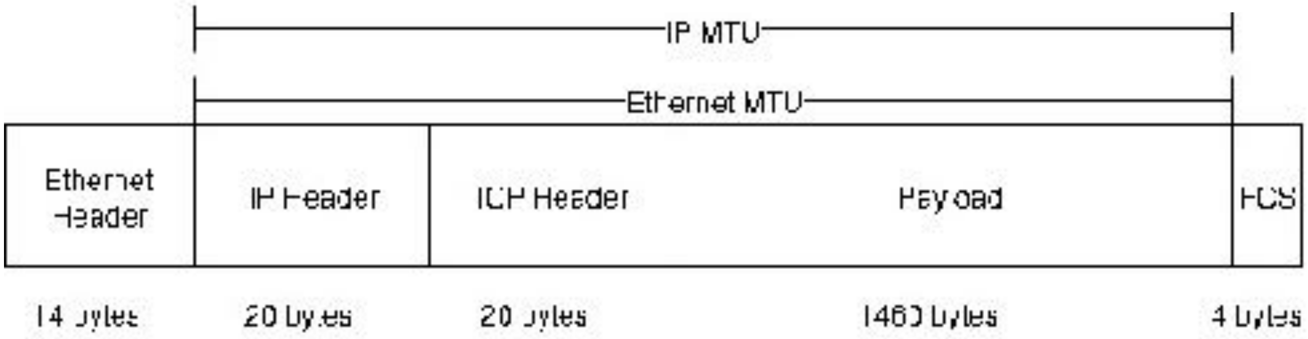
Exploring usable Path MTU in the Internet

Ana Custura
Gorry Fairhurst
Iain Learmonth

University of Aberdeen

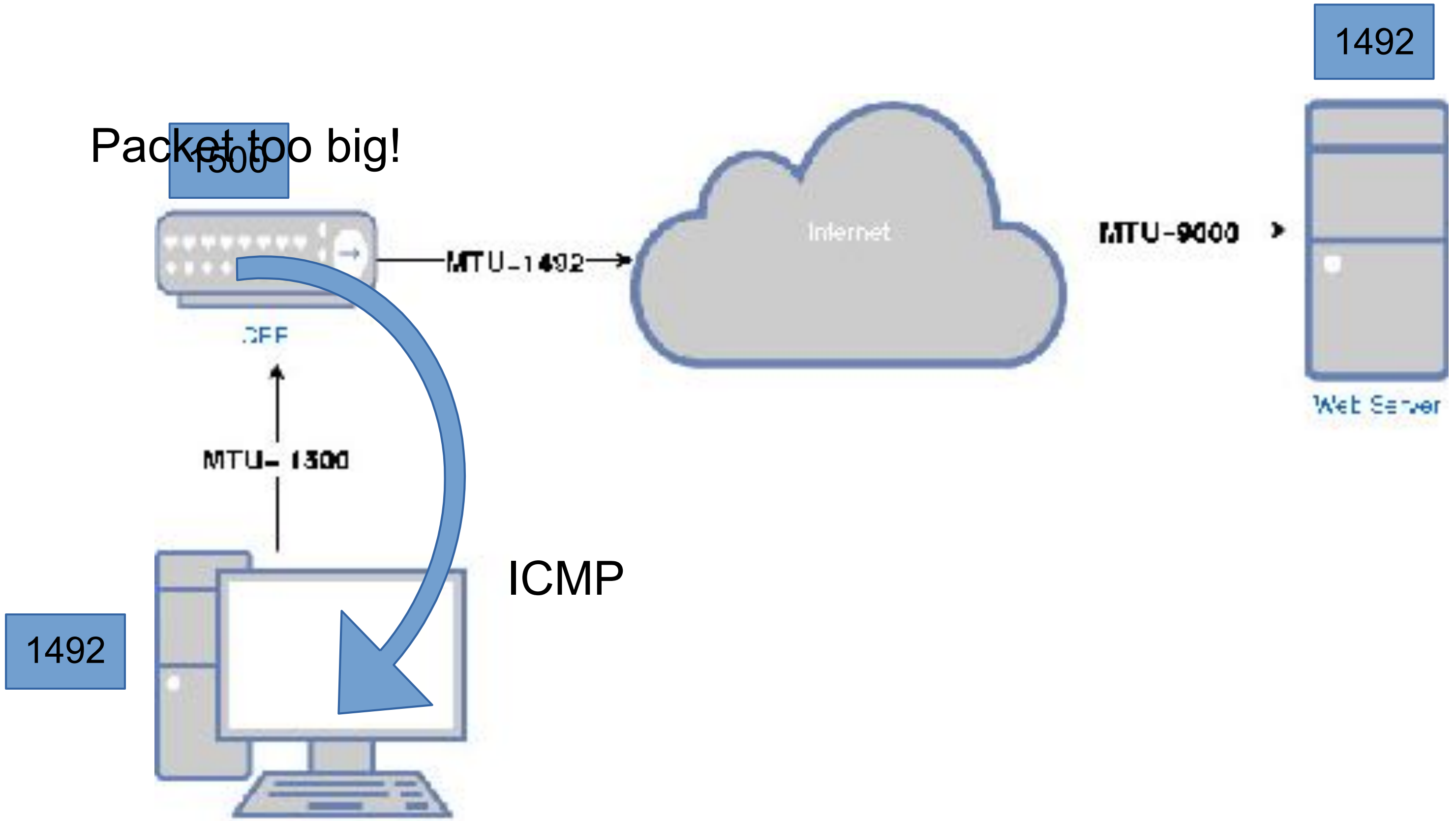
Endpoints need to know how large a packet they can send

- MTU = Maximum transmission unit = Largest size of packet that a link can forward
- Sender does not know the largest size of packet it can send on a specific Internet path
- Path MTU = MTU of an Internet path
- Small packets means more overhead
- Large packets may exceed link MTU



What is Path MTU Discovery?

- PMTU Discovery (PMTUD) = network layer mechanism to determine the PMTU using ICMP



Packet too big!

ICMP

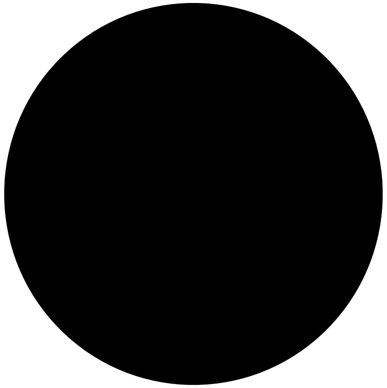
MONROE

Electronics Research Group

mami

Avoiding black holes

- Small packets are inefficient…
- But, PMTUD considered unreliable
  - ICMP firewalls, CPE
  - ECMP (and others) make ICMP unreliable



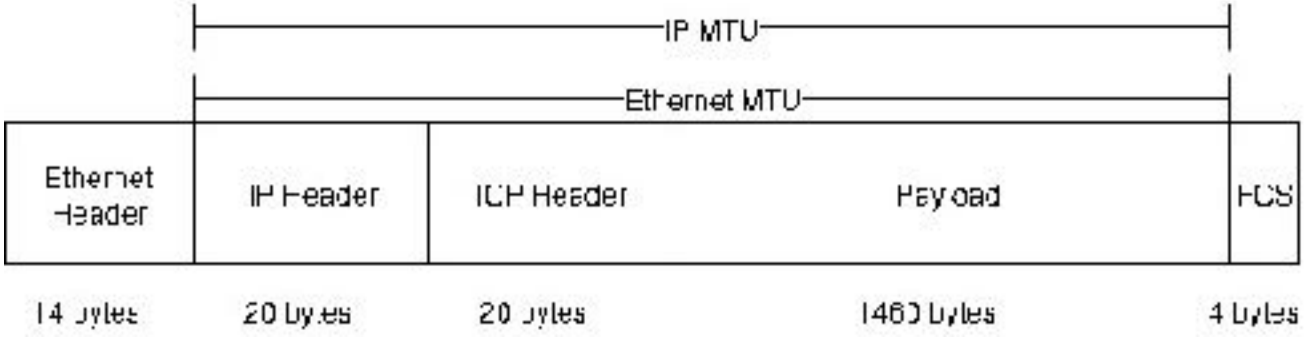
MONROE

Electronics Research Group

mami

Avoiding black holes

- PLPMTUD – exists, not yet enabled
- Other black-hole detection mechanisms
- TCP MSS



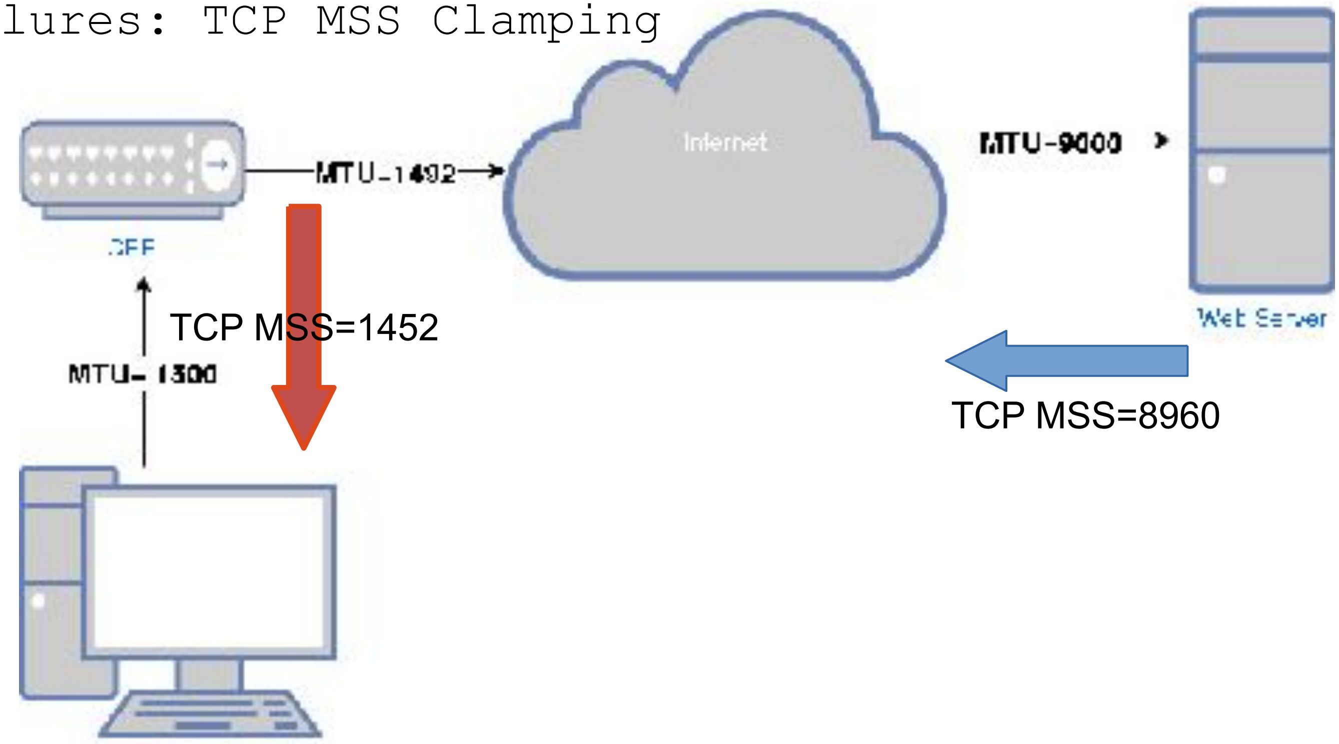
IP MTU

Ethernet MTU

Ethernet Header	IP Header	TCP Header	Payload	FCS
14 bytes	20 bytes	20 bytes	1460 bytes	4 bytes

What is TCP MSS?

- Maximum Segment Size (MSS) = A TCP option to advertise remote link MTU
- Middleboxes can change TCP MSS option to avoid PMTUD failures: TCP MSS Clamping



TCP MSS=1452

TCP MSS=8960

Tools and datasets

Purpose	**Tool used**	**Dataset name**
Collect server advertised MSS	PATHspider	A.1 “PATHspider”
Validate server advertised MSS	Ping	A.2 “Ping”

Server advertised MSS- "PATHspider"- IPv4

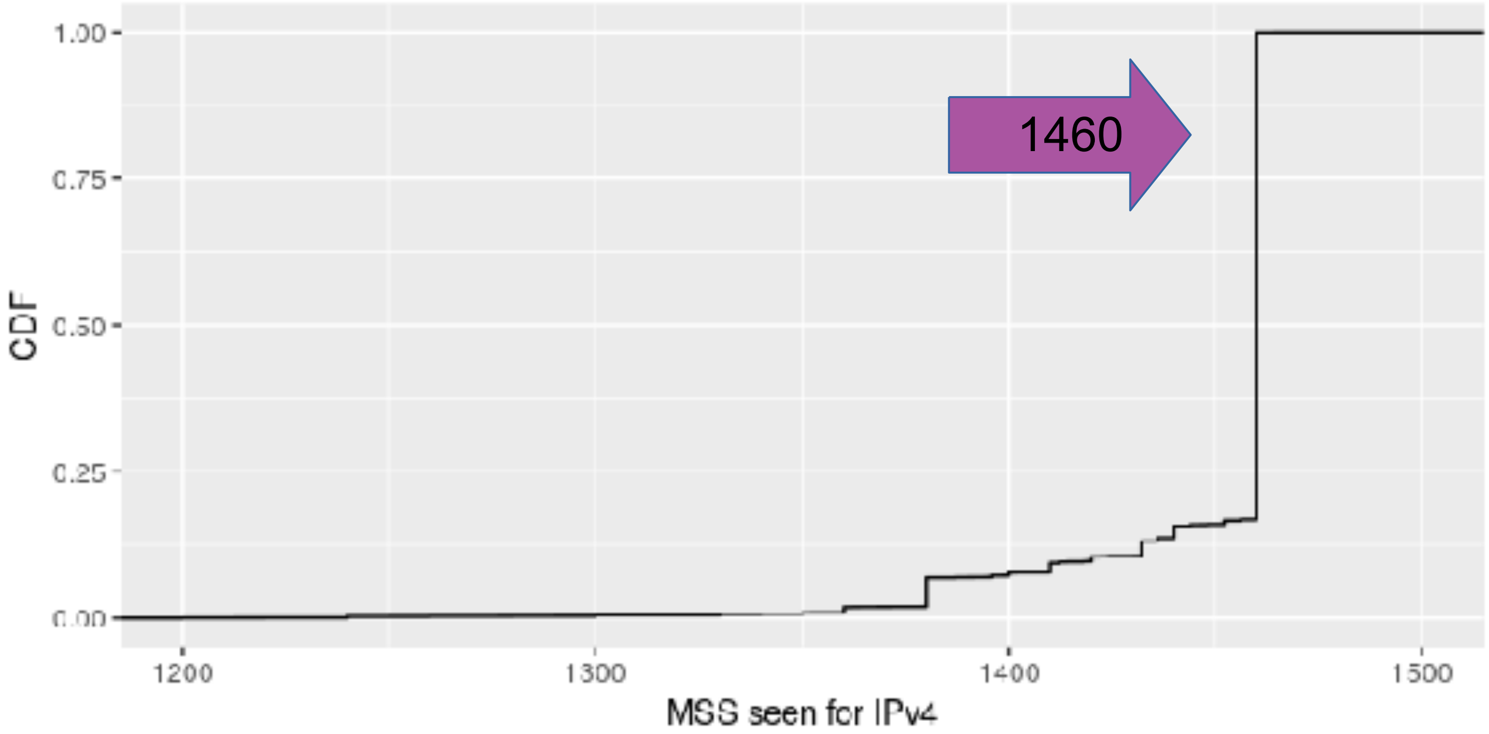


Figure: Avertised MSS (in bytes) on TCP SYN/ACK server response seen at Janet academic network

Server advertised MSS- “PATHspider”- IPv6

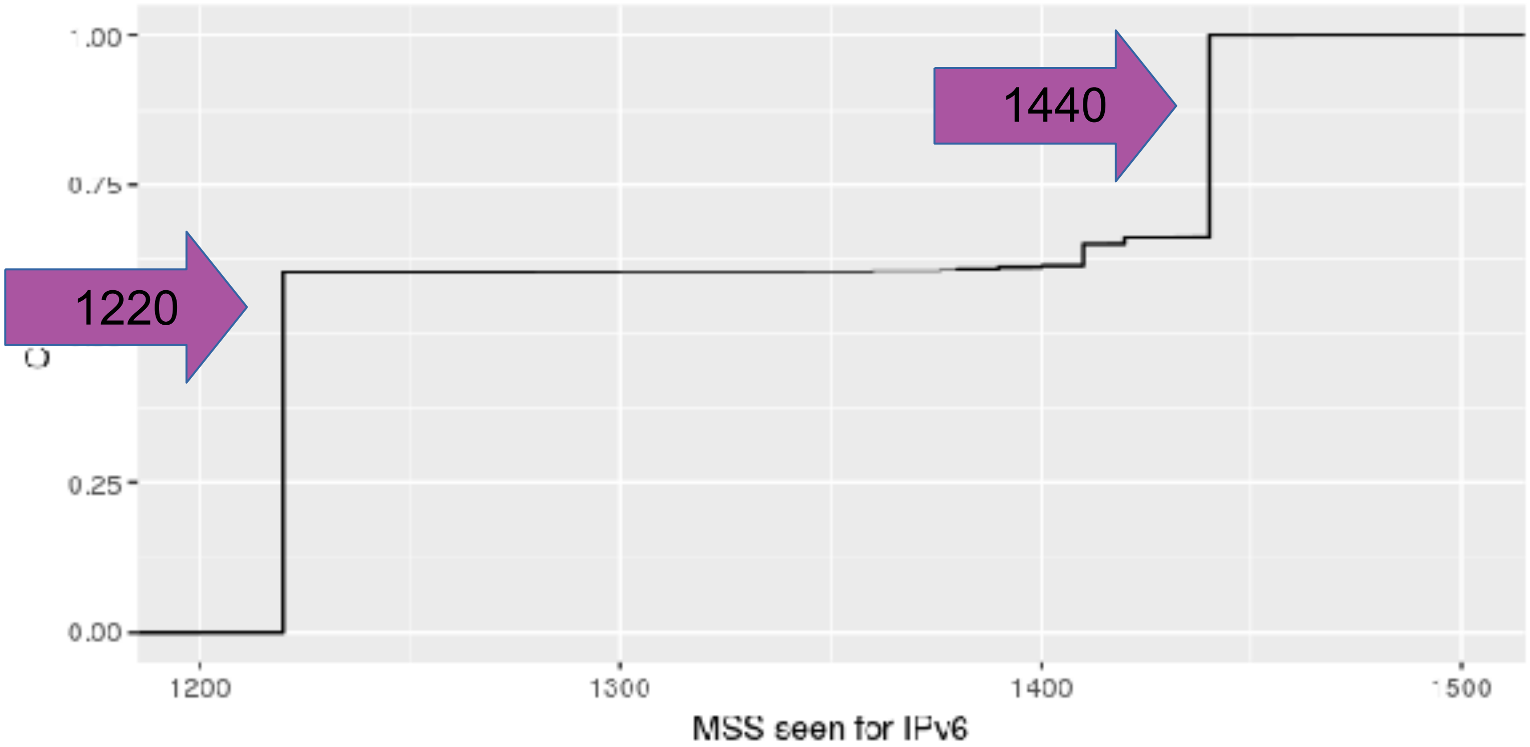


Figure: Avertised TCP MSS (in bytes) on TCP SYN/ACK server response seen at Janet academic network

Server advertised MSS and “Ping” results

- For 295,000 PATHspider targets: We sent a probe was the size of the advertised TCP MSS
- We also sent an ICMP probe to see if the target can be reached with a 1500B packet(A.1 ”Ping”)
- Of the subset that advertised MSS < 1460B (34,920), **93% were reached with a 1500B probe.**

Tools and datasets

Purpose	Tool used	Dataset name
Collect server advertised MSS	PATHspider	A.1 “PATHspider”
Validate server advertised MSS	Ping	A.2 “Ping”
Collect wireless/mobile client advertised MSS	Pathtrace	B.1 “MONROE”
Collect wired edge client MSS	RIPE Atlas Traceroute	B.2 “RIPE”

Client advertised MSS – Mobile edge results

- Dataset B.1, “MONROE”, consists of traceroute-style measurements collected from the MONROE platform
- A total of 888 hops (21%) returned an MSS Option
- TCP MSS Clamping can reduce the MSS to allow for headers added by a tunnel

Network	**Inserted MSS option**
Telenor Norway	1410 bytes
Telia Sweden	1400 bytes
Vodafone Italy	1400 bytes
Wind Italy	1420 bytes

Table: Inserted MSS options by mobile network, n = 10 paths

Client advertised MSS - Wired edge results

- TCP traceroute from 3000 RIPE Atlas probes towards our server (Dataset B.2, "RIPE")
- 4.8% of probes arrive carrying an MSS option, some larger than allowed by standard Ethernet
- 764 of the MSS values (23%) in received probes differed from the sent value of 1460
- Some box in the network is "trying" to help!

Tools and datasets

Purpose	Tool used	Dataset name
Collect server advertised MSS	PATHspider	A.1 “PATHspider”
Validate server advertised MSS	Ping	A.2 “Ping”
Collect wireless/mobile client advertised MSS	Pathtrace	B.1 “MONROE”
Collect wired edge client MSS	RIPE Atlas Traceroute	B.2 “RIPE”
Explore server PMTUD	Scamper	C.1 “Scamper”
Explore client PMTUD	Netalyzr Traceroute	C.2 “Netalyzr”
Inspect ICMP quotations	Pathtrace	D “ICMP”

Client PMTU - Mobile edge results

- We sent a 1500 byte UDP probe to our server with the DF flag set on 10 paths
- 16 mobile operators were tested from over 40 vantage points using the MONROE platform (Dataset C.2 - “Netalyzr”)
- Both experiments consistently reported a PMTU of 1500 bytes

MTU in the Internet - IPv4

- 60k Cisco Umbrella domains - Dataset C.1, “Scamper”

	1420 MTU	576 MTU	576 Black-hole
PMTUD too small	7.45%	3.7%	0.95%
PMTUD success	68.2%	63.9%	8.2%
PMTUD failure	16.4%	19.5%	67.4%
No DF set	12.5%	12.3%	15.2%
Clear DF	2.7%	4.1%	NIL

- 68% for IPv4 servers succeed in performing PMTUD
  - Up to 20% failed for IPv4, twice the amount reported in 2010
  - Over 10% did not attempt PMTUD (no DF)

MTU in the Internet - IPv6

- 60k Cisco Umbrella domains - Dataset C.1, “Scamper”

	1280 MTU	1280 Black-hole
PMTUD too small	59.6%	53.1%
PMTUD success	95.5%	32%
PMTUD failure	4.5%	67.9%

- 95% tested IPv6 succeeded in performing PMTUD
- ..but 60% of webservers did not attempt it
- 68% IPv6 and 76% IPv4 webservers failed PMTUD when local messages were blackholed

Discussion

- 60% of IPv6 hosts were configured to advertise a TCP MSS corresponding to the minimum IPv6 MTU.
  - Many servers artificially lower their MSS
  - MSS clamping in the network also common - in both mobile and wired edge
  - A smaller MSS prevents PMTUD working for TCP

Conclusion and next steps

- People do not trust PMTUD - they probably fear black holing their data, and use a lower MSS
    - Not helped by current PMTUD implementation problems
    - PLPMTUD *could* help, but not enabled/tested
    - Lowering the MSS only works for TCP
- Growing interest in transports using UDP
    - DPLPMTUD being developed to provide a robust PMTUD for UDP.
- We are also expanding our measurement set

This work is funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 644399 (MONROE). It used tools supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 688421 (MAMI).

Answers