MTU ("jumbo frames") recommendation for LHCONE and LHCOPN

Shawn McKee/Univ. of Michigan
Mike O'Connor/ESnet/BNL

LHCONE/LHCOPN FNAL Meeting
October 30, 2018

Overview

- Since the startup of LHCOPN and LHCONE we have had recurring issues related to MTU ("jumbo frames")
- At the Spring 2018 meeting in RAL our group decided to put together a small task force to make a recommendation on MTU for LHCONE/LHCOPN
- Mike O'Connor and I will cover the technical details related to MTU and setup a discussion about what our recommendation to LHCONE/LHCOPN network operators should be
  - What is MTU and related terms?
  - Why we would want MTU > 1500
  - Potential problems
  - **Recommendation** (for discussion)

MTU Details

- **MTU** is the Maximum Transmission Unit, i.e., the largest layer 3 data unit that can be communicated in a **single** network transaction
  - Often very confusing because vendors may refer to “media MTU” and “protocol MTU” (or similar vendor specific terms)
  - We will use **MTU** as above (Largest **IP** packet)
  - We will use Frame Size for phys/media/etc “MTU”
- “Jumbo frame”: https://en.wikipedia.org/wiki/Jumbo_frame
- Summarizing Wikipedia:
  - jumbo frames are ethernet frames with IP payloads > **1500B**
  - A related item to define is the **Frame Size** (for ethernet).
- What ultimately matters for **LHCONE** and **LHCOPN** is that the networks carrying that traffic **are able to support ethernet frame sizes suitable for MTU = 9000**.
- We need to determine what that frame size is and we go through the details below, after discussing “Why” we care...

Why Do We Care about MTU > 1500?

- Why do we want “Jumbo Frames” anyway?
  - Larger frames are more efficient in network resource use:

Frame type	MTU	Layer 1 overhead		Layer 2 overhead		Layer 3 overhead	Layer 4 overhead	Payload size	Total transmitted[13]	Efficiency[14]
Standard	1500	preamble 8 byte	IPG 12 byte	frame header 14 byte	FCS 4 byte	IPv4 header 20 byte	TCP header 20 byte	1460 byte	1538 byte	94.93%
Jumbo	9000	preamble 8 byte	IPG 12 byte	frame header 14 byte	FCS 4 byte	IPv4 header 20 byte	TCP header 20 byte	8960 byte	9038 byte	99.14%

- Jumbo frames can deliver better throughput
  - Mathis formula: ***Rate <= (MSS/RTT)*(1 / sqrt{p})***
  - Less CPU overhead in packet processing
- We want end-sites to be able to set their NIC **MTU=9000** and have those packets be able to traverse the **LHCONE/LHCOPN** networks without fragmentation.

Frame Size or What goes into “Media MTU”?

- The **Frame Size** is shown by the green box below

802.3 Ethernet packet and frame structure

Layer	Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46-1500 octets	4 octets	12 octets
Layer 2 Ethernet frame	← 64–1522 octets →								
Layer 1 Ethernet packet & IPG	← 72–1530 octets →								← 12 octets →

- If 802.1Q (VLAN tagging) is not in use, the maximum “standard” frame size is **1518** bytes or **1522** if there are **VLAN tags**
- For Jumbo frames the “Payload” increases. Typical jumbo MTU is 9000 bytes which means the layer-2 Frame Size can be up to **9022 (with VLAN tags)**
- Other protocols can add more information to the packet:
  - MPLS adds 8 bytes, VXLAN adds 50 bytes
- So, supporting a **Frame Size** of 9022+8+50 = **9080 Bytes** should be safe
  - Many operators configure **9100** or even **9192**

Aside: Vendor terminology can be confusing

- https://www.juniper.net/documentation/en_US/junos/topics/task/configuration/interfaces-setting-the-protocol-mtu.html

802.3 Ethernet packet and frame structure

Layer	Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interpacket gap
	7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	46-1500 octets	4 octets	12 octets
Layer 2 Ethernet frame			← 64–1522 octets →						
Layer 1 Ethernet packet & IPG	← 72–1530 octets →								← 12 octets →

- "The actual frames transmitted also contain cyclic redundancy check (CRC) bits, ***which are not part of the MTU***. For example, the default protocol MTU for a Gigabit Ethernet interface is 1500 bytes, but the **largest possible frame size** is actually 1504 bytes; you need to consider the extra bits in calculations of MTUs for interoperability."
  - This is really confusing; why call out the CRC and not the MAC src/dest, optional VLAN tag and Ethertype?
  - The real *largest possible frame size* is **1522** in this case

Potential Problems with non-standard MTU

We discussed earlier why we may want Jumbo frames but what are the downsides?

First, all network interfaces in a given subnet must use the same MTU because there is no router involved in the inter-host communication (i.e. PMTUD doesn't work)

- If two hosts use different MTUs, initial communication (with smaller packets) may work but then fail when large data is transferred

If Path MTU Discovery (PMTUD) is blocked by over-zealous network firewalls, similar issues can happen across the WAN

We have seen cases where just changing the allowed MTU on a network caused hosts running **buggy** applications to try to use jumbo frames (and fail…)

BUT, having jumbo frames allowed on networks where PMTUD works should be safe and doesn't **require** any end-systems to use jumbo frames

Recommendation (For Discussion)

We have documented all the details in a Google doc available at https://docs.google.com/document/d/1Iut-ncRsV1-9Z4o56S9vLVuc3IHbR-0uIx9IiXVLXpY/edit?usp=sharing

Recommendation for the Group:

LHCONE/LHCOPN network paths should **allow** MTU size up to 9000 bytes and not block PMTUD packets (RFCs 1911, 1981 and 4821)

- In practice this means that the **frame size** should be at least **9080 bytes** for all devices on the path
- ICMP “Fragmentation Needed” (**Type 3, Code 4**) should **not** be blocked by any devices on the path

Discussion?

Questions?

Comments?

Shawn McKee smckee@umich.edu
Mike O'Connor moc@es.net