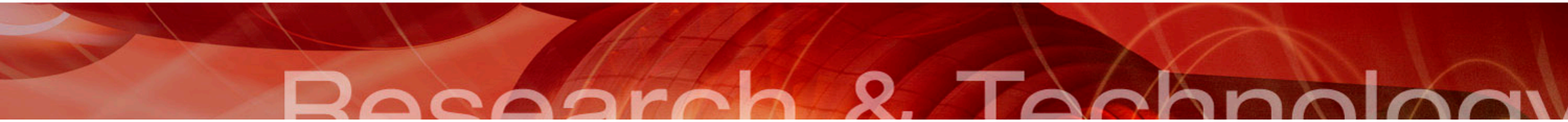
Engineering, Operations & Technology
Boeing Research & Technology



Operational Issues with Tunnel Maximum Transmission Unit (MTU)

Boeing Research & Technology

Fred Templin (fred.l.templin@boeing.com)

11/08/2012

The Problem

- **De facto "Internet Cell Size" is 1500 bytes**
- **Tunnels add encapsulation overhead that reduces the effective path MTU**
- **Tunnels often adapt by setting a conservative and fixed MTU (e.g., 1480 bytes). However:**
  - Path MTU Discovery messages are often filtered
  - IP fragmentation is problematic
  - Larger packets that might make it through the tunnel in one piece are discarded at the ingress
- **Issues apply to tunnels over both IPv4 and IPv6**

Path MTU Discovery (PMTUD) Issues

- **When a too-large packet is dropped at the tunnel ingress:**
  - Packet Too Big (PTB) message produced by the ingress may be dropped on the path to the original source
- **When a too-large packet is dropped inside the tunnel:**
  - PTB message may be dropped on the path to the tunnel ingress, or
  - PTB message may not contain enough information for translation into PTB to send back to the original source, or
  - PTB message may be fabricated by an adversarial middlebox within the tunnel

IP Fragmentation Issues

- **Original source could use IP fragmentation *before* encapsulation**
- **Tunnel ingress could use IP fragmentation *after* encapsulation**
- **However:**
  - For IPv4, IP_ID is only 16bits
  - For IPv6 (and probably also IPv4) middleboxes are being configured more and more to drop all IP fragments

Current Mitigations

- **As a result, common tunnel types set a fixed and static MTU of at most 1500 minus the length of the encapsulation headers (e.g., 1480 bytes for IPv6-in-IPv4)**
- **However:**
  - Minimum MTU is only 1280 bytes for IPv6 and 576 (68?) bytes for IPv4 so there is no way to set a “low enough” static MTU
  - MTU loss within the tunnel still result in black holes
  - **Especially problematic for tunnels-within-tunnels**

Alternative Approach

- **Tunnel ingress could use “tunnel fragmentation” *before* encapsulation**
  - application-layer segmentation (the tunnel ingress is the “application”)
  - Reassembly performed by the tunnel egress
  - Each segment appears as an individual IP packet on the wire (i.e., and not as an IP fragment)
  - Extra “mid-layer” of encapsulation needed

Other Considerations

- **The tunnel should set an indefinite MTU (i.e., admit all packets into the tunnel regardless of their size and make any necessary adaptations from within the tunnel)**
- **“Take care of the smalls, and let the bigs take care of themselves”**
  - Make sure packets no larger than 1500 get through
  - Let larger packets sink or swim on their own
- **Assumes that original sources that send packets larger than 1500 use RFC4821**

Problem Statement and Approach

- **Operational Issues with Tunnel Maximum Transmission Unit (MTU)**
  - draft-generic-v6ops-tunmtu
  - https://datatracker.ietf.org/doc/draft-generic-v6ops-tunmtu/
- **The Subnetwork Encapsulation and Adaptation Layer (SEAL)**
  - RFC5320 (early experimental version)
  - draft-templin-intarea-seal (SEAL(bis))
  - https://datatracker.ietf.org/doc/draft-templin-intarea-seal/