Stateless Transport Tunneling

draft-davie-stt-01.txt

Bruce Davie, Jesse Gross, Igor Gashinsky et al.

Outline

- Motivation
  - Why Network Virtualization needs tunnels
  - Performance for software
  - Backwards Compatibility
  - Flexible Control Plane
  - Context Identification
- Frame and Segment formats
- Open Issues & Next Steps

Why tunnels?

- Manage overlapping addresses between multiple tenants
- Decouple virtual **topology** provided by tunnels from physical network topology
- Decouple virtual **network service** from physical network (e.g. provide an L2 service over an L3 fabric)
- Support VM mobility independent of the physical network
- Support larger numbers of virtual networks (vs. VLANs for example)
- Reduce state requirements for physical network (e.g. MAC addresses)
- Because all CS problems can be solved with another level of indirection

Performance for SW

- Tunnels for Network Virtualization often originate in the hypervisor
- Lots of NICs support TSO/LRO (TCP Segmentation Offload/Large Receive Offload)
- Most NICs won't do TSO with any existing tunneling encaps ➔ significant performance loss when tunneling
- STT uses a header that can be generated by today's NICs when performing TSO
- A few other details in the draft to improve SW performance

Backwards Compatibility

- NICs
- Routers and switches
  - Source Port chosen to be constant per microflow, randomized for ECMP
- WAN services
  - They carry IP or Ethernet
- Middleboxes
  - Some work required (often true for tunnels)
  - Many also use SW implementation, reap TSO benefits

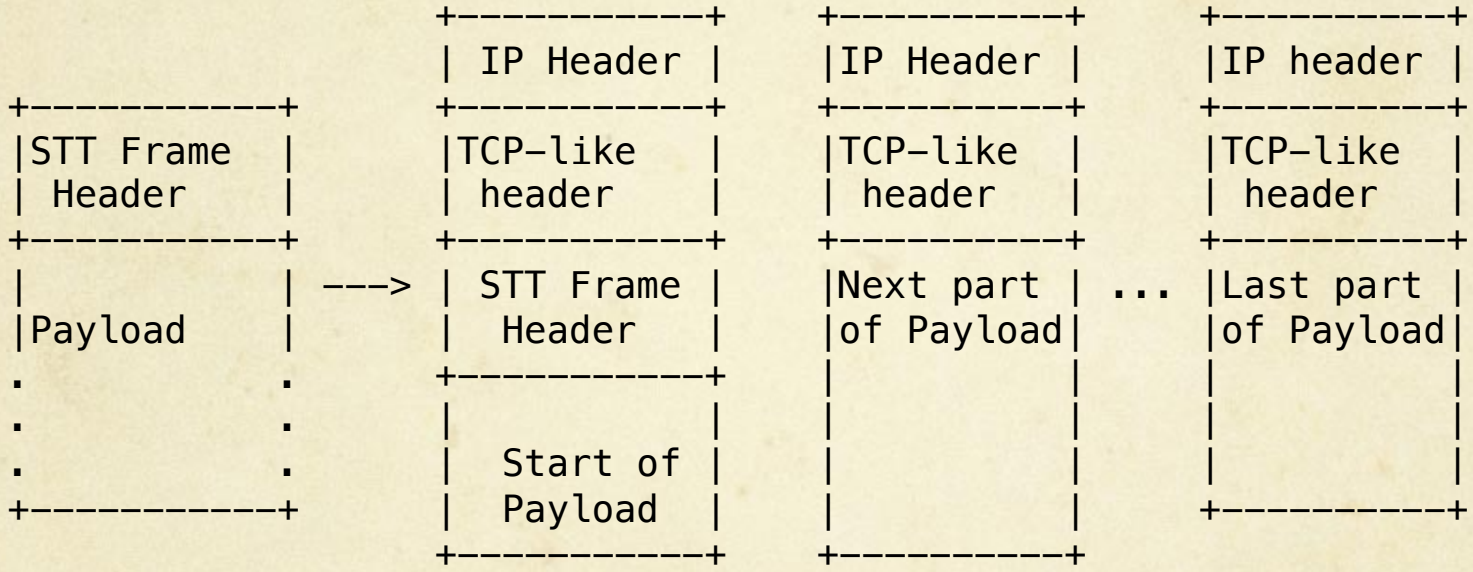
Flexible Control Plane

- Control plane should not be specified as part of tunnel encaps
  - Allow control plane to evolve
- Even putting "Virtual Network Instance ID" in data plane starts to constrain the control plane
  - Note that MPLS VPNs have a much more rich notion of VPN membership than a single VPN-ID can offer

Context Identification

- As packets exit from tunnels, need to deliver them to the right "context"
  - A context may be simply a "tenant" or "virtual network instance", but they are special cases
  - Can also use it for other metadata (state versioning, distributed lookup, etc.)
  - An opaque context ID with control-plane defined semantics also supports control-plane independence goal

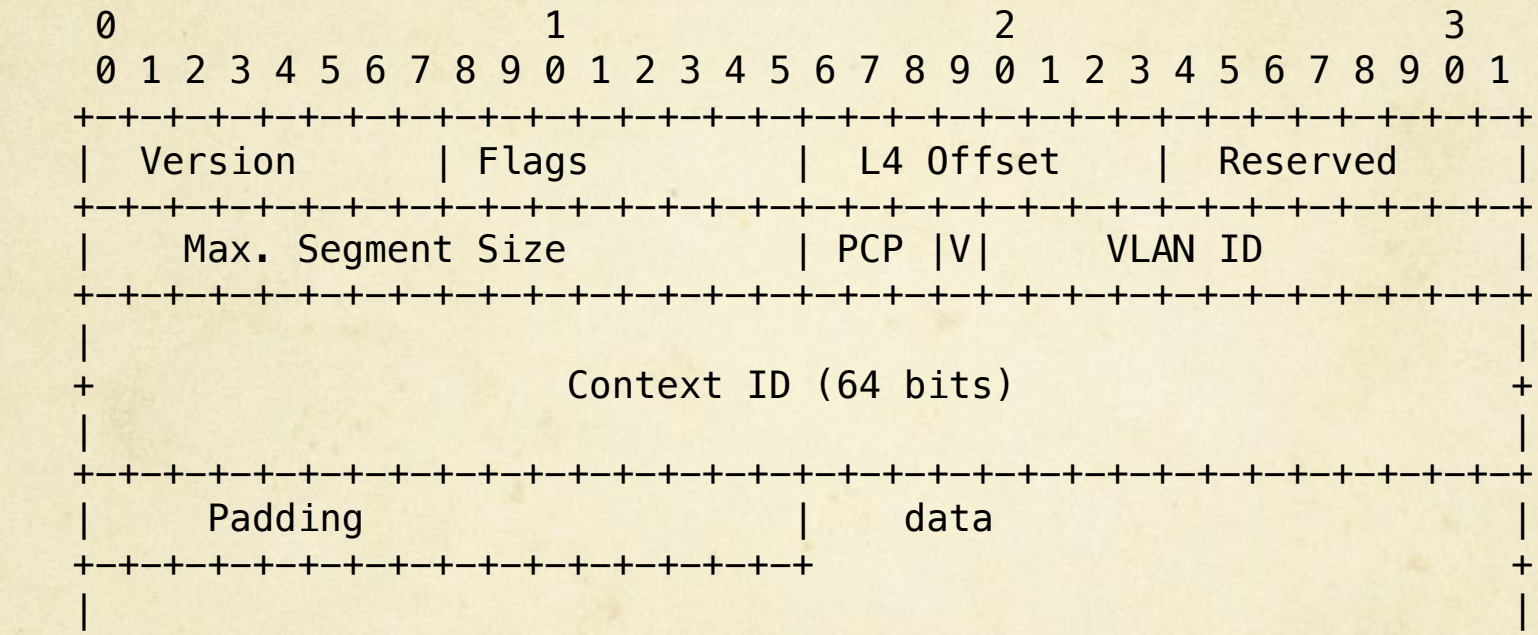
STT Encapsulation



Original data frame is encapped with STT Header

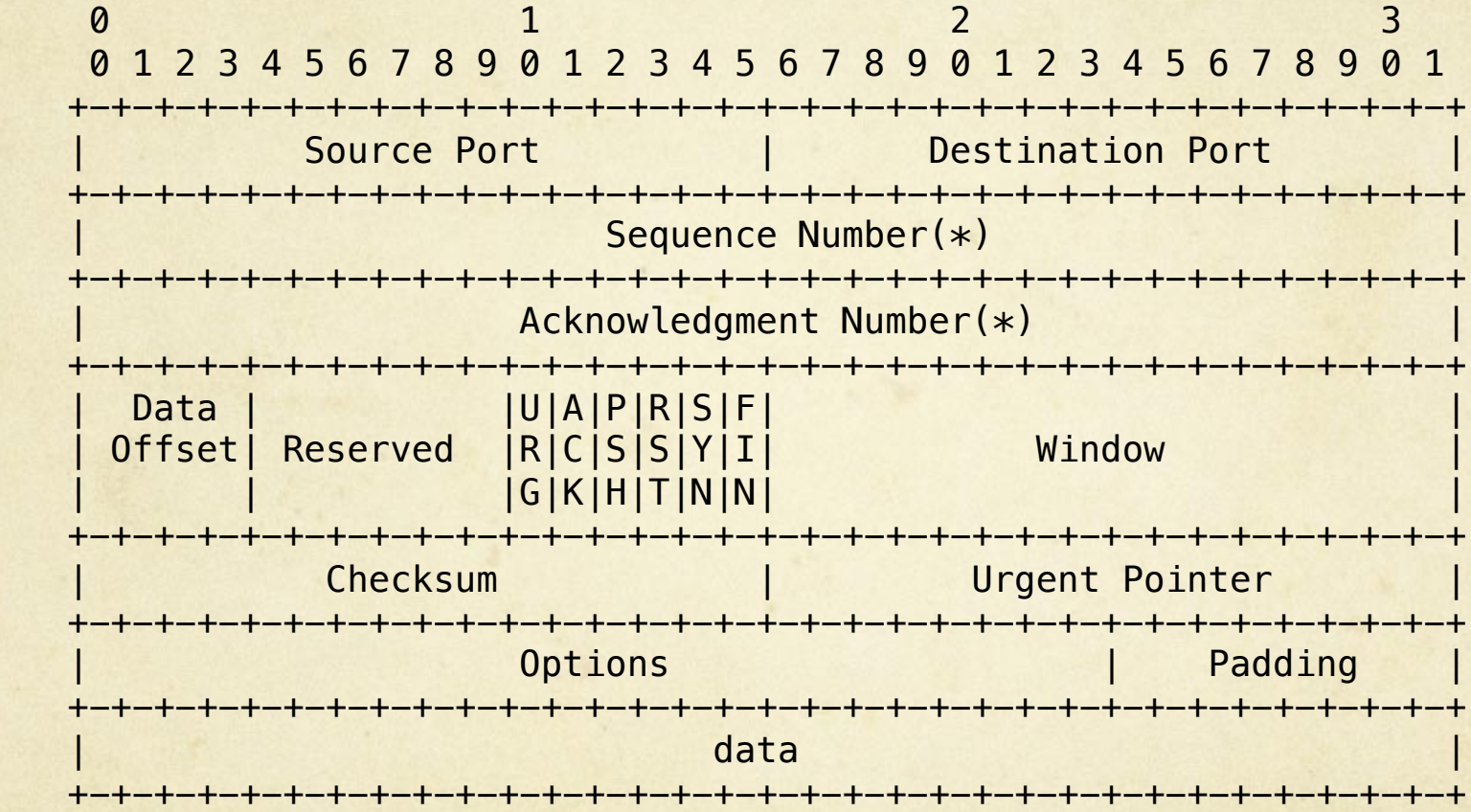
STT Frame is segmented and transmitted as a set of TCP segments (MAC headers not shown)

STT Frame



Up to 64kB, transmitted as a series of STT Segments

STT Segment



SEQ and ACK are repurposed to support reassembly of STT Frames

Open Issues

- Clearly this will confuse devices that expect a complete TCP state machine to exist
  - Most common result would be drop the packets
- Well-known port to be requested from IANA
- When middle boxes are in the path, need to
  A. Teach them at least to pass the packets, or
  B. Enable them to reassemble Frames for further processing

Next Steps

- Everything that is in STT could be done without "repurposing" the TCP header
  - We'd like to see the STT requirements considered in NVO3
  - See you in a few years when the NIC vendors build this
- Meanwhile, would be good to get more implementations (e.g. middleboxes)
- Not clear if any IETF WG is chartered to work on this yet, but L2VPN is closest fit