# The Case Against Jumbo Frames

Richard A Steenbergen <ras@gtt.net>
GTT Communications, Inc.

# What's This All About?

- What the heck is a Jumbo Frame?
  - Technically, the IEEE 802.3 Ethernet standard defines the maximum frame payload (MTU) value at 1500 bytes.
  - Supporting anything larger than 1500 bytes is outside of this standard, and we call it a "Jumbo Frame".
- How much bigger?
  - Nobody really knows. It's non-standard, remember.
  - The "rough guideline" for most people is around 9000.
    - This is a historical number from the original Alteon proposal.
  - Most vendors today actually offer a different number.

# It’s Over 9000!!!

# The Goal of Jumbo Frames

- Improved High Speed Transfer Efficiency
  - Most basic host operations are aligned around 4096 bytes.
    - Memory pages, iSCSI data blocks, etc, etc.
    - Sending 1500 byte packets doesn't align well with these operations
  - Results in poor chunking and waiting for buffers to fill.
- Reduced packet/sec routing lookup load.
  - Increasing packet sizes can decrease the PPS rate.
- Increased flexibility when tunneling.
  - 1500 byte MTUs are the "defacto standard" of the Internet.
  - So doing 1500 plus tunnel overhead is hard to do well.

# The Intended Use of Jumbo Frames

- The current paradigm is to fill the MTU to the max.
  - Sending many MB of data in 1460 + Headers chunks.
  - Most of packets on the Internet are 1500 bytes long.
    - The rest are mostly TCP ACKs to those 1500 byte packets. ☺
- The proposed Jumbo Frame usage is different.
  - You aren't supposed to send 9K packets every time.
  - Send 4096*2 = 8192 bytes of payload, plus headers.
  - 8192 payload + IP (20) + TCP (20) + Ethernet (14), etc.
  - The extra buffer up to 9000(something) is intended to increase flexibility for using different packet headers.

# Starting To See Inter-Network Deployment

- Until recently, jumbo frames have primarily been an “internal network only” thing at best.
- But now some IX operators are starting to roll out Jumbo Frame VLANs at major exchanges.
- This ***could*** eventually lead to the ability to deliver a > 1500 byte packet end-to-end.
- And many people are cheerleading this effort.
  - With a lot of idealism about improving the efficiency of high-speed end-to-end flows, which is a good thing.

# But this could all be a Very Bad Idea…

# Picking an MTU Number

- Picking an MTU value is really hard work.
  - Remember, there is no standard value defined anywhere.
  - Nor are their any negotiation protocols to automate it.
  - So we're down to manual negotiation between operators.
- And it's all made even harder by router vendors.
  - There isn't even a standard for using the same number!
  - Cisco IOS 1500 equals Juniper & IOS XR 1514.
- But wait, it gets even worse…
  - Cisco IOS 1500 on an 802.1q tagged interface equals Juniper 1518, or 1522 on a Q-in-Q link, etc.
- And if you get it wrong, you silently blackhole traffic.

# Path MTU Discovery Sucks

- Path MTU Discovery (PMTUD) is how the Internet deals with MTU mismatch today.
  - When a router encounters an MTU mismatch and a frame that is too large, it drops the packet and sends an ICMP.
  - The host receives the ICMP, and reduces the packet size for the retransmission and the flow. Hopefully it now fits.
- But PMTUD is broken beyond words.
  - Routers can only generate these ICMP packets so fast.
  - It's incredibly vulnerable to Denial of Service attacks too.
  - ICMP packets are often limited/blocked by ISPs or users.
  - The only reason stuff works today ***IS*** the 1500 byte MTU.

# Other Problems

- Larger packet sizes increases jitter.
  - Making 100 Gigabit flows more efficient is a noble cause.
  - But a single 9000 byte packet down a 10 Mbps link has 7.2ms of serialization delay alone.
- The benefits are non-existent until you have a 100% Jumbo enabled end-to-end path.
- No content network will ever enable a service which has minimal benefit and which risks breaking the flows for any percentage of their customer base.

# And It's All Totally Unnecessary

- Jumbo Frames are a nearly 15 year old idea.
  - 15 years ago, they helped a host deliver a gigabit flow.
  - Today, they're really completely unnecessary for that.
- Modern NICs help eliminate most of this.
  - With techniques like Large Segment Offload (LSO).
  - Instead of making it a big hassle to move data around in 1500-byte chunks, you just hand the NIC a 64k buffer.
  - So we're solving a problem we really don't even have.

# Tunneling PROVES How Broken PMTUD Is

- The argument that establishing > 1500 byte MTU IX's helps enable tunneling is an interesting one.
  - But it actually proves the point that PMTUD is broken.
  - If MTU mismatches in the Internet actually worked, you wouldn't have a problem tunneling 1476 over GRE.
  - Creating more MTU mismatches to solve the problem of MTU mismatch is fundamentally flawed logic.

# What You'd Need To Raise the MTU Bar

- So what do you need before you can even try to dabble in the space of raising the Internet MTU?
- Reliable MTU Negotiation between end-points.
  - Making operators negotiate a value between every connected device, even if the numbers meant the same thing on every device, doesn't scale.
  - You need a negotiation protocol to find Layer 2 MTU.
  - And with Ethernet, this needs to be per-MAC.
  - And for PMTUD to work you need to know it at L3.
  - So the only sensible place to do it would be ARP.

# But What About Path MTU Discovery

- Path MTU Discovery is still the fundamental flaw.
  - Requiring that a router drop a packet, generate an ICMP, have that ICMP successfully make it back to the host before we even KNOW about an MTU mismatch is horrible in every sense of the word.
    - This is not supportable at speed in modern router architectures.
    - This is incredibly vulnerable to Denial of Service attacks.
    - It adds latency and stalls performance on every flow.
  - And there is no replacement on the horizon.
- In Short:
  - Internet-wide Jumbo Frames will probably cause infinitely more harm than good under the current technology.

# Send questions, comments, complaints to:

Richard A Steenbergen <ras@gtt.net>
GTT Communications, Inc.