Inferring and Debugging Path MTU Discovery Failures

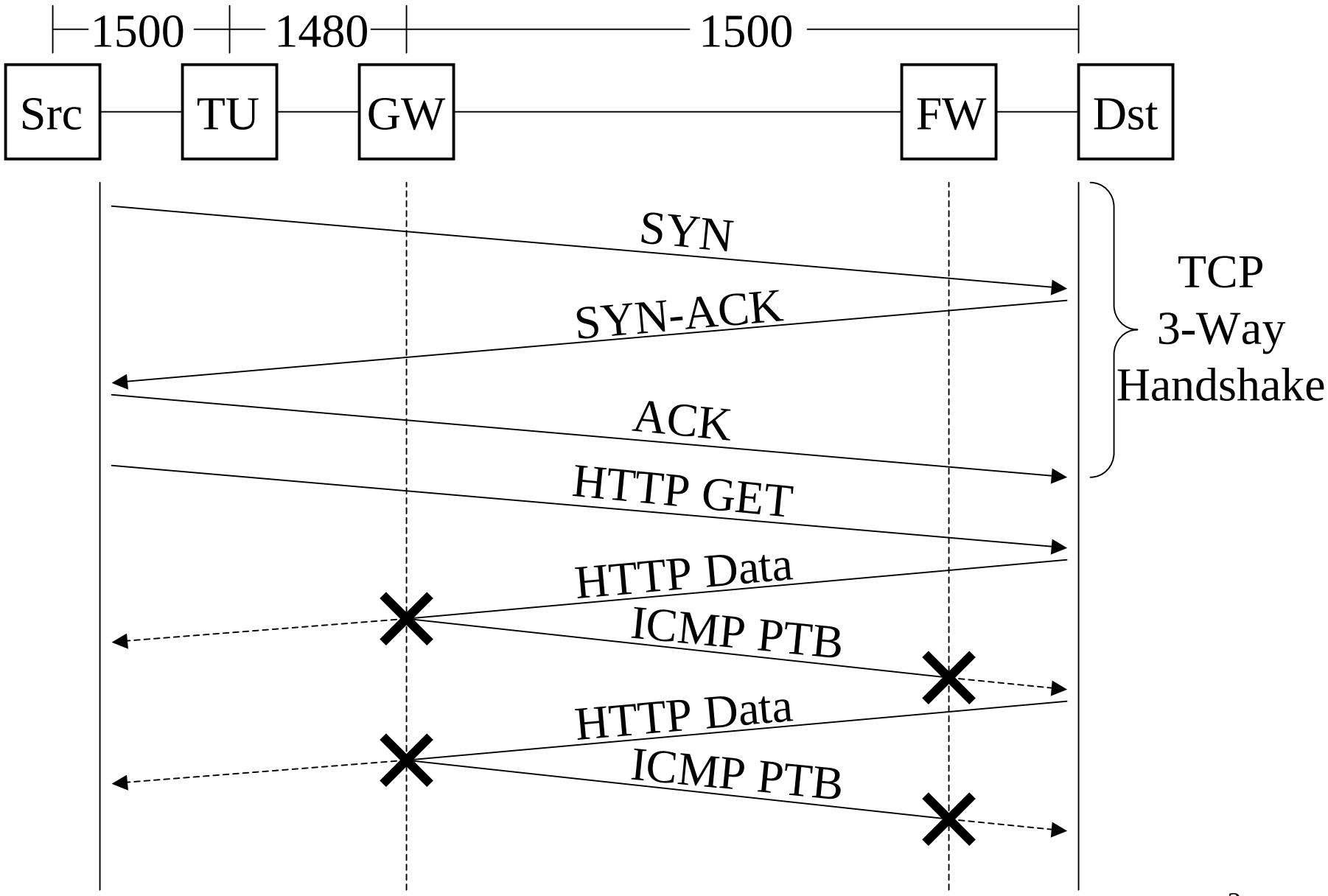
Matthew Luckie (U. Waikato)

Kenjiro Cho (IIJ/WIDE)

Bill Owens (NYSERNet)

The Problem

- It is desirable to send data in the fewest number of packets possible
  - Path MTU Discovery (PMTUD)
  - iterative process to determine the largest packet size (MTU) supported to a destination
  - uses feedback from ICMP Fragmentation Required / Packet Too Big (PTB) messages
- ICMP packets are not first class citizens
- PMTUD relies on these messages to work
  - Unreliable at best
  - New PMTUD method from IETF on the way


1500
1480
1500
Src
TU
GW
FW
Dst
SYN
SYN-ACK
ACK
TCP
3-Way
Handshake
HTTP GET
HTTP Data
ICMP PTB
HTTP Data
ICMP PTB

*

The Reverse-Path Problem

A. Medina, M. Allman, S. Floyd. (2005) Measuring the Evolution of Transport Protocols in the Internet

- 17% of 81776 targets failed at PMTUD
  - 35% of 500 ‘popular’ websites failed
  - Did not find any which tried smaller packets
  - Assumed middle-boxes filtering ICMP
- 41% of targets did PMTUD successfully
- 30% did not attempt PMTUD

Contribution of this Work

- A forward-path PMTUD debugging tool
  - infers the hop where large packets are discarded without the source receiving a PTB message
  - largest packet that can be forwarded through
  - uses a traceroute-like method
- A look at the problems we found when measuring targets on networks which peer with the jumbo-capable Internet2

Debugging Technique: Stage 1 of 2

- Begin with a traceroute using small packets
  - Infer the forward path
  - So we can later distinguish between all packets being silently discarded and just the large ones:
    - Determine which hops will send ICMP feedback (Time Exceeded) to small packets
    - Ensure that packets can actually reach a destination

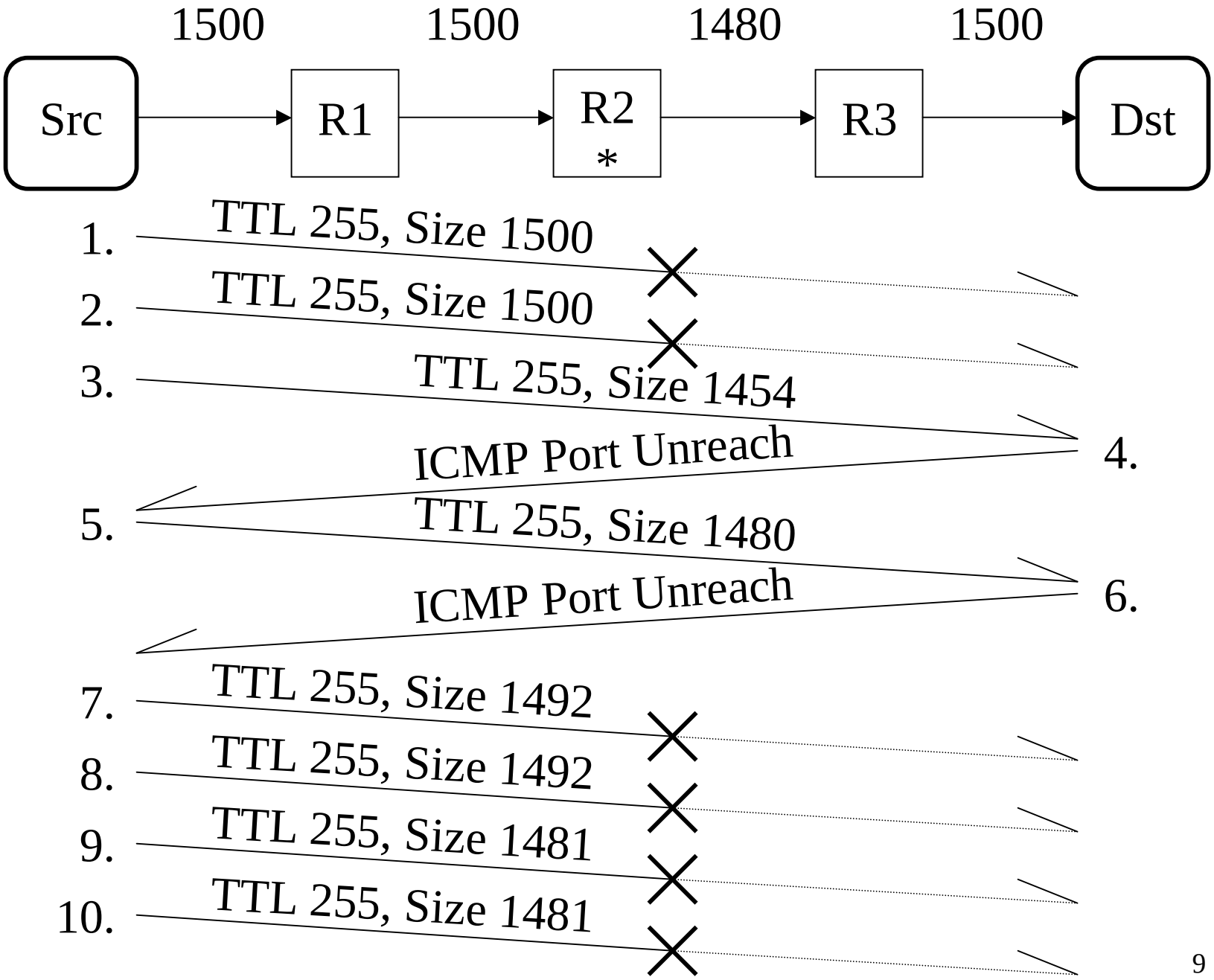
Debugging Technique: Stage 2 of 2

- Determine the Path-MTU
  - start with the outgoing interface's MTU
  - for each PTB message, reduce the working Path-MTU value until we reach the destination
  - we do an MTU search if
    - large packets are silently discarded
    - if we get a PTB message with a next-hop MTU of zero or larger than the probe we sent
  - we do a TTL search to infer the hop that we don't receive a PTB message from

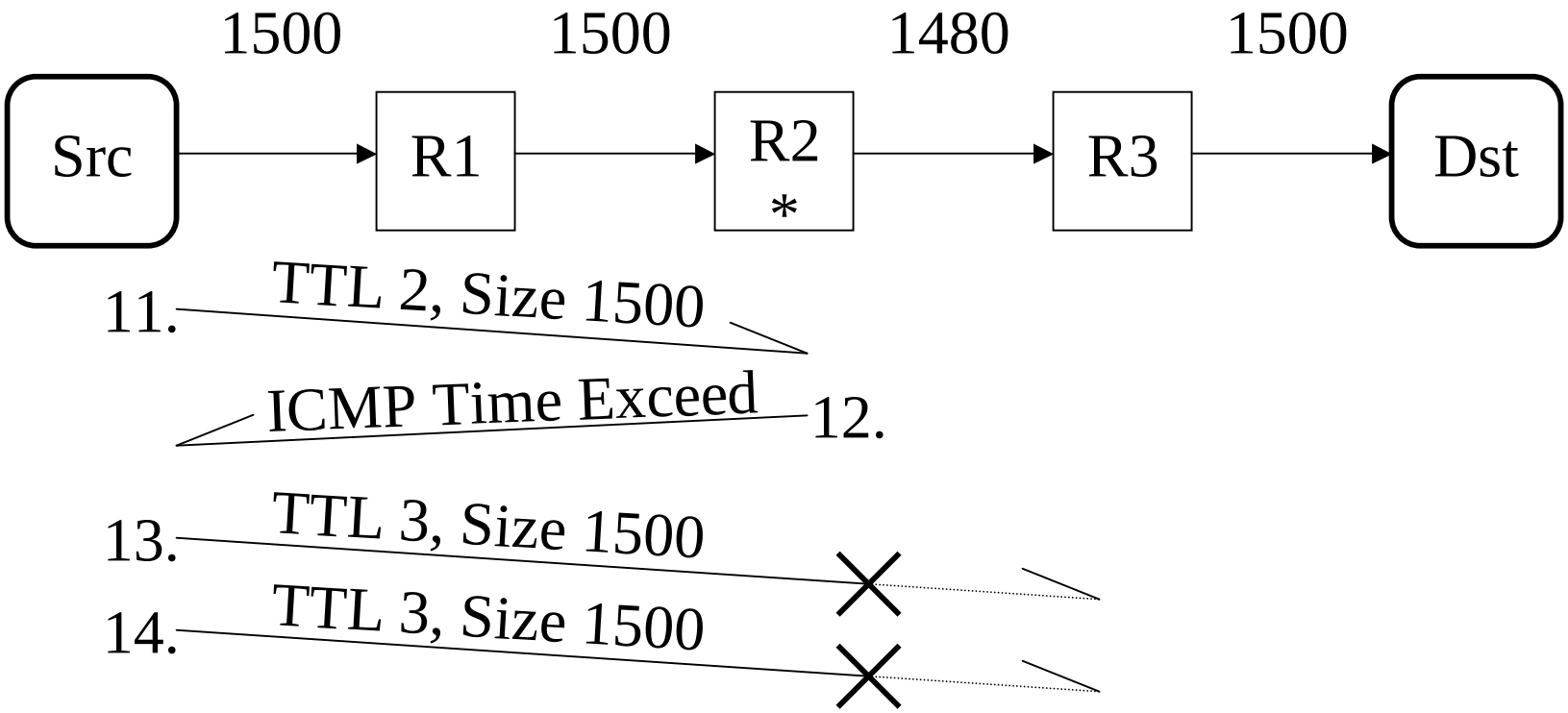
*

The MTU Search

- Define
  - lower bound: largest packet to get a reply
  - upper bound: smallest packet to not get a reply
- In practice, a binary search is not suited
  - MTU values tend to cluster around fairly limited numbers of media MTUs
  - Each probe that is silently discarded incurs two five-second timeouts (by default)
  - Cheaper to send a packet that gets ICMP feedback than one that does not
  - Use a table of MTU values to guide the search
- We use a number of heuristics to guide the search, see the paper for complete coverage


1500
1500
1480
1500
Src
R1
R2
*
R3
Dst
1.
TTL 255, Size 1500
2.
TTL 255, Size 1500
3.
TTL 255, Size 1454
4.
ICMP Port Unreach
5.
TTL 255, Size 1480
6.
ICMP Port Unreach
7.
TTL 255, Size 1492
8.
TTL 255, Size 1492
9.
TTL 255, Size 1481
10.
TTL 255, Size 1481

*


1500
1500
1480
1500
Src
R1
R2
*
R3
Dst
11.
TTL 2, Size 1500
ICMP Time Exceed
12.
13.
TTL 3, Size 1500
14.
TTL 3, Size 1500

*

Methodology

- Two IPv4 hosts with 9000-byte MTU Interfaces
  - connected to networks that peer with Internet2
  - east.nysernet.org
  - nms1-chin.abilene.ucaid.edu
- 147 NLANR AMP targets
  - all with 1500-byte MTU interfaces
  - vast majority are hosted on networks that peer with I2
- April 28th 2005, 21:50 EDT

Results

	NYSERNet	nms1-chin	Intersection	Total
Target Count:	147	147	147	-
Reachable:	136 (93%)	134 (91%)	134	-
Failures:	41 (30%)	40 (30%)	25	-
No ICMP Messages:	6 (6)	5 (5)	4 (4)	7 unique
No PTB Messages:	26 (17)	27 (18)	13 (13)	22 unique
Incorrect PTB:	2 (2)	2 (2)	2 (2)	2 unique
Target MTU Mismatch:	7 (7)	6 (6)	6 (6)	7 unique

The number on the left is the number of AMP targets on a path with this failure mode.
The number in brackets is the number of unique failure points.

*

Results: No ICMP Messages

- 7 failures (6 x 1500, 1 x 1536)
- Two were due to ingress filters
  - one originated ICMP with 127.0.0.1
  - another originated ICMP with RFC 1918
- Another due to an ‘Internet Free Zone’
- Another due to routing issue that allowed end-to-end connectivity, but routers in the forward path had no route back to our source.

*

Results: No PTB Messages

- 22 hops sent TTL Expired, but no PTB messages
  - 16 x 1500
  - (4 x 4472, 2 x 4540, 1 x 4470, 1 x 2002)
- Some repetition in source of the problem, 20 distinct problem locations
  - Obtained technical diagnosis for seven
  - Two were upgraded before diagnosis could be obtained
- Two main causes:
  - no ip unreachables (does not suppress TTL Expired)
  - MTU Mismatches

Incorrect PTB Messages

- Two hops from one location sent a PTB message with an incorrect next-hop MTU
  - We sent 9000 byte probes
  - It said Packet Too Big: send 4586 byte packets
  - But the path to the next-hop could only carry packets up to 4472 bytes in size
- an MTU mismatch

Target MTU Mismatches

- We found 7 AMP machines were plugged into a subnet with a router which forwarded packets larger than 1500 bytes
  - An MTU mismatch with the router, as these machines (strictly speaking) can't receive packets larger than 1500 bytes.
  - Two did: one managed 1506 bytes, another managed 2016 bytes.

An Anecdote

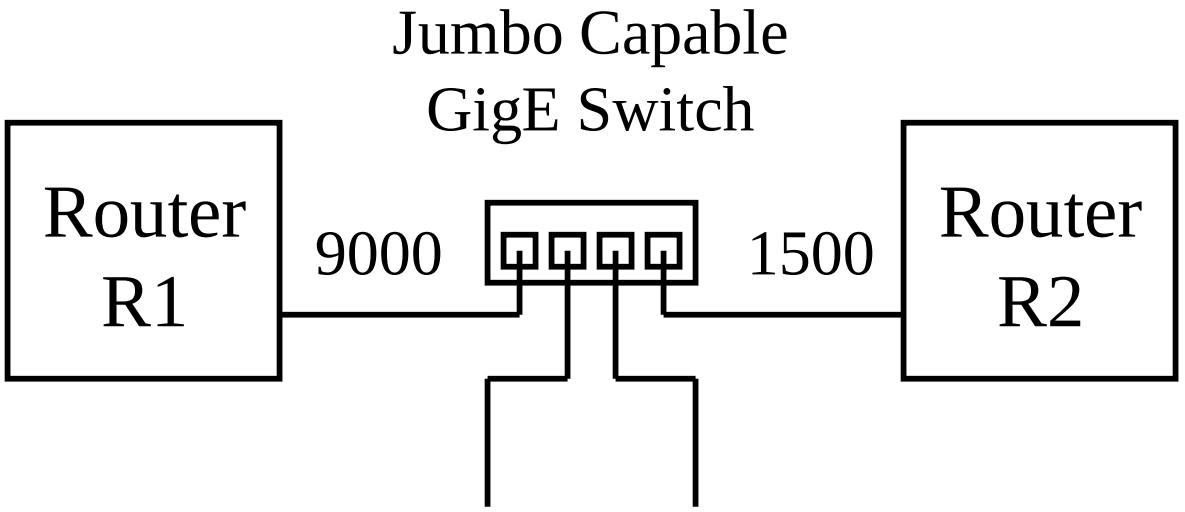
- A router in a commercial ISP in NYC sends PTB messages with a next-hop MTU of 4470 bytes
  - For all packets larger than 4458 bytes!
  - That's a 12 byte discrepancy
  - Could be related to 3 4-byte MPLS labels being appended.
  - Could be mis-configuration
  - Could be a bug in the router
- would really like to know why

Acknowledgements

- Matt Zekauskas (Internet2) collected nms1-chin
- WIDE + CAIDA funded development of scamper
- NLANR/MNA is funded by NSF ANI-0129677

http://www.wand.net.nz/scamper/

MTU Mismatch Example



- Router R1 thinks it can send 9000 byte packets to R2, which can only receive 1500
- Drop happens at the switch, where no ICMP can be sent.