Multi-MTU subnets

draft-van-beijnum-multi-mtu-04

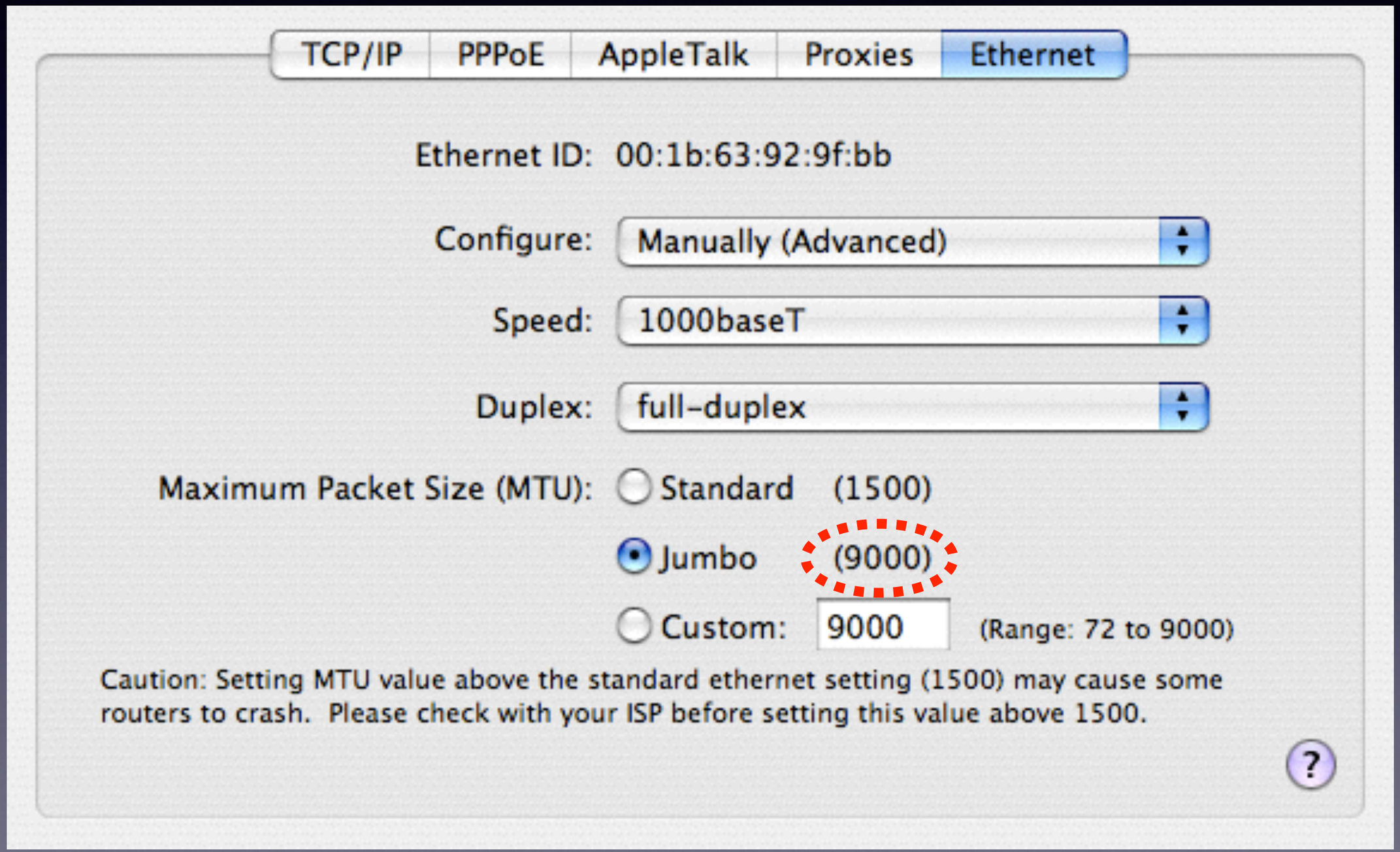
int-area @ IETF-91, November 14, 2014
Honolulu

Iljitsch van Beijnum

Ethernet MTU = 1500

- IEEE 802 values interoperation
  - you can connect 10 Mbps and 10000 Mbps Ethernets and it just works
- But: packets per second getting out of hand
  - 10 Mbps: 813 pkts/sec
  - 10 Gbps: 812744 pkts/sec

But...


TCP/IP
PPPoE
AppleTalk
Proxies
Ethernet
Ethernet ID: 00:1b:63:92:9f:bb
Configure: Manually (Advanced)
Speed: 1000baseT
Duplex: full-duplex
Maximum Packet Size (MTU): Standard (1500)
Jumbo (9000)
Custom: 9000 (Range: 72 to 9000)
Caution: Setting MTU value above the standard ethernet setting (1500) may cause some routers to crash. Please check with your ISP before setting this value above 1500.
?

Jumboframes

- Lots of gigabit ethernet equipment supports larger packets: "jumboframes"
- Common value: ± 9000 bytes
  - but no standard non-standard size
- "Mini jumbos" / "baby giants" up to ± 2000 bytes common in lower-speed switches

IEEE 802.3as

- Increase of the framesize to 2000 bytes
  - this would allow for an IP MTU of 1982
- But: larger size is only used to for additional IEEE 802 headers and is *not* exposed to higher layers
  - IP MTU remains 1500 bytes

Big packet advantages

- More room for additional headers without path MTU discovery breakage
- Lower overhead, especially with large headers
- Less per packet work in hosts = faster
- Less per packet work in routers/switches = possible power/heat savings
- Better TCP performance

Disadvantages

- Routers/switches need more buffer memory for a given queue size
- More delay and jitter:
  - packets take longer to transmit, subsequent packets have to wait longer
- So only do 1500+ at 1000+ Mbps?

Path MTU Discovery

- Only triggered if both ends are > 1500
  - otherwise TCP MSS limits packet size
- Today, PMTUD black holes happen:
  - < 1500 MTU user has problems if *others* filter ICMP too bigs
- **Less of an issue with > 1500 MTU:**
  - > 1500 MTU user has problems if they filter ICMP too bigs *themselves*

Bit errors

- More data loss from bit errors:
  - one flipped bit takes out more data
    - so only use on low-BER links!
- More undetected bit errors?
  - naive: more errors/packet, but fewer packets = no difference
  - CRC32 Hamming distance = 4 up to 11.5k, bigger packets more vulnerable

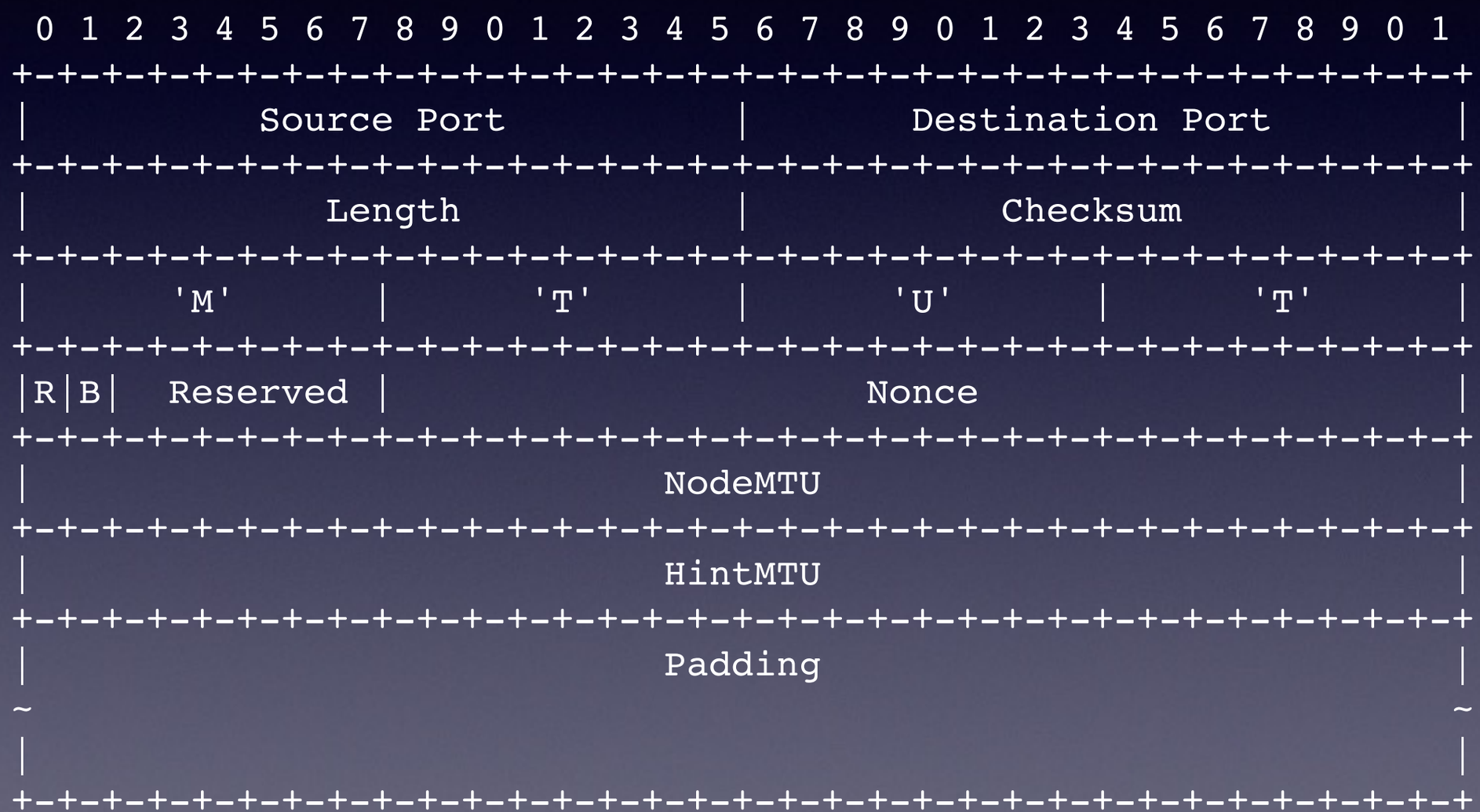
What we need

- Ability to turn on jumbos without touching all hosts on a subnet
- Take advantage of hardware improvements without protocol work
  - no more hardcoding of MTU sizes
- Be backward compatible!
  - also with current jumbo deployments

How?

- UDP protocol to:
  - discover neighbor's MTU size
  - see if that packet size works
  - if not, probe for a packet size that works
- Monitor sending/receiving of large packets
  - (similar to IPv6 neighbor unreachability detection or Shim6 REAP)

Packet format



Probing

- Discover capability/remote MTU with minimum size probe
- Establish upper bound quickly:
  - 320, 640, 1280, 2560, 5240, 10240, ...
- Then use hints:
  - 576, 1492, 1500, 1530, 1982, 2304, 4070, 8092, 9000, 16384, 32000, 64000

Changes since -03

- Rely less on changes to Neighbor Discovery
  - (although there is still a new ND option to indicate supported MTU)
- Allow for probing with a userspace daemon
  - started on an implementation!

Demo

Next steps

- My thinking, depending on interest, either:
  - adopt as an int-area wg draft, work on it here, publish standards track
  - finish implementation, write -05, publish as individual submission / AD sponsored experimental
- Chairs, wg, please chime in!

Thanks, all.

Questions?