Inferring and Debugging Path MTU Discovery Failures

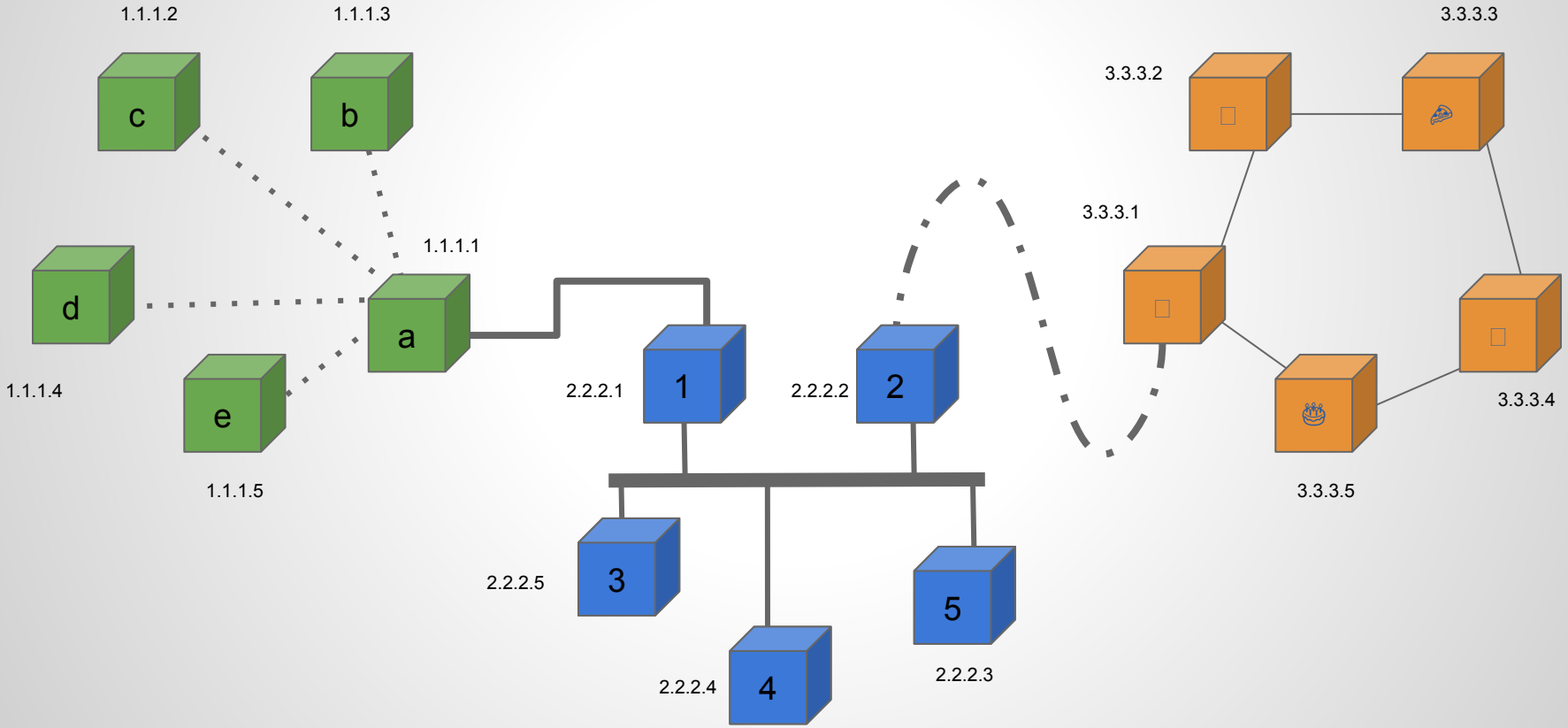
Matthew Luckie
University of Waikato

Kenjiro Cho
Internet Initiative Japan

Bill Owens
NYSERNet

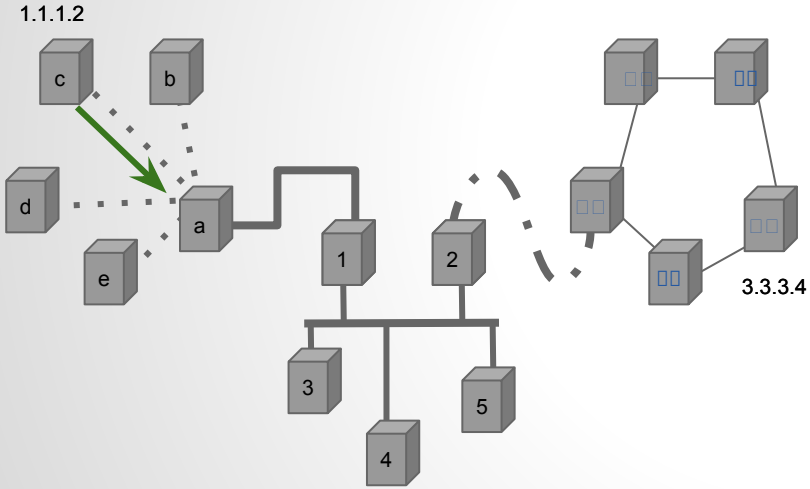
(https://www.usenix.org/legacy/event/imc05/tech/full_papers/luckie/luckie.pdf)

Background: The Internet



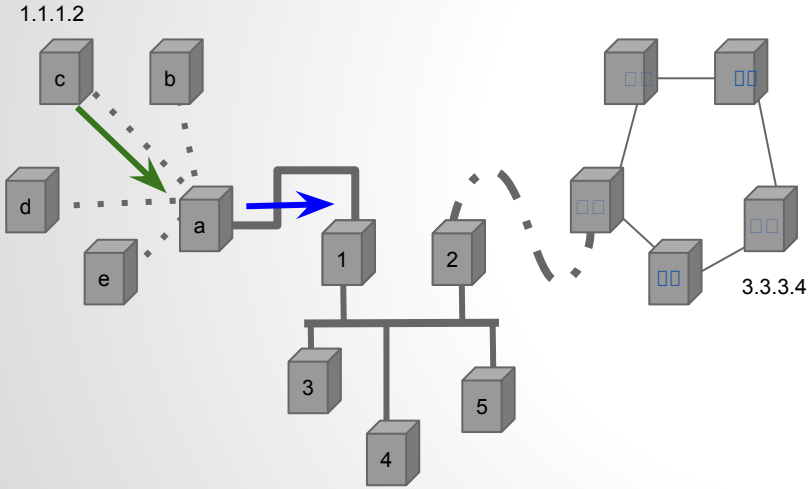
What Could Possibly Go Wrong?

Background: Path MTU




```
|src-------||dst-------|
[0x00][0x0c][0x00][0x0a]

|type||size------------|
{IPv4}[0x01][0x00][0x07]

|src-ip----------------|
[0x01][0x01][0x01][0x02]

|dst-ip----------------|
[0x03][0x03][0x03][0x04]

|type------||size------|
[0x01][0x00][0xFF][0xFF]
```

Background: Path MTU




```
|src-||dst-||type||size|
[0x01][0x05]{IPv4}[????]

|src-ip----------------|
[0x01][0x01][0x01][0x02]

|dst-ip----------------|
[0x03][0x03][0x03][0x04]

|type------||size------|
[0x01][0x00][0xFF][0xFF]
```

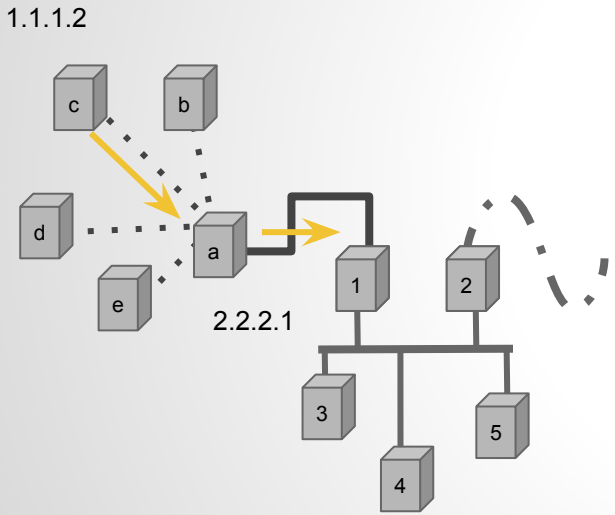
Fragmentation Considered Harmful

Christopher A. Kent
Jeffrey C. Mogul

December, 1987

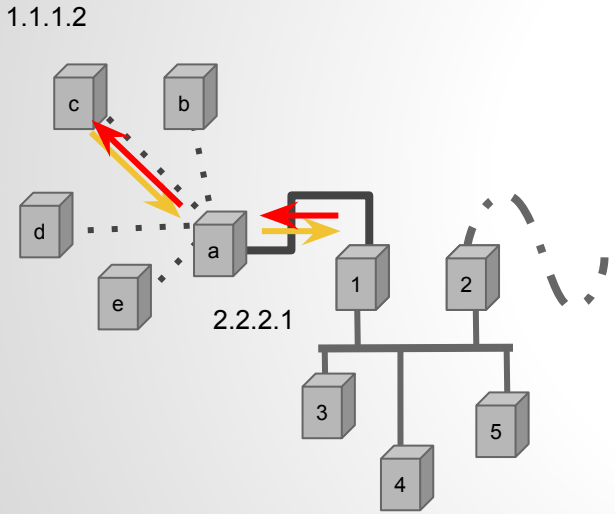
(http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-87-3.pdf)

Background: Path MTU Discovery



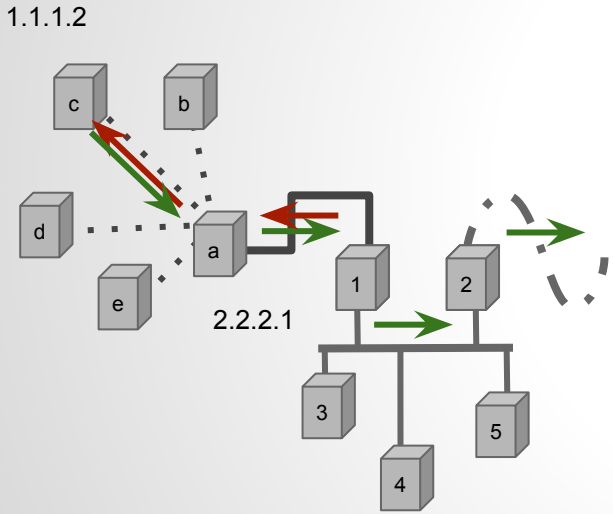
src ip	dst ip	type	payload
1.1.1.2	3.3.3.4	TCP	(0x0000 - 0xFFFF)

Background: Path MTU Discovery



src ip	dst ip	type	payload
1.1.1.2	3.3.3.4	TCP	(0x0000 - 0xFFFF)
2.2.2.1	1.1.1.2	ICMP	Too Big (max 0xFF)

Background: Path MTU Discovery



src ip	dst ip	type	payload
1.1.1.2	3.3.3.4	TCP	(0x0000 - 0xFFFF)
2.2.2.1	1.1.1.2	ICMP	Too Big (max 0xFF)
1.1.1.2	3.3.3.4	TCP	(0x0000 - 0x00FF)
1.1.1.2	3.3.3.4	TCP	(0x0100 - 0x01FF)
1.1.1.2	3.3.3.4	TCP	(0x0200 - 0x02FF)
1.1.1.2	3.3.3.4	TCP	(0x0300 - 0x03FF)
	...		

Inferring and Debugging Path MTU Discovery Failures

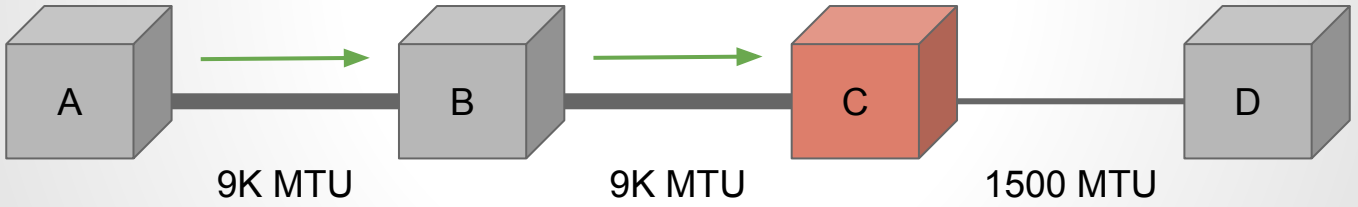
Matthew Luckie
University of Waikato

Kenjiro Cho
Internet Initiative Japan

Bill Owens
NYSERNet

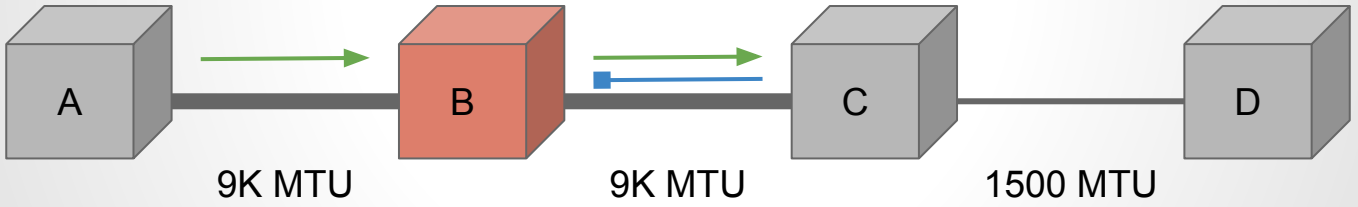
Expected Failure Modes

- Router (C) does not send ICMP Too Big
  - “For Security Reasons™”



Expected Failure Modes

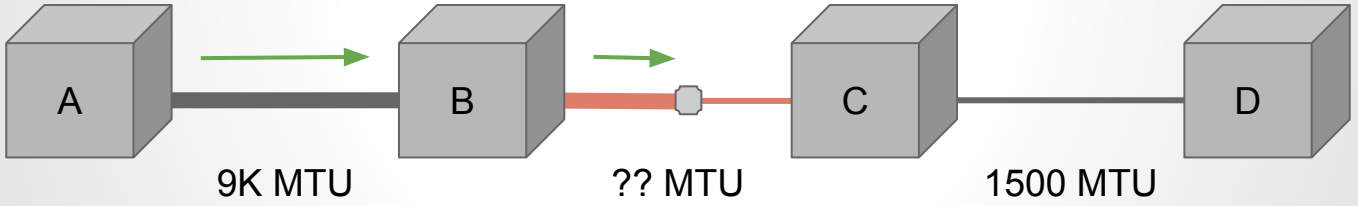
- Firewall (B) blocks ICMP Too Big
  - “For Security Reasons™”



* This was David’s problem

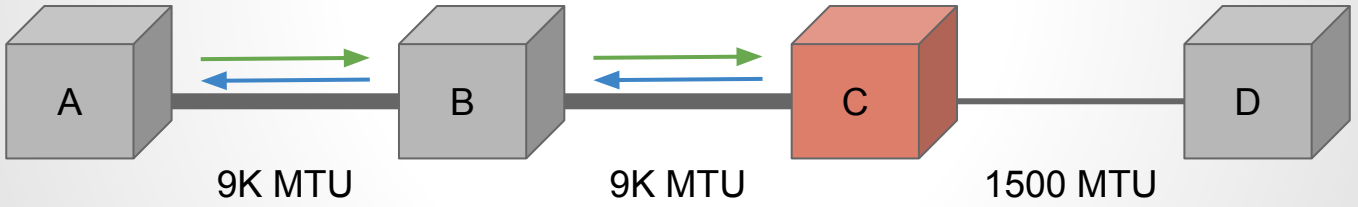
Expected Failure Modes

- Layer 2 MTU Mismatch
  - B thinks it can send jumbo frames to C
  - … but C is not *actually* capable of receiving them



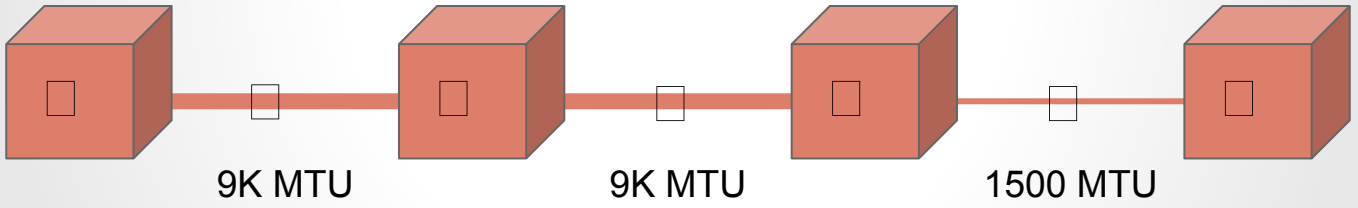
Expected Failure Modes

- Router (C) does not send a Next-Hop MTU
  - (A) has to guess, which is inefficient



Expected Failure Modes

- Implementation Bugs
  - … They’re a Thing



Scamper

- Traceroute w/ small packets
- Test with big packets
- “Brute force” Path MTU Discovery
- Traceroute w/ big packets to find faulty link

Results

Dataset:	NYSERNet-east	nms1-chin	Intersection	Total
Location:	New York, NY	Chicago, IL	–	–
Hostname:	east.nysernet.org	nms1-chin.abilene.ucaid.edu	–	–
Date / Time:	Apr 28 2005, 21:50 EDT	Apr 28 2005, 20:10 CDT	–	–
Target Count:	147	147	147	–
Reachable:	136 (92.5%)	134 (91.2%)	134	–
PMTUD Failures:	41 (30.1%)	40 (29.9%)	25	–
No ICMP messages:	6 (6 unique)	5 (5 unique)	4 (4 unique)	7 unique
No PTB messages:	26 (17 unique)	27 (18 unique)	13 (13 unique)	22 unique
Incorrect PTB messages:	2 (2 unique)	2 (2 unique)	2 (2 unique)	2 unique
Target MTU Mismatch:	7 (7 unique)	6 (6 unique)	6 (6 unique)	7 unique

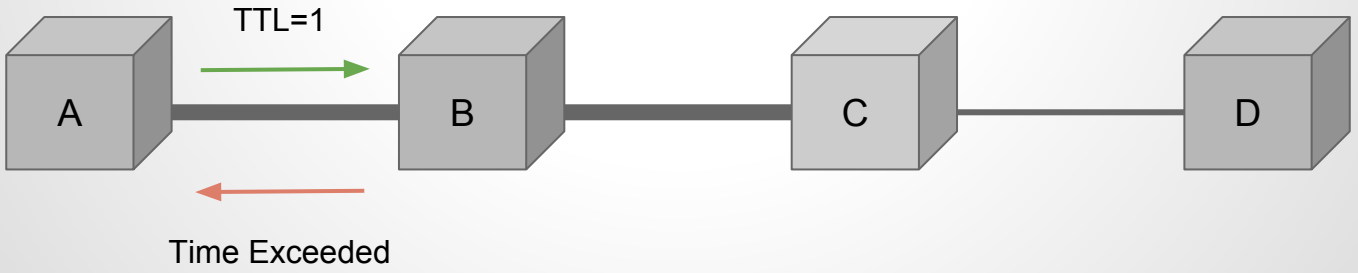
Table 1: Summary of the two data collections. 30% of reachable targets had a PMTUD failure.

Conclusions

- Ping (with small packets) is not enough
  - `tracepath` may be useful these days?
- Jumbo frames are coming
  - … sort of?
- New approach to PMTUD
  - Now RFC 4821 - “PLPMTUD”
  - Linux implements but it’s generally off by default
    - /proc/sys/net/ipv4/tcp_mtu_probing

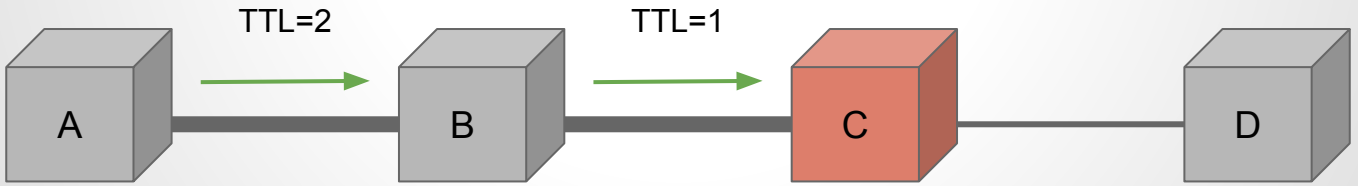
Debugging Techniques

1. Traceroute w/ Small UDP Packets
    a. Can we reach (D) at all?
    b. Find routers that don't send *any* ICMPs



Debugging Techniques

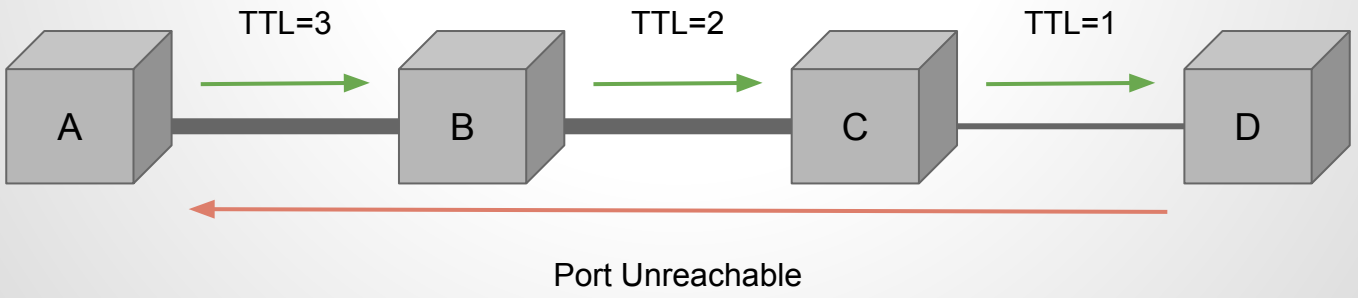
1. Traceroute w/ Small UDP Packets
    a. Can we reach (D) at all?
    b. Find routers that don't send *any* ICMPs



...

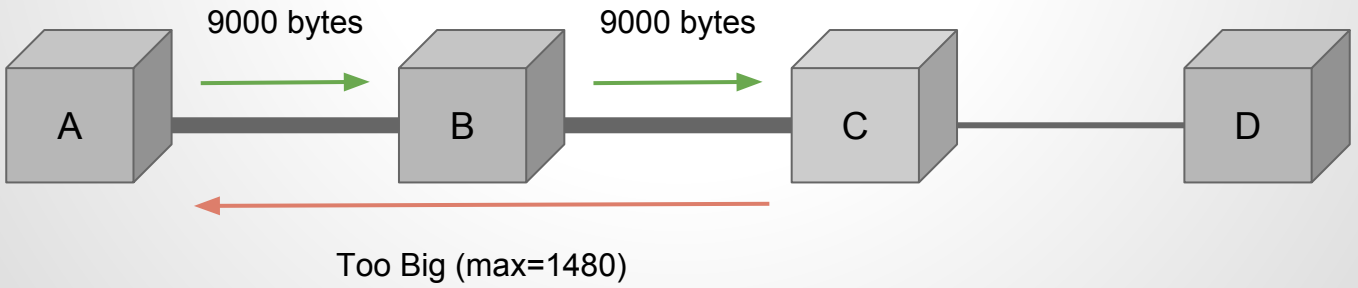
Debugging Techniques

1. Traceroute w/ Small UDP Packets
    a. Can we reach (D) at all?
    b. Find routers that don't send *any* ICMPs



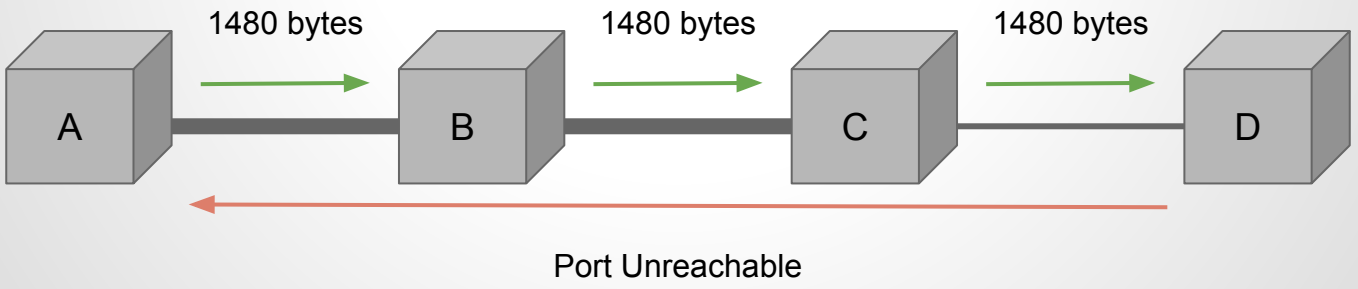
Debugging Techniques

2. Send Big UDP Packets
    a. Hopefully we get an ICMP Too Big
    b. If so, decrease size until port unreachable/timeout



Debugging Techniques

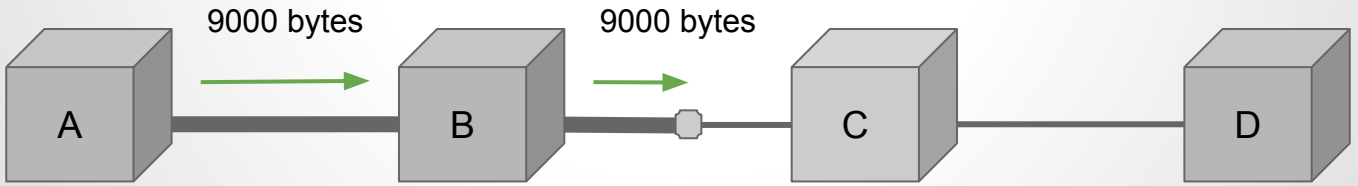
2. Send Big UDP Packets
    a. Hopefully we get an ICMP Too Big
    b. If so, decrease size until port unreachable/timeout



Debugging Techniques

3a. No Feedback?

a. Try to find the *actual* PMTU

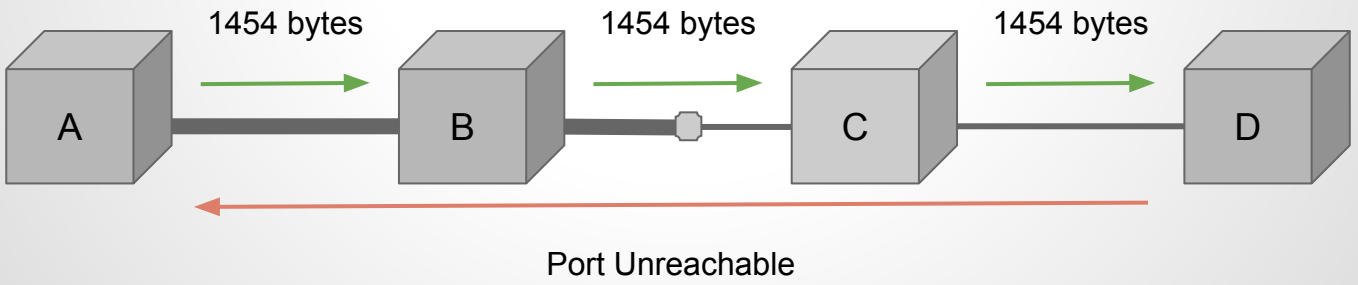


...

Debugging Techniques

3a. No Feedback?

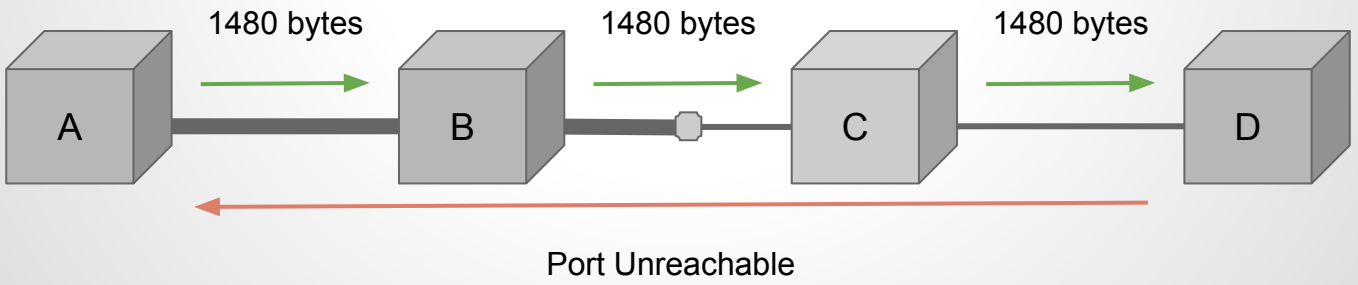
a. Try to find the *actual* PMTU
   i. Try the smallest (reasonable) PMTU



Debugging Techniques

3a. No Feedback?

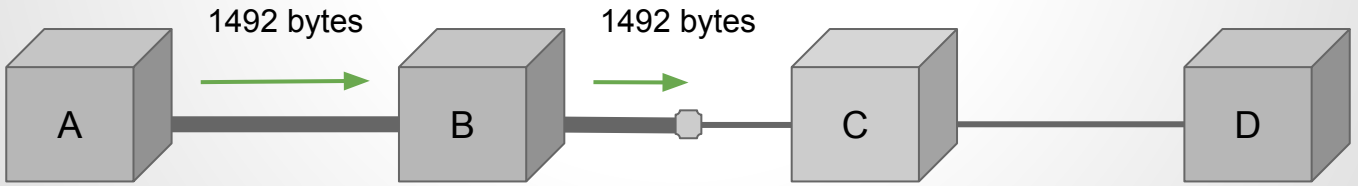
a. Try to find the *actual* PMTU
   i. Work your way up the chart...



Debugging Techniques

3a. No Feedback?

a. Try to find the *actual* PMTU
   i. … until you get no feedback again

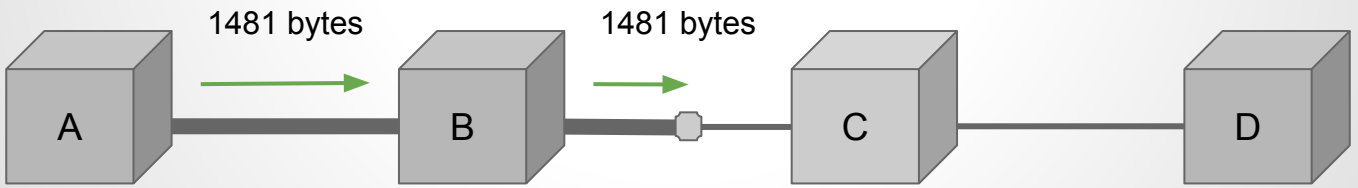


...

Debugging Techniques

3a. No Feedback?

a. Try to find the *actual* PMTU
   i. Then try candidate PMTU + 1 to be sure
   ii. (if not, resort to binary search)

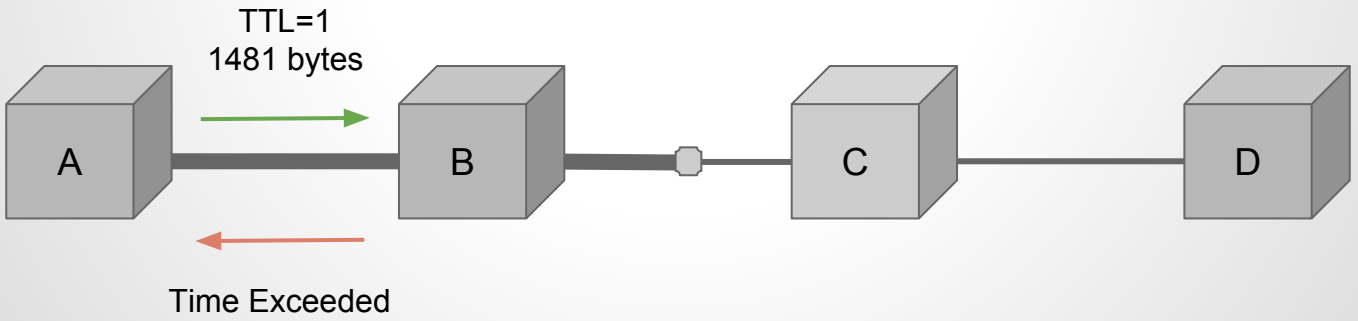


...

Debugging Techniques

3a. No Feedback

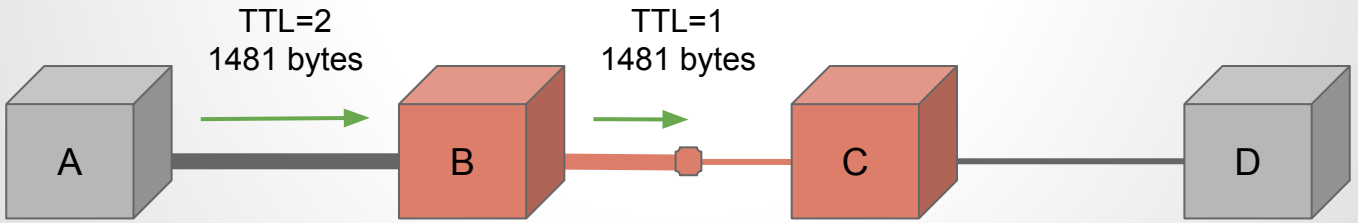
a. Try to find the *actual* PMTU [✓]
b. Try to find out where the problem is



Debugging Techniques

3a. No Feedback

a. Try to find the *actual* PMTU [✓]
b. Try to find out where the problem is

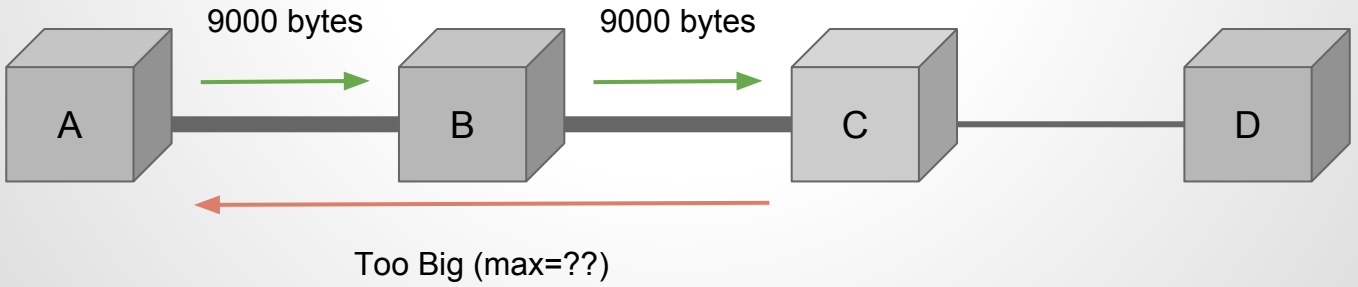


...

Debugging Techniques

3b. Invalid Feedback

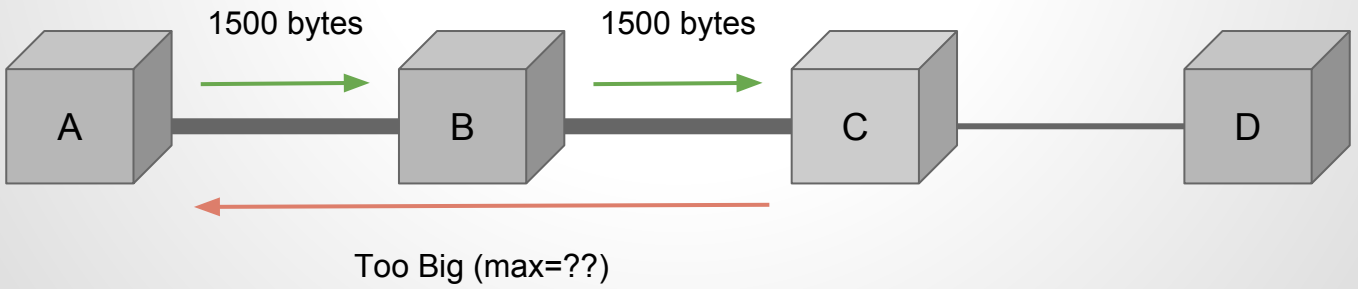
a. Try to find the *actual* PMTU



Debugging Techniques

3b. Invalid Feedback

a. Try to find the *actual* PMTU
   i. Working downwards now instead of upwards



Debugging Techniques

3b. Invalid Feedback

a. Try to find the *actual* PMTU
   i. Working downwards now instead of upwards