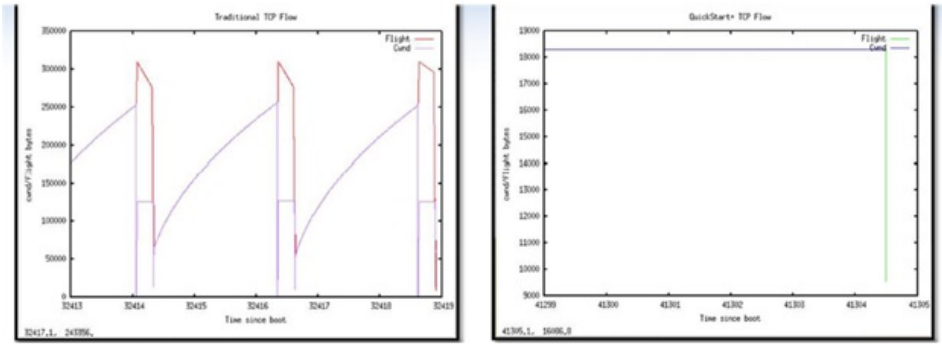

Figure 6 – Traditional TCP vs. TCP QS+ router option with end-to-end SDN

Quick start+ for TCP and IP supports both IPv4 and IPv6. TCP and IP Quick start utilize a nonce value (QS none) to provide the sender some protection that receivers are not lying about the receive rate. TCP Quick start with SDN end-to-end traffic steering operates over IP tunnels and MPLS providing a multi-path transport such as STT desires, but with flow control and acknowledgements. Running over tunnels avoids any router executing "slow start" path on the QuickStart option.

TCP Quick+ start may also aid in resetting the TCP transmit parameters in a persistent-HTTP connection that has started to exchange data after an idle period.

What is STT Doing?

STT makes sense run simply within a data center on lossless links that have TCP acceleration (TCO or LRO). Running STT on DEOs does not make sense. The WANs and mobile devices will lose data and have delays. If loss means STT will run TCP over STT, then all traffic will have two TCP headers on WANs. The STT approach adds bytes to the WAN packet, and does not fix the TCP window issues (slow start and congestion) for the DEO environment.

Why not examine a combination of IETF Quick Start plus a "router-add function" to augment the multi-path end-to-end SDN control?

Sue Hares is vice president of technology and strategy at ADARA Networks.