NAVAL
POSTGRADUATE
SCHOOL

A Middlebox-Cooperative TCP for a non End-to-End Internet

Ryan Craven (NPS / SPAWAR)
Robert Beverly (NPS)
Mark Allman (ICSI)

ACM SIGCOMM
19 Aug 2014

Support from:

TCP’s knowledge of end-to-end path conditions *a priori*

- ???
- ???
- ???
- ???
- ???

But TCP has questions…

- How fast can I send?
- How much should I send at once?
- Did the other end get my data?
    - Was a piece lost?
    - Was it in the right order?
    - Was it error-free?

…so it makes inferences

- How fast can I send?
- How much should I send at once?
- Did the other end get my data?
  - Was a piece lost?
  - Was it in the right order?
  - Was it error-free?

Congestion Control

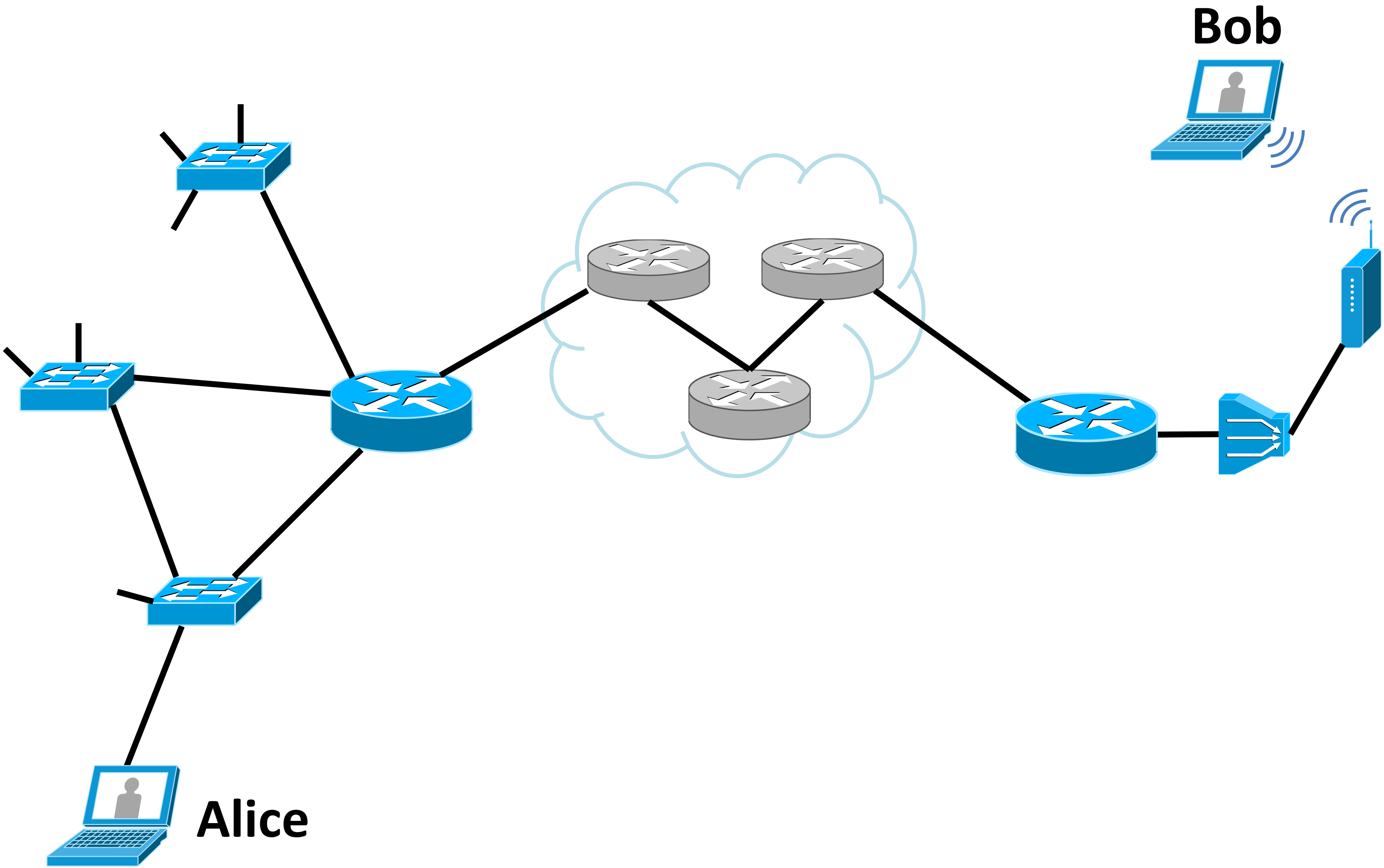
- Sequence Numbers
- Duplicate Acknowledgements
- Selective Acknowledgements

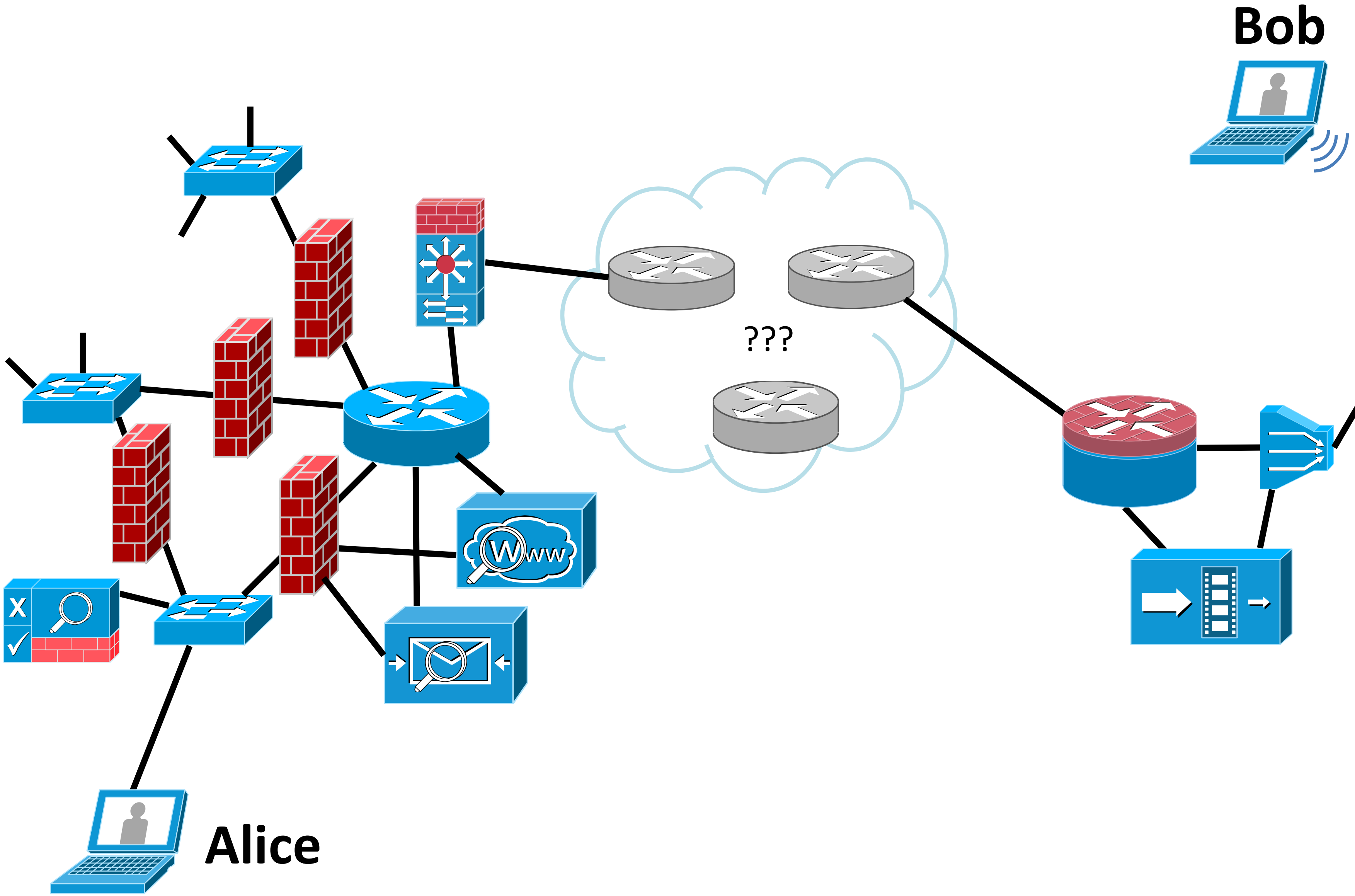
Checksums

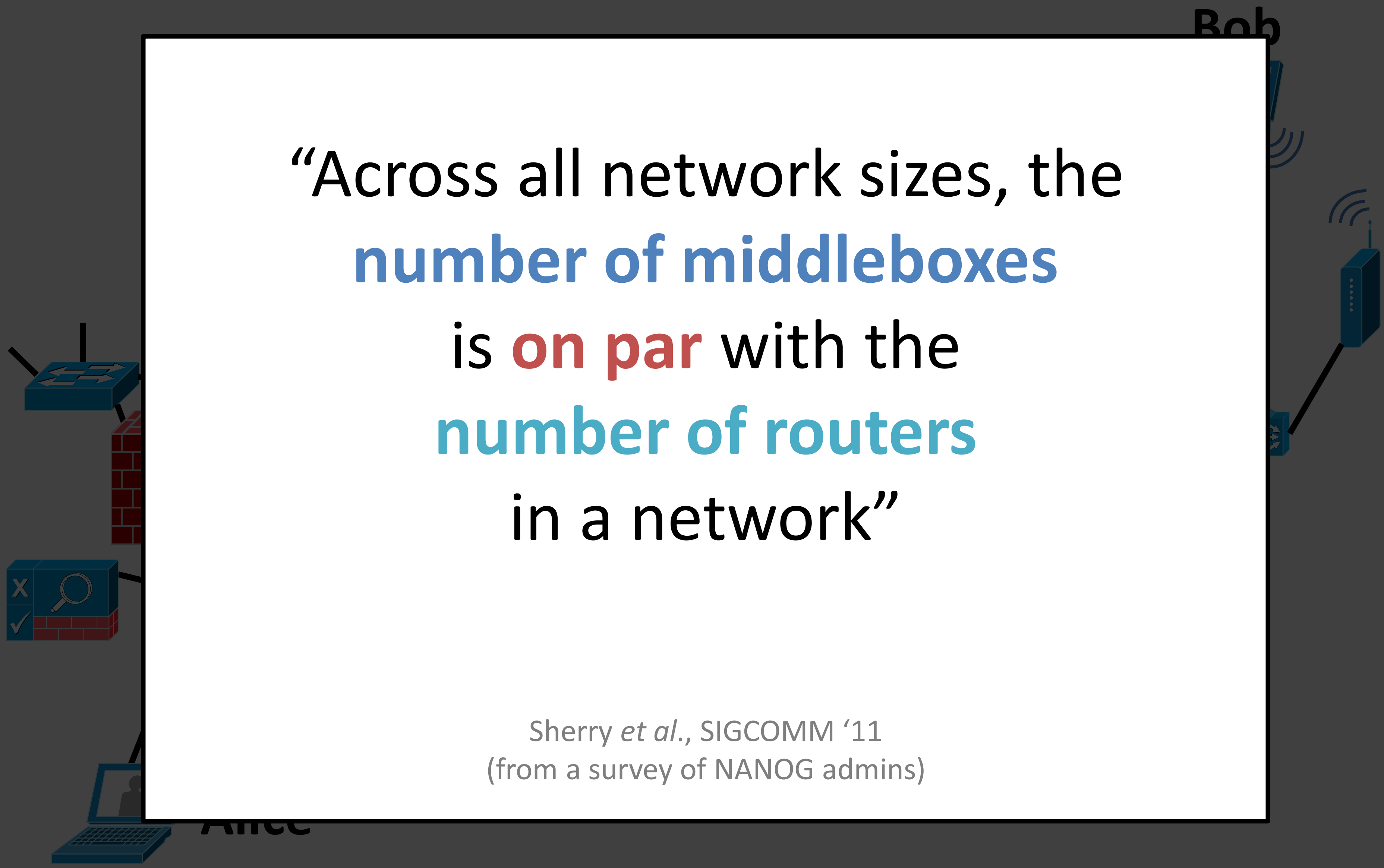
One more…

- How fast can I send?
- How much should I send at once?
- Did Bob get my data?
    - Was a piece lost?
    - Was it in the right order?
    - Was it error-free?

- Am I being misinterpreted?


Bob
Alice


Bob
???
Alice

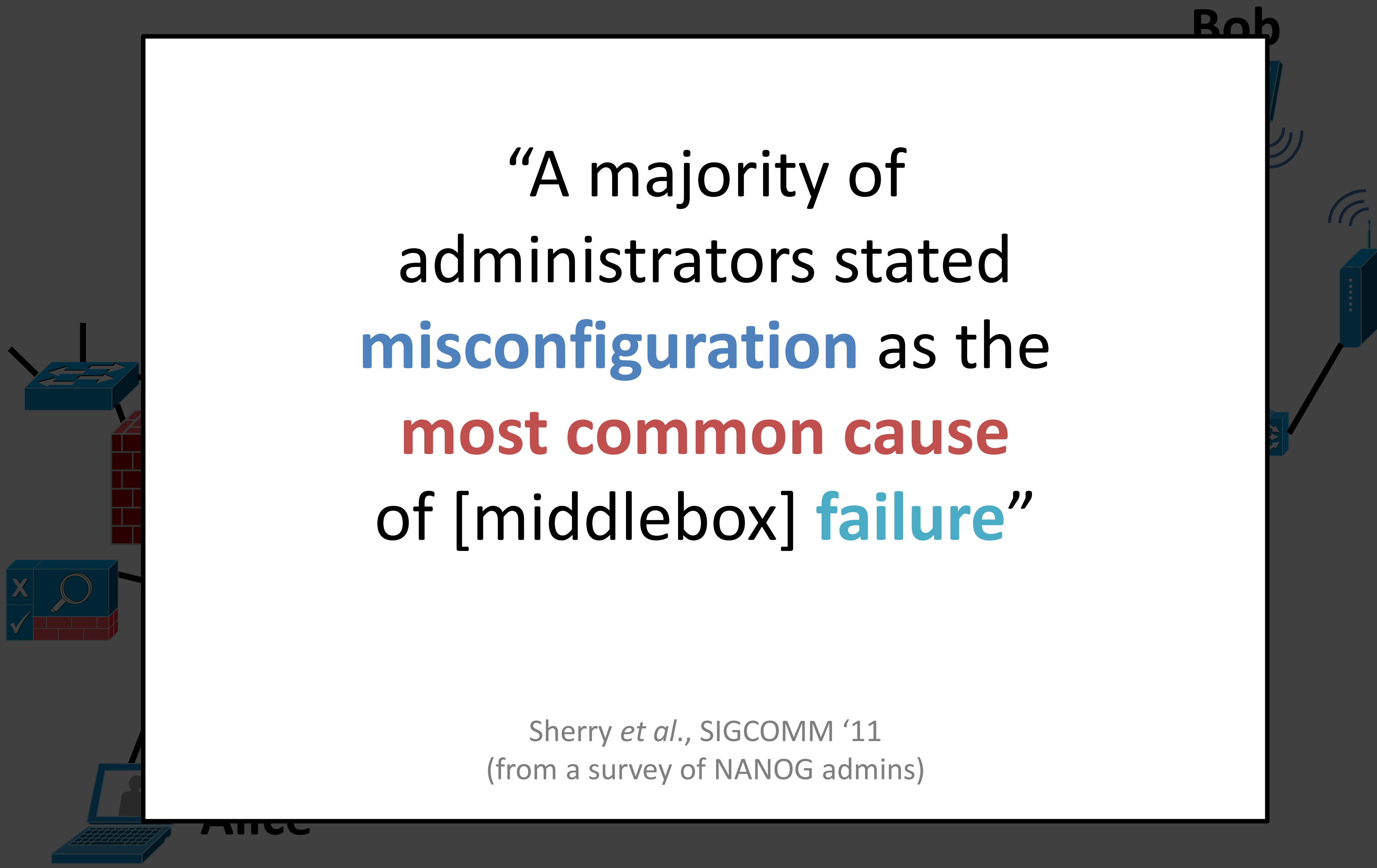


“Across all network sizes, the **number of middleboxes** is **on par** with the **number of routers** in a network”

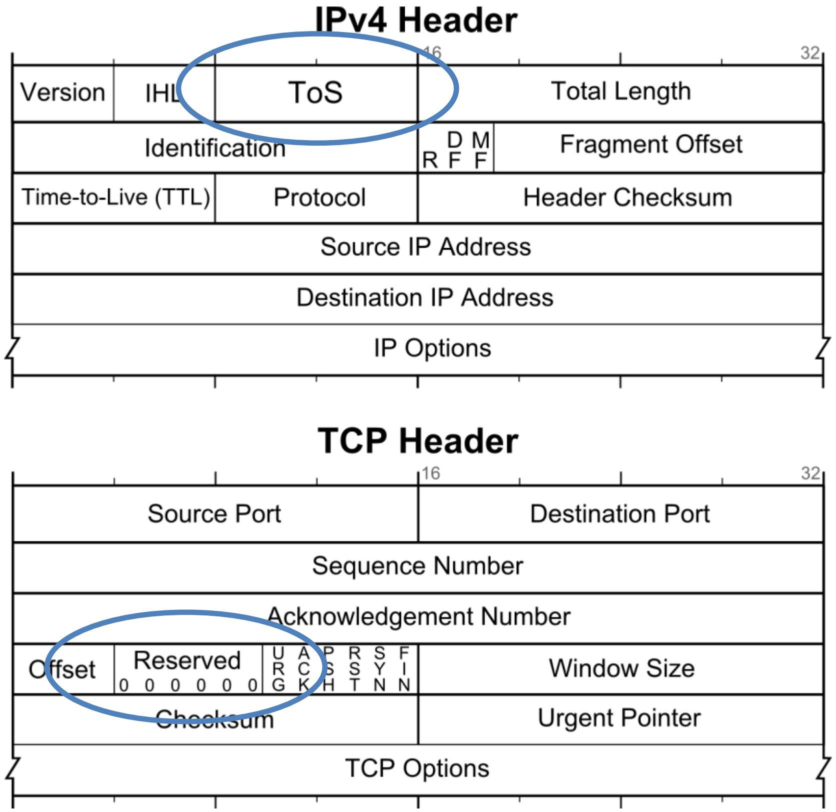
Sherry *et al.*, SIGCOMM ‘11
(from a survey of NANOG admins)

“A majority of administrators stated **misconfiguration** as the **most common cause** of [middlebox] **failure**”

Sherry *et al.*, SIGCOMM ‘11
(from a survey of NANOG admins)



Example: ECN



IPv4 Header

16 32

Version | IHL | ToS | Total Length

Identification | R DF MF | Fragment Offset

Time-to-Live (TTL) | Protocol | Header Checksum

Source IP Address

Destination IP Address

IP Options

TCP Header

16 32

Source Port | Destination Port

Sequence Number

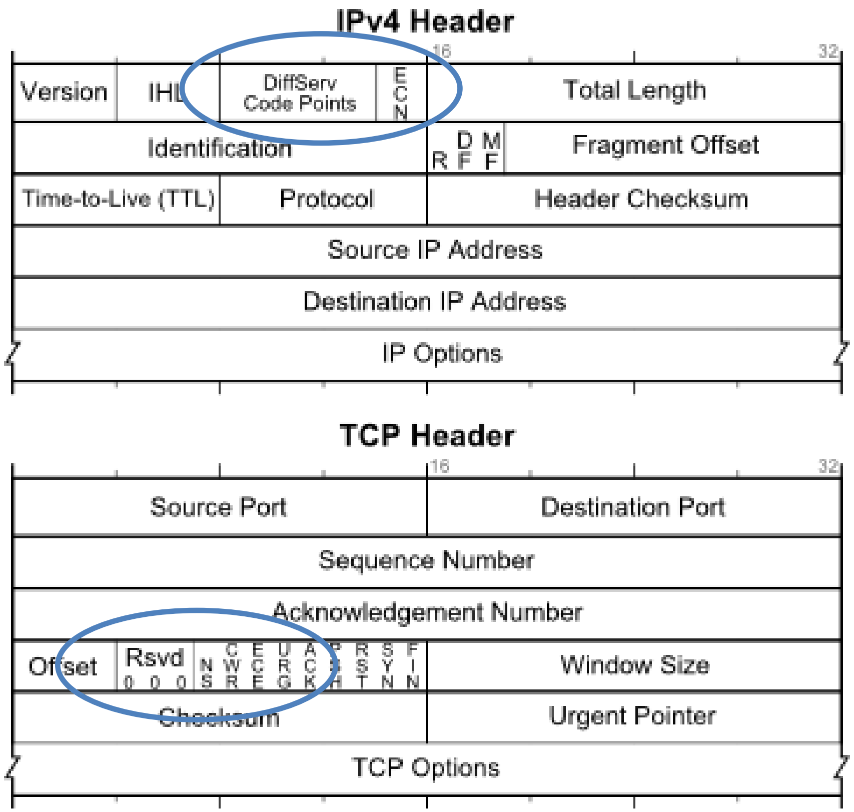
Acknowledgement Number

Offset | Reserved 0 0 0 0 0 0 | URG ACK PSH RST SYN FIN | Window Size

Checksum | Urgent Pointer

TCP Options

1980



IPv4 Header

16 32

Version | IHL | DiffServ Code Points | ECN | Total Length

Identification | R DF MF | Fragment Offset

Time-to-Live (TTL) | Protocol | Header Checksum

Source IP Address

Destination IP Address

IP Options

TCP Header

16 32

Source Port | Destination Port

Sequence Number

Acknowledgement Number

Offset | Rsvd 0 0 0 | NS CWR ECE URG ACK PSH RST SYN FIN | Window Size

Checksum | Urgent Pointer

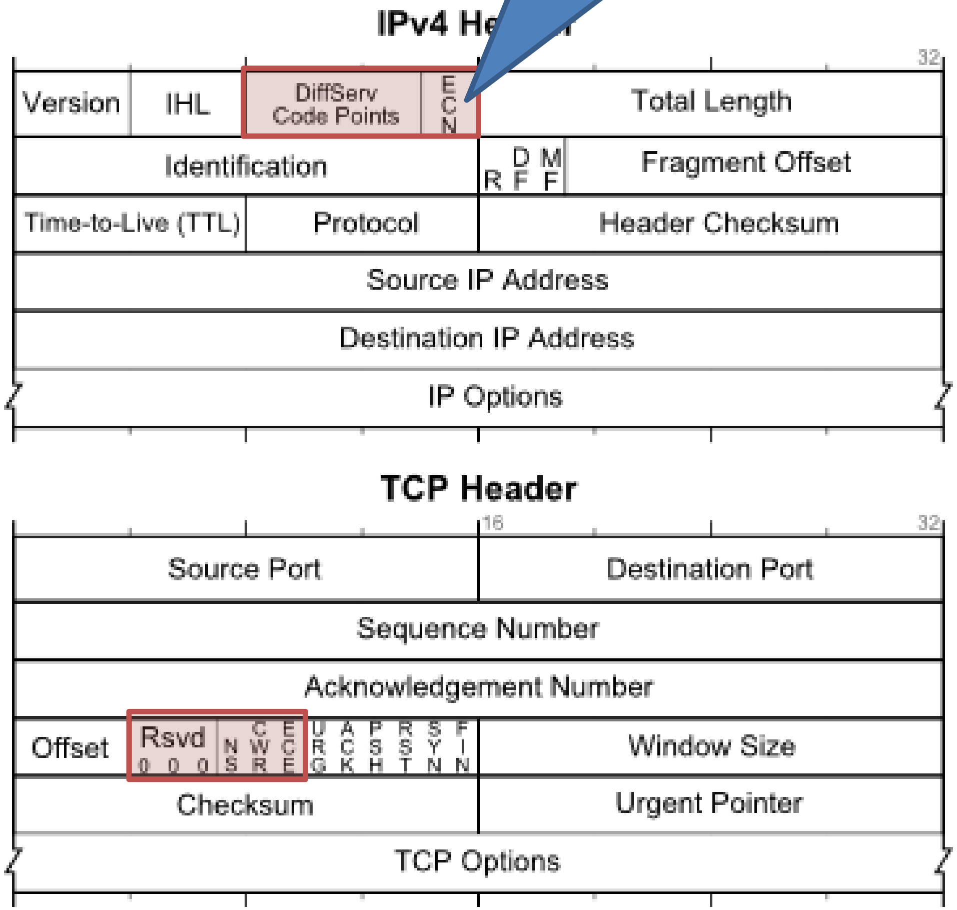
TCP Options

2000

Example: ECN

0b11 == congestion experienced

- Switch was copying a value to the ToS byte[1]



IPv4 Header

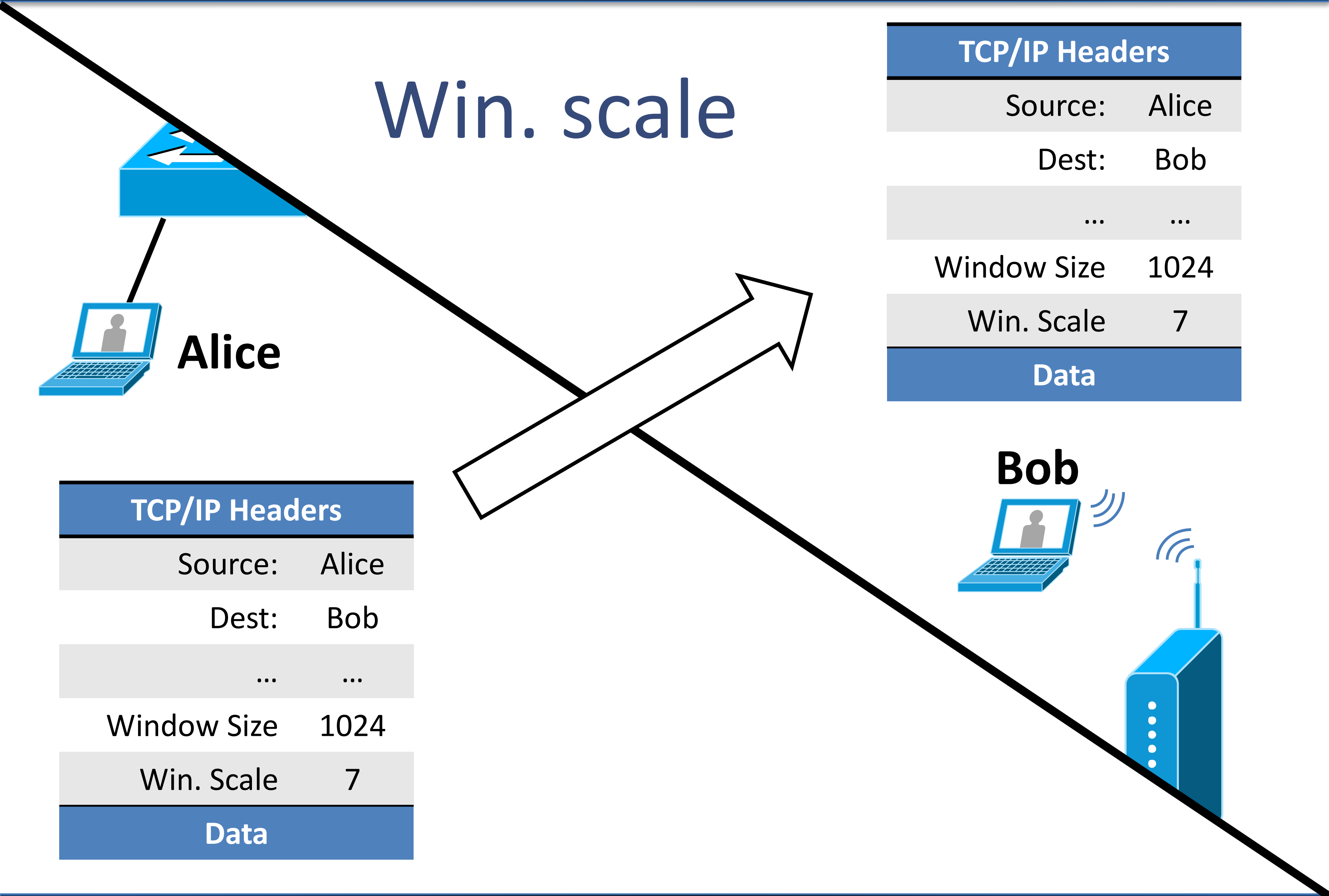
Version	IHL	DiffServ Code Points	ECN	Total Length	
Identification			R DF MF	Fragment Offset	
Time-to-Live (TTL)	Protocol			Header Checksum	
Source IP Address					
Destination IP Address					
IP Options					

TCP Header

Source Port													Destination Port
Sequence Number													
Acknowledgement Number													
Offset	Rsvd 0 0 0	NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN			Window Size
Checksum													Urgent Pointer
TCP Options													

[1] Bauer *et al*. "Measuring the State of ECN Readiness in Servers, Clients, and Routers." In *Proc. of IMC 2011*.

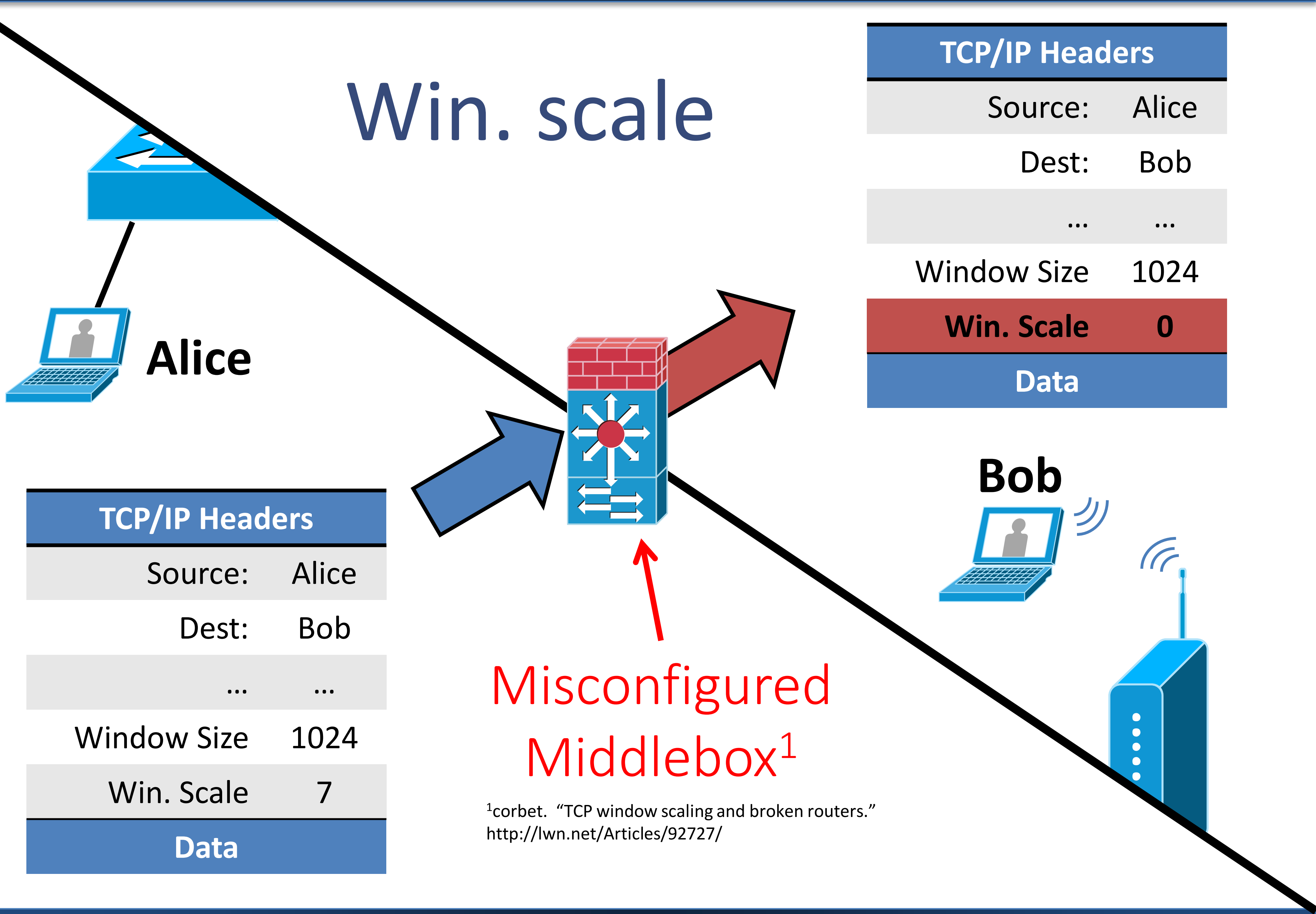
Win. scale



TCP/IP Headers	
Source:	Alice
Dest:	Bob
…	…
Window Size	1024
Win. Scale	7
Data	

TCP/IP Headers	
Source:	Alice
Dest:	Bob
…	…
Window Size	1024
Win. Scale	7
Data	

Win. scale



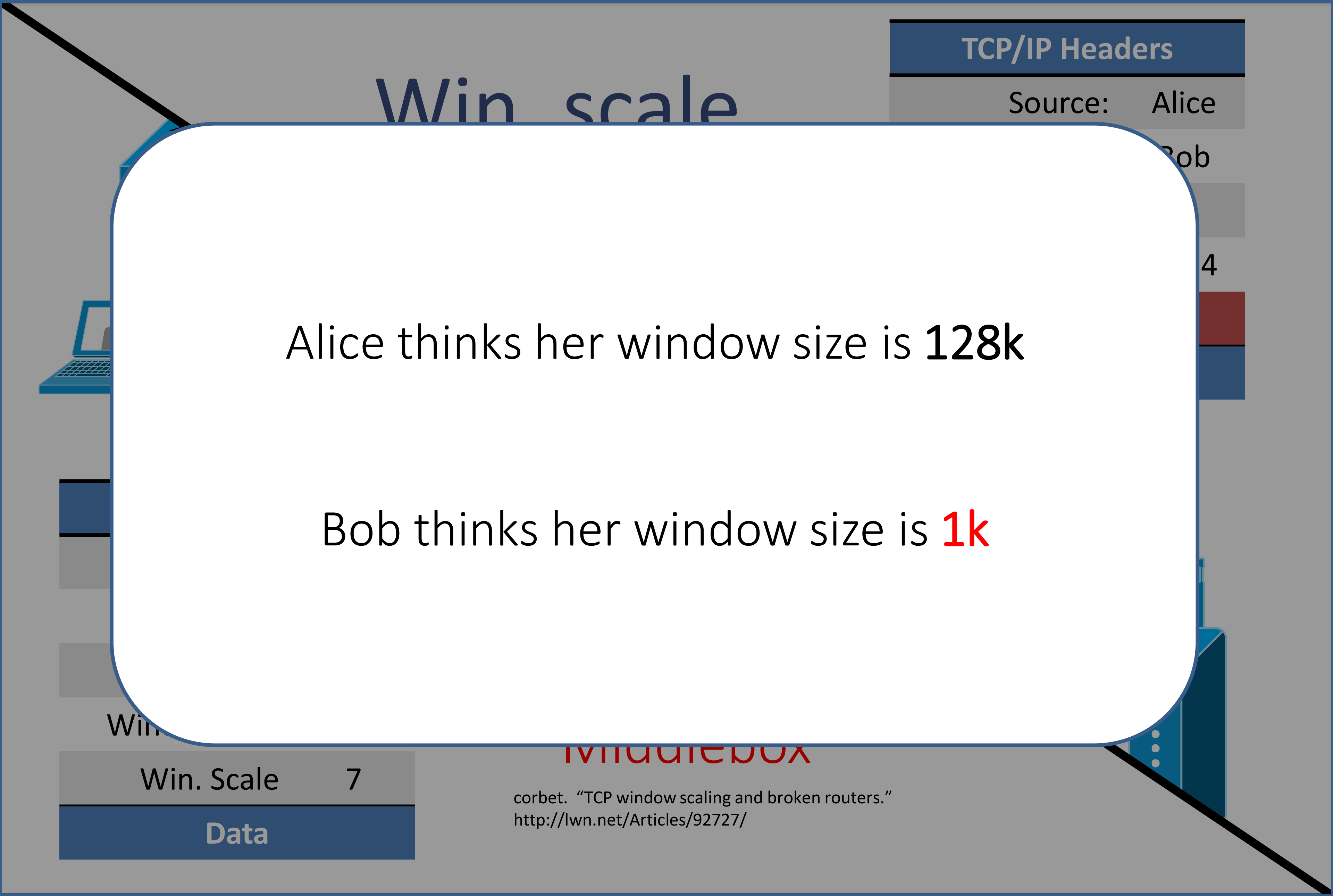
TCP/IP Headers	
Source:	Alice
Dest:	Bob
…	…
Window Size	1024
Win. Scale	7
Data	

TCP/IP Headers	
Source:	Alice
Dest:	Bob
…	…
Window Size	1024
Win. Scale	0
Data	

Misconfigured Middlebox[1]

[1] corbet. "TCP window scaling and broken routers." http://lwn.net/Articles/92727/

Win. scale



TCP/IP Headers	
Source:	Alice
	ob
	4

Alice thinks her window size is **128k**

Bob thinks her window size is 1k

Wi	
Win. Scale	7
Data	

corbet. “TCP window scaling and broken routers.” http://lwn.net/Articles/92727/

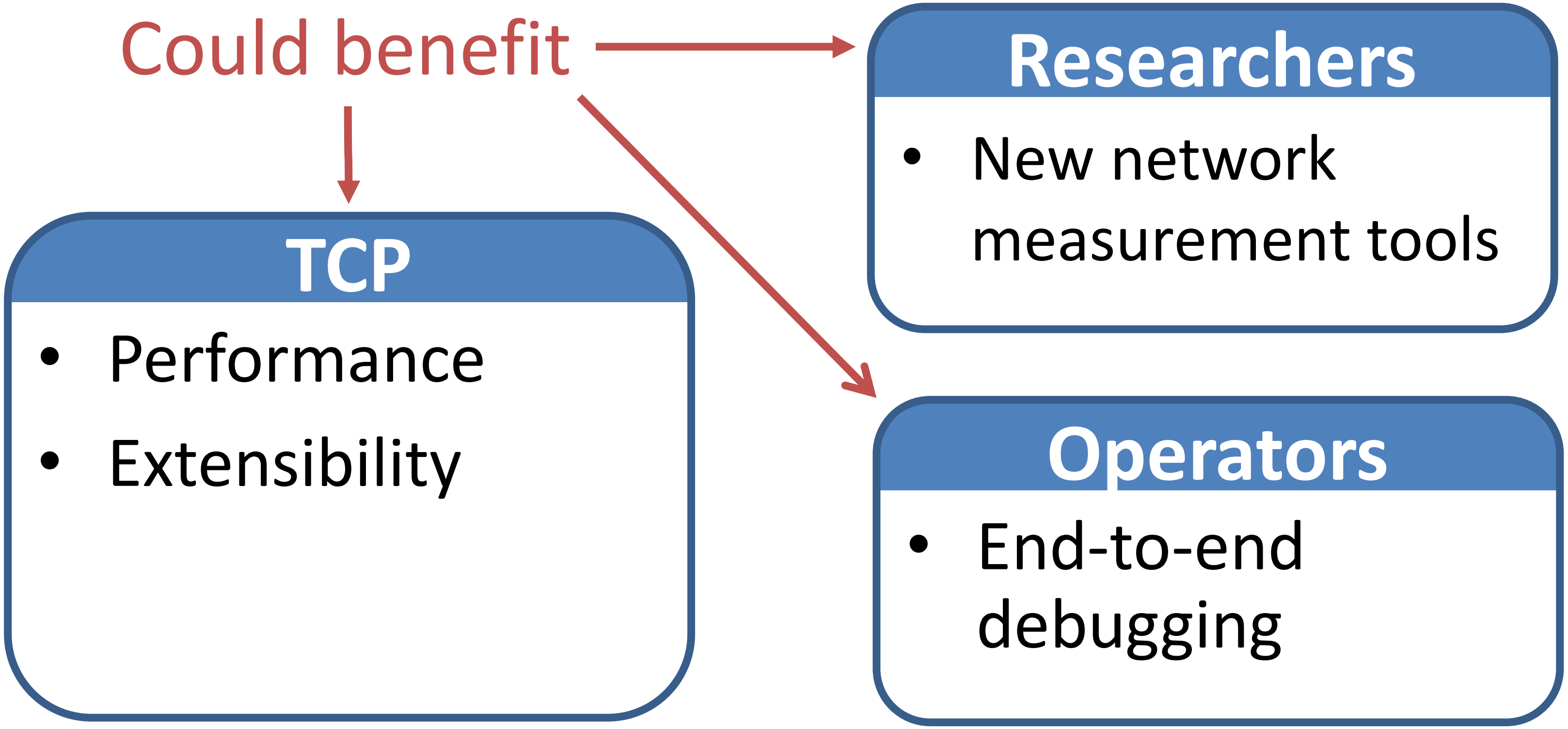
Other Examples

- TCP SACK
- Artificial TCP flow control
- Path MTU discovery
    - ICMP blocking
    - ICMP misquoting
    - TCP MSS alterations
- IP and TCP options stripped
    - Extra problematic:
        - Asymmetric (stripped on SYN-ACK but not SYN)
        - Allowed in handshake, then stripped

Middlebox Misconfiguration

- These are real problems
- Will continue to occur
  - The network is not getting any less intelligent
- Are critical and timely right now
  - Multipath TCP
  - TCP Fast Open
  - Gentle Aggression TCP (proactive/reactive/corrective)
  - tcpcrypt
  - ECN (still)

Wouldn't it be great if we had an easy way to detect these?



Could benefit

TCP

- Performance
- Extensibility

Researchers

- New network measurement tools

Operators

- End-to-end debugging

Challenges

- Available and reliable communications channel
    - Out-of-band ICMP?
    - New IP or TCP option?
    - Redefine a field?
- Capacity
- Incrementally deployable
- Middlebox-cooperative
- Inform both endpoints

HICCUPS

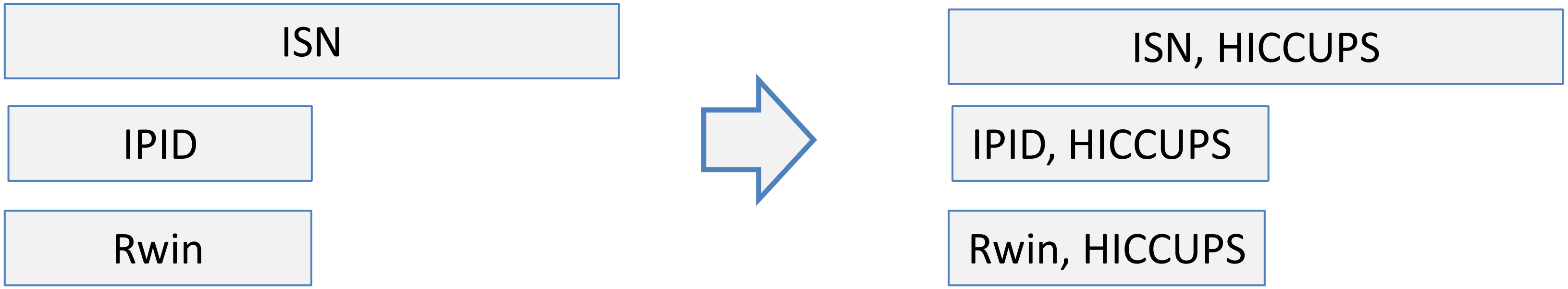
HICCUPS is a lightweight TCP extension that exposes in-flight packet header modification to endpoints

- HICCUPS seeks to automate the question:

"Did my packet arrive at the destination with the same headers as sent?"

HICCUPS Methodology

- Overloads three header fields in TCP 3WHS…



ISN

IPID

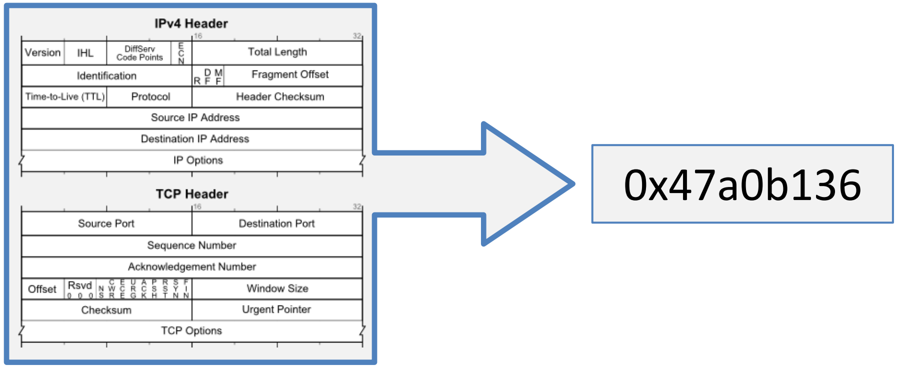
Rwin

ISN, HICCUPS

IPID, HICCUPS

Rwin, HICCUPS

- …with a function of the packet headers



IPv4 Header

Version | IHL | DiffServ Code Points | ECN | Total Length

Identification | R DF MF | Fragment Offset

Time-to-Live (TTL) | Protocol | Header Checksum

Source IP Address

Destination IP Address

IP Options

TCP Header

Source Port | Destination Port

Sequence Number

Acknowledgement Number

Offset | Rsvd 0 0 0 | NS CWR ECE URG ACK PSH RST SYN FIN | Window Size

Checksum | Urgent Pointer

TCP Options

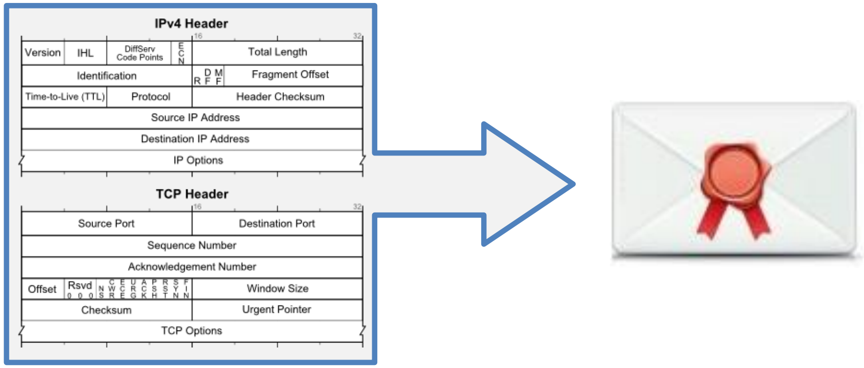
0x47a0b136

HICCUPS Methodology

- Spread over 3 fields in case one is changed
- Lightweight hash function
  - Only have three sets of 12-bits
  - Assume no shared secret available
  - Preimage and hash sent together
  - Primary goal is to reduce collisions
- Add randomness (salt) to ISN

HICCUPS Methodology

- Creates an end-to-end *tamper-evident* seal over the packet headers
- Different than a checksum
    - If mods occur, we still accept the packet



Using HICCUPS

- Once a host's TCP stack is HICCUPS-enabled,

HICCUPS can be used

without endpoint coordination

- Our long-term vision: all TCP stacks include HICCUPS

TCP Congestion Control	**TCP HICCUPS**
Infers e2e congestion state	Infers e2e packet header modification state

Implementation

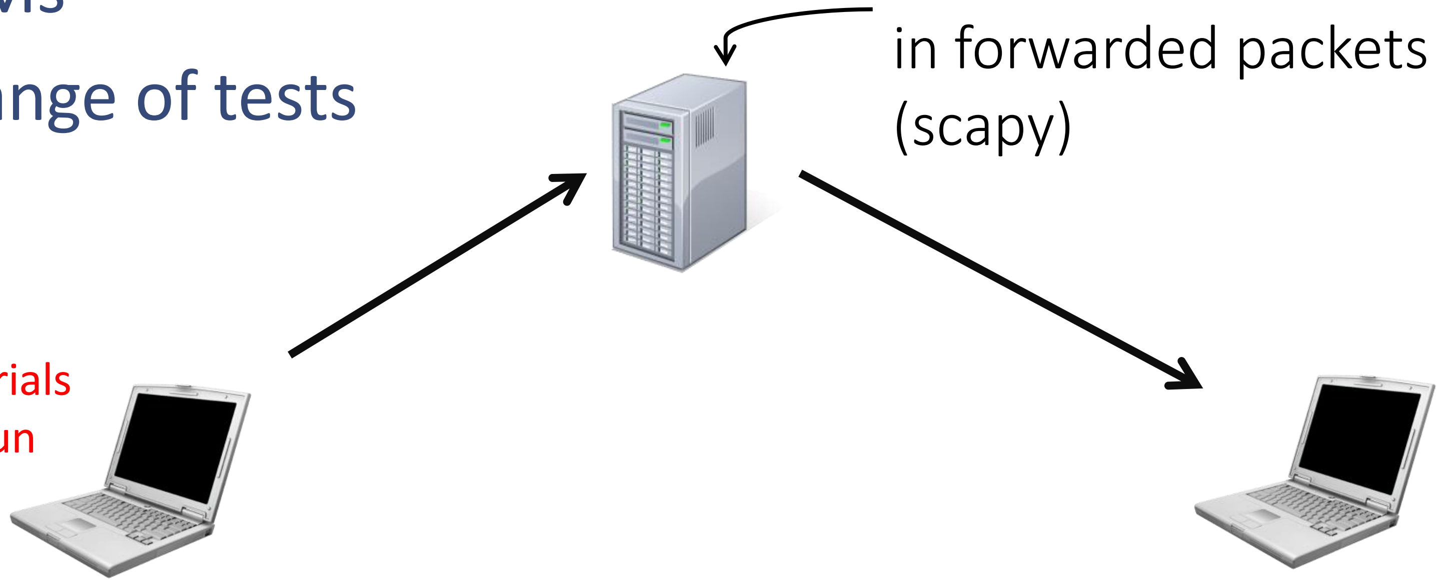
- Patch written for Linux kernel v3.9.4 TCP stack
- Requires no action by applications
- However, we do provide optional features:
  - Get HICCUPS status
  - Manually specify fields to check
  - Engage AppSalt mode (see paper)
- Set of cross-platform userspace tools

Performance

- Analyzed HICCUPS kernel overhead with ftrace
- Increases mean processing time by about 10µs
    - About 8.5% of the total SYN/ACK processing time
- If load gets too high, automatically mitigates with SYN cookies

Validation

- Controlled environment
- VMs
- Range of tests



Simulates a middlebox that overwrites different fields in forwarded packets (scapy)

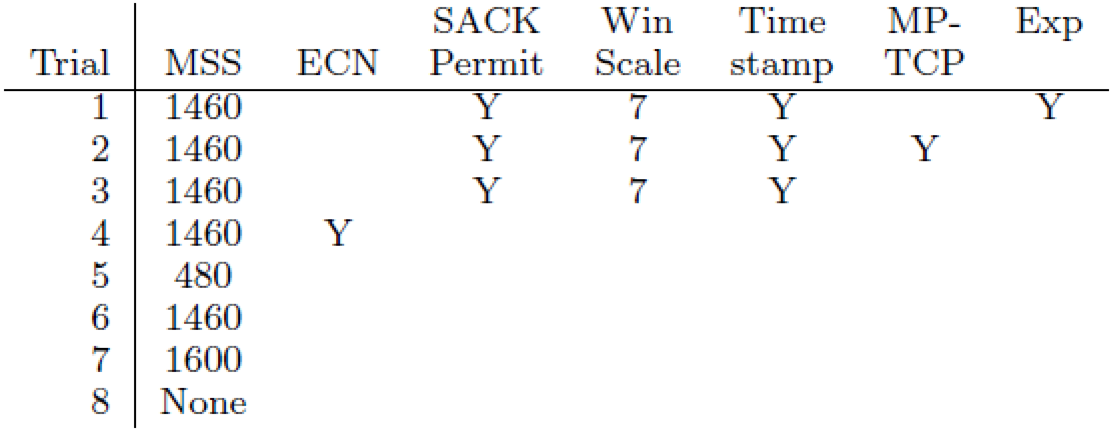
50,000 trials each run

Host A
HICCUPS-enabled

Host B
HICCUPS-enabled

Measurements

- Over 26k directed port/path pairs across 197 ASes and 48 countries
- Different ports: 22, 80, 443, and 34343
- Range of parameters

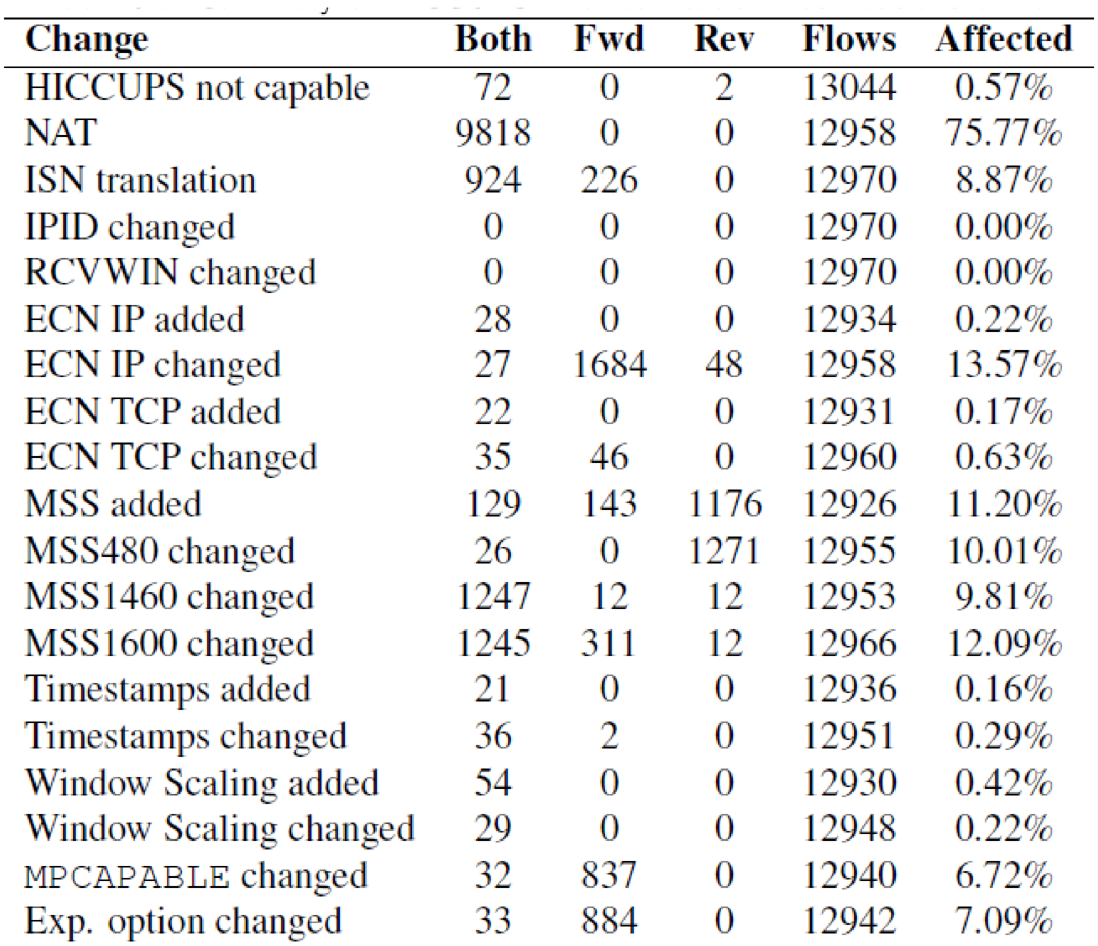


Trial	MSS	ECN	SACK Permit	Win Scale	Time stamp	MP-TCP	Exp
1	1460		Y	7	Y		Y
2	1460		Y	7	Y	Y	
3	1460		Y	7	Y		
4	1460	Y					
5	480						
6	1460						
7	1600						
8	None						

Meas. Summary

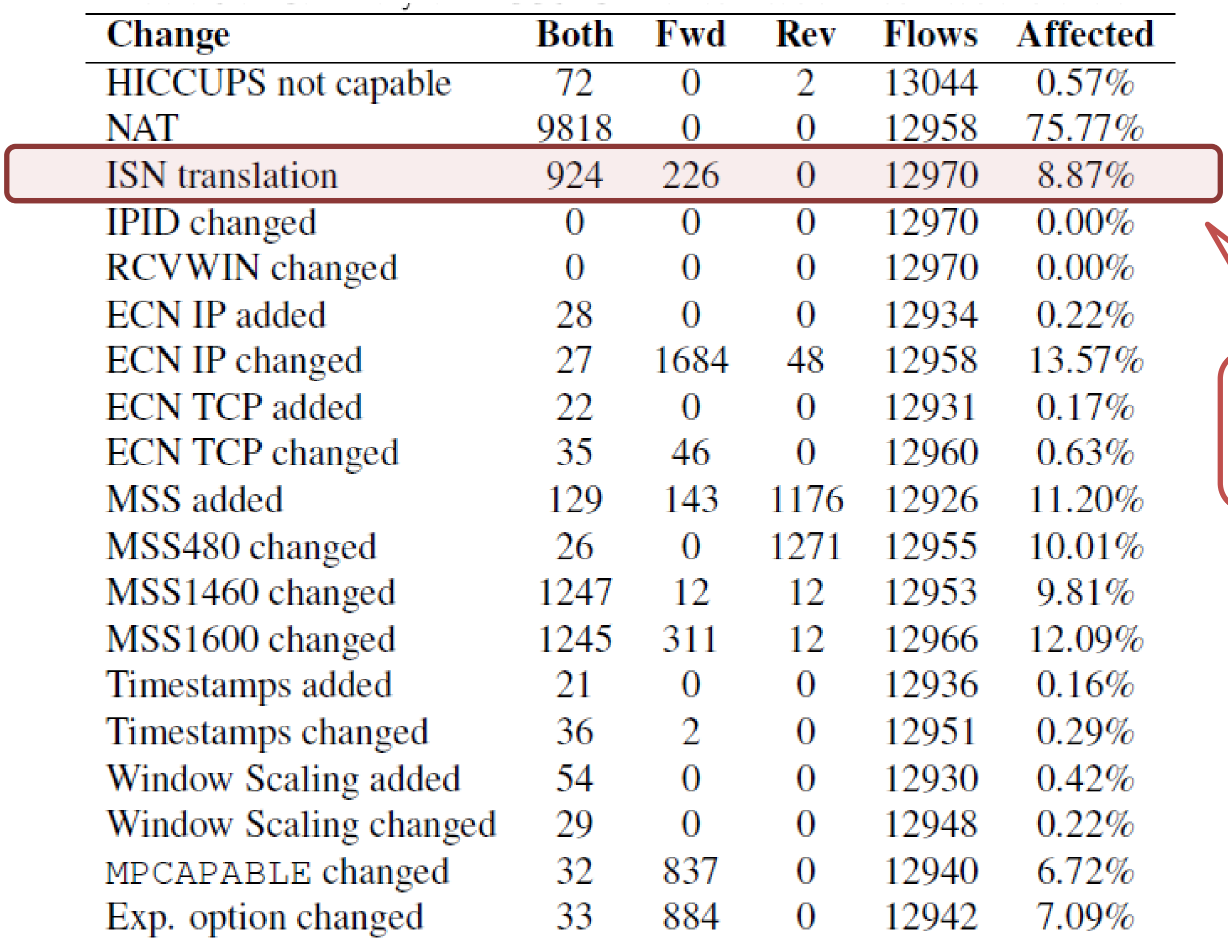
- Almost half of the nodes saw at least one in-path header modification
- More than we expected to find
- Saw asymmetric cases

Mods Detected



Change	Both	Fwd	Rev	Flows	Affected
HICCUPS not capable	72	0	2	13044	0.57%
NAT	9818	0	0	12958	75.77%
ISN translation	924	226	0	12970	8.87%
IPID changed	0	0	0	12970	0.00%
RCVWIN changed	0	0	0	12970	0.00%
ECN IP added	28	0	0	12934	0.22%
ECN IP changed	27	1684	48	12958	13.57%
ECN TCP added	22	0	0	12931	0.17%
ECN TCP changed	35	46	0	12960	0.63%
MSS added	129	143	1176	12926	11.20%
MSS480 changed	26	0	1271	12955	10.01%
MSS1460 changed	1247	12	12	12953	9.81%
MSS1600 changed	1245	311	12	12966	12.09%
Timestamps added	21	0	0	12936	0.16%
Timestamps changed	36	2	0	12951	0.29%
Window Scaling added	54	0	0	12930	0.42%
Window Scaling changed	29	0	0	12948	0.22%
MPCAPABLE changed	32	837	0	12940	6.72%
Exp. option changed	33	884	0	12942	7.09%

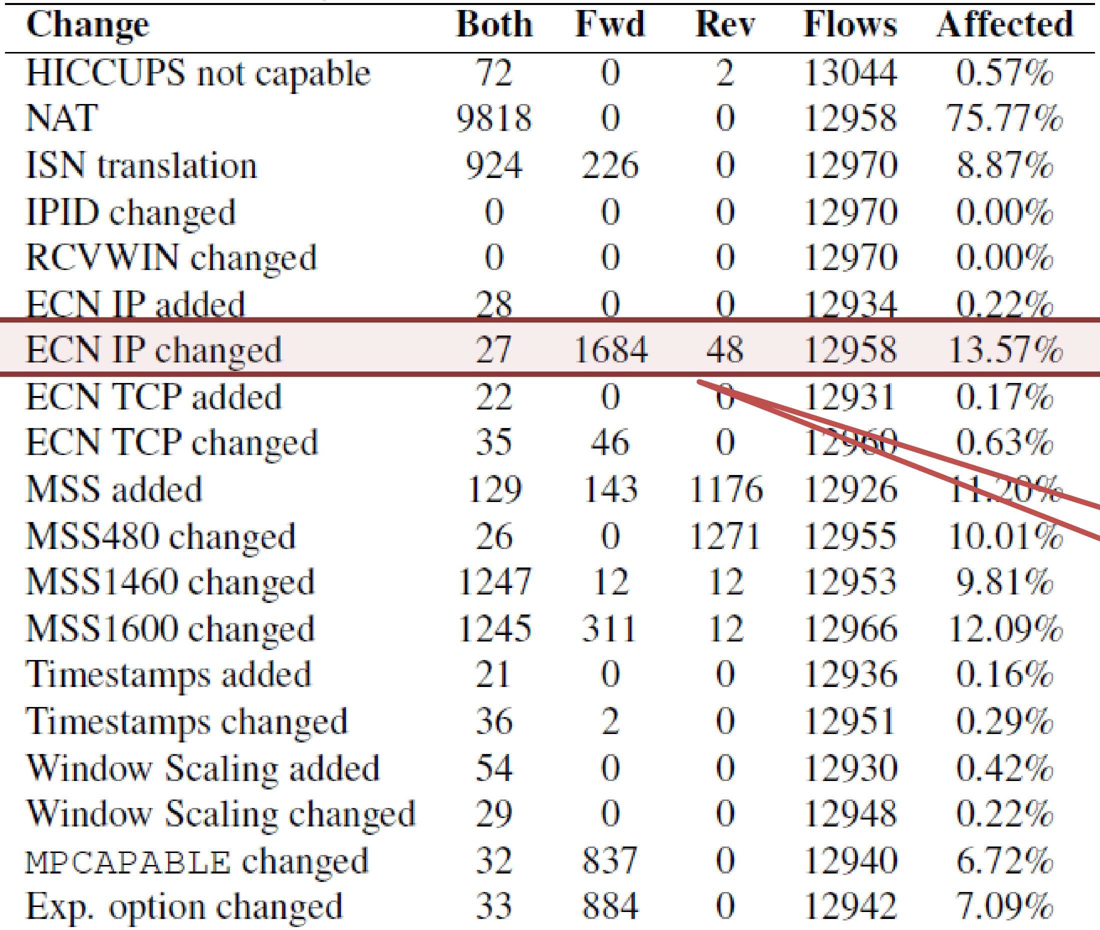
What can go wrong?



Change	Both	Fwd	Rev	Flows	Affected
HICCUPS not capable	72	0	2	13044	0.57%
NAT	9818	0	0	12958	75.77%
ISN translation	924	226	0	12970	8.87%
IPID changed	0	0	0	12970	0.00%
RCVWIN changed	0	0	0	12970	0.00%
ECN IP added	28	0	0	12934	0.22%
ECN IP changed	27	1684	48	12958	13.57%
ECN TCP added	22	0	0	12931	0.17%
ECN TCP changed	35	46	0	12960	0.63%
MSS added	129	143	1176	12926	11.20%
MSS480 changed	26	0	1271	12955	10.01%
MSS1460 changed	1247	12	12	12953	9.81%
MSS1600 changed	1245	311	12	12966	12.09%
Timestamps added	21	0	0	12936	0.16%
Timestamps changed	36	2	0	12951	0.29%
Window Scaling added	54	0	0	12930	0.42%
Window Scaling changed	29	0	0	12948	0.22%
MPCAPABLE changed	32	837	0	12940	6.72%
Exp. option changed	33	884	0	12942	7.09%

Potential SACK disruption

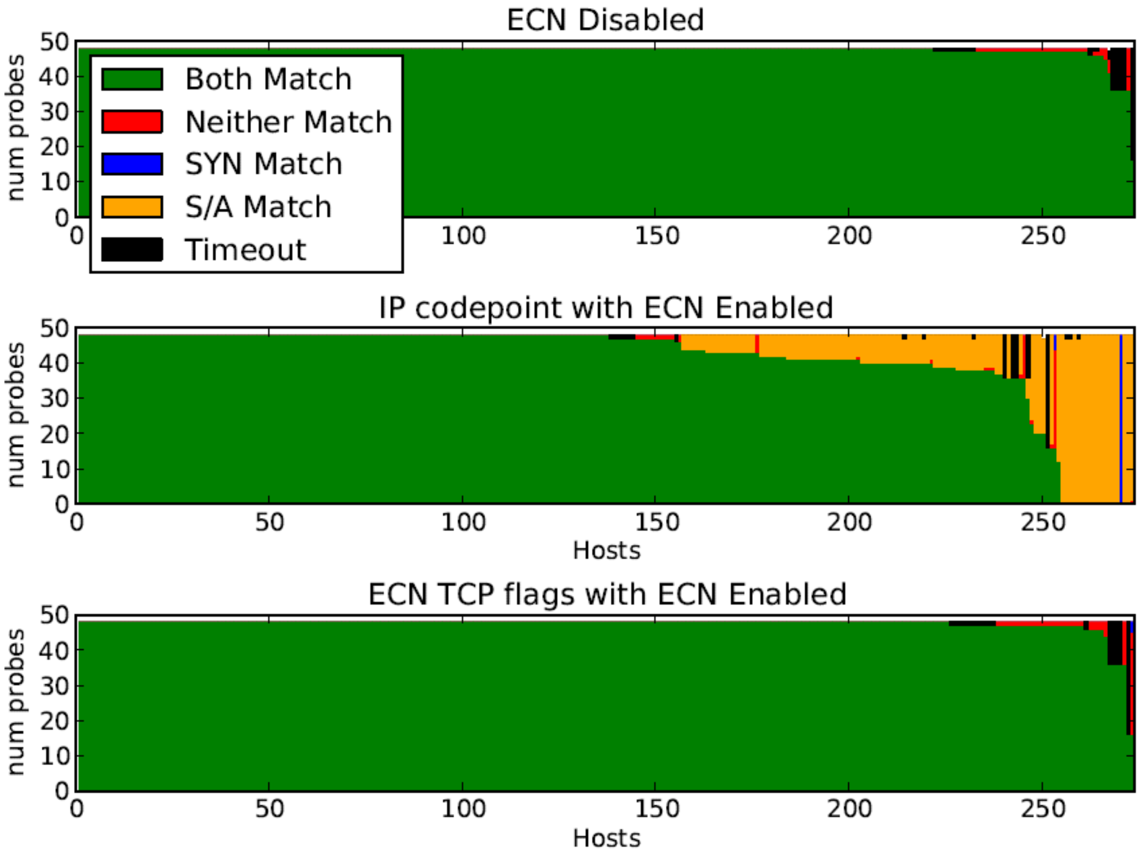
What can go wrong?



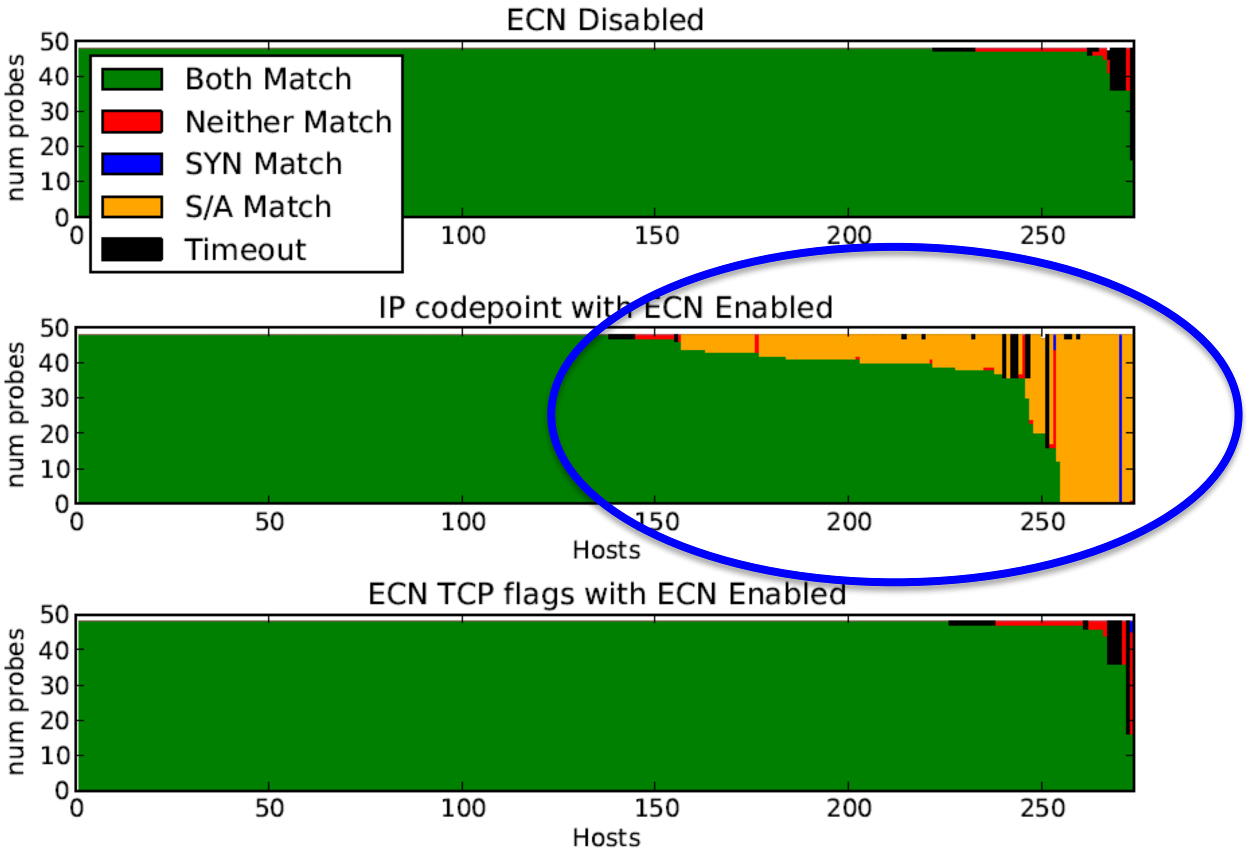
Change	Both	Fwd	Rev	Flows	Affected
HICCUPS not capable	72	0	2	13044	0.57%
NAT	9818	0	0	12958	75.77%
ISN translation	924	226	0	12970	8.87%
IPID changed	0	0	0	12970	0.00%
RCVWIN changed	0	0	0	12970	0.00%
ECN IP added	28	0	0	12934	0.22%
ECN IP changed	27	1684	48	12958	13.57%
ECN TCP added	22	0	0	12931	0.17%
ECN TCP changed	35	46	0	12960	0.63%
MSS added	129	143	1176	12926	11.20%
MSS480 changed	26	0	1271	12955	10.01%
MSS1460 changed	1247	12	12	12953	9.81%
MSS1600 changed	1245	311	12	12966	12.09%
Timestamps added	21	0	0	12936	0.16%
Timestamps changed	36	2	0	12951	0.29%
Window Scaling added	54	0	0	12930	0.42%
Window Scaling changed	29	0	0	12948	0.22%
MPCAPABLE changed	32	837	0	12940	6.72%
Exp. option changed	33	884	0	12942	7.09%

Potential ToS byte semantics

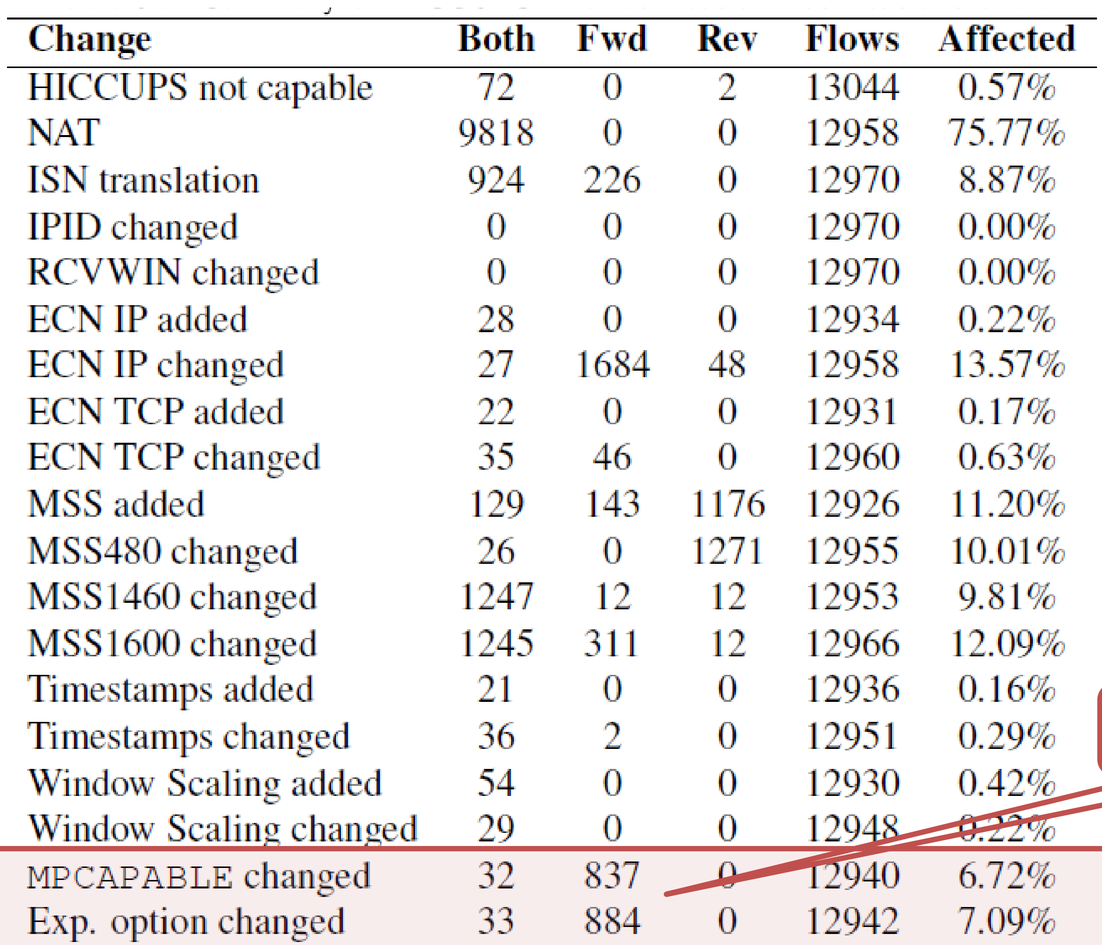
ECN IP bits


ECN Disabled
num probes
0
10
20
30
40
50
Both Match
Neither Match
SYN Match
S/A Match
Timeout
0
100
150
200
250
IP codepoint with ECN Enabled
num probes
0
10
20
30
40
50
0
50
100
150
200
250
Hosts
ECN TCP flags with ECN Enabled
num probes
0
10
20
30
40
50
0
50
100
150
200
250
Hosts

ECN IP bits


ECN Disabled
num probes
50
40
30
20
10
0
0
100
150
200
250
Both Match
Neither Match
SYN Match
S/A Match
Timeout
IP codepoint with ECN Enabled
num probes
50
40
30
20
10
0
0
50
100
150
200
250
Hosts
ECN TCP flags with ECN Enabled
num probes
50
40
30
20
10
0
0
50
100
150
200
250
Hosts

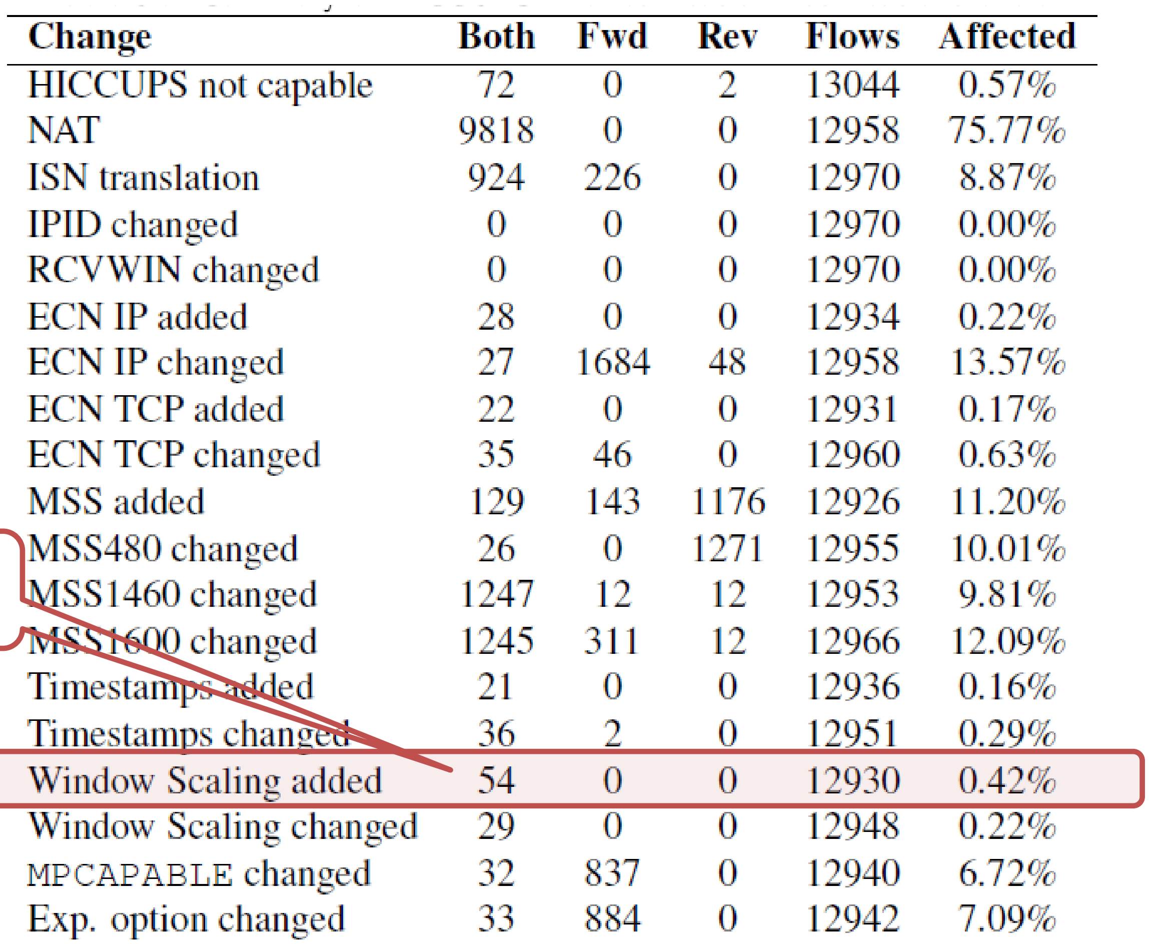
What can go wrong?



Change	Both	Fwd	Rev	Flows	Affected
HICCUPS not capable	72	0	2	13044	0.57%
NAT	9818	0	0	12958	75.77%
ISN translation	924	226	0	12970	8.87%
IPID changed	0	0	0	12970	0.00%
RCVWIN changed	0	0	0	12970	0.00%
ECN IP added	28	0	0	12934	0.22%
ECN IP changed	27	1684	48	12958	13.57%
ECN TCP added	22	0	0	12931	0.17%
ECN TCP changed	35	46	0	12960	0.63%
MSS added	129	143	1176	12926	11.20%
MSS480 changed	26	0	1271	12955	10.01%
MSS1460 changed	1247	12	12	12953	9.81%
MSS1600 changed	1245	311	12	12966	12.09%
Timestamps added	21	0	0	12936	0.16%
Timestamps changed	36	2	0	12951	0.29%
Window Scaling added	54	0	0	12930	0.42%
Window Scaling changed	29	0	0	12948	0.22%
MPCAPABLE changed	32	837	0	12940	6.72%
Exp. option changed	33	884	0	12942	7.09%

Options stripped

What can go wrong?

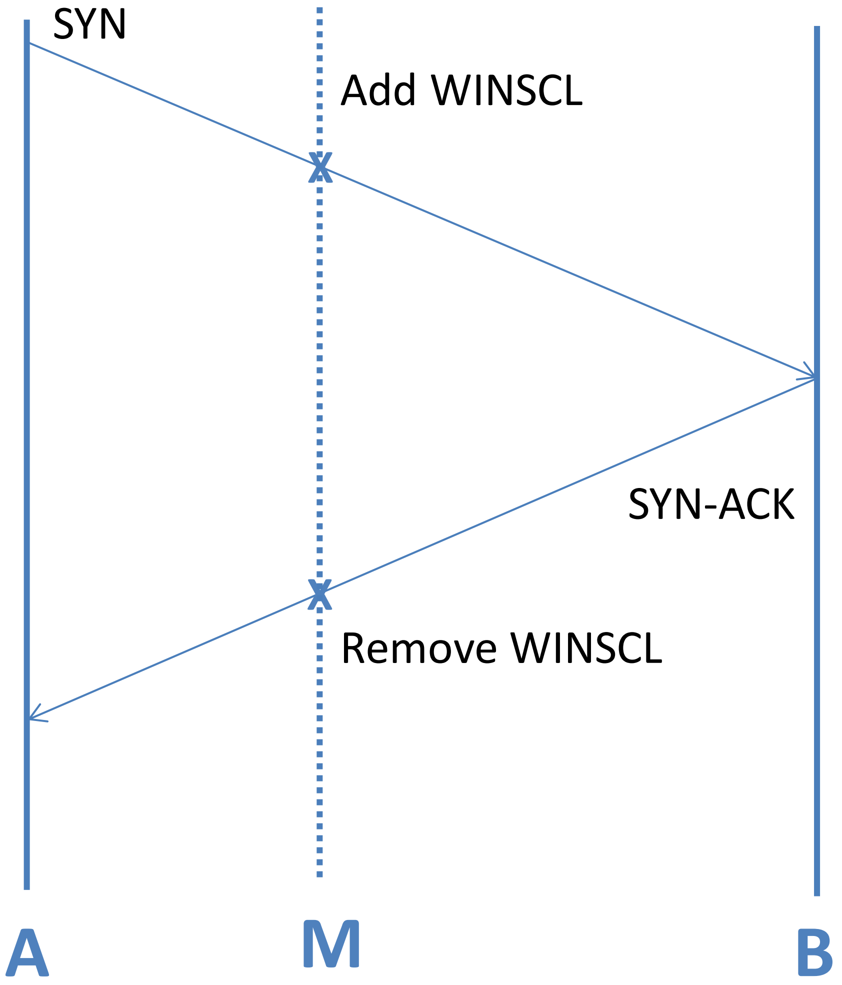


Change	Both	Fwd	Rev	Flows	Affected
HICCUPS not capable	72	0	2	13044	0.57%
NAT	9818	0	0	12958	75.77%
ISN translation	924	226	0	12970	8.87%
IPID changed	0	0	0	12970	0.00%
RCVWIN changed	0	0	0	12970	0.00%
ECN IP added	28	0	0	12934	0.22%
ECN IP changed	27	1684	48	12958	13.57%
ECN TCP added	22	0	0	12931	0.17%
ECN TCP changed	35	46	0	12960	0.63%
MSS added	129	143	1176	12926	11.20%
MSS480 changed	26	0	1271	12955	10.01%
MSS1460 changed	1247	12	12	12953	9.81%
MSS1600 changed	1245	311	12	12966	12.09%
Timestamps added	21	0	0	12936	0.16%
Timestamps changed	36	2	0	12951	0.29%
Window Scaling added	54	0	0	12930	0.42%
Window Scaling changed	29	0	0	12948	0.22%
MPCAPABLE changed	32	837	0	12940	6.72%
Exp. option changed	33	884	0	12942	7.09%

New behavior

Window Scaling

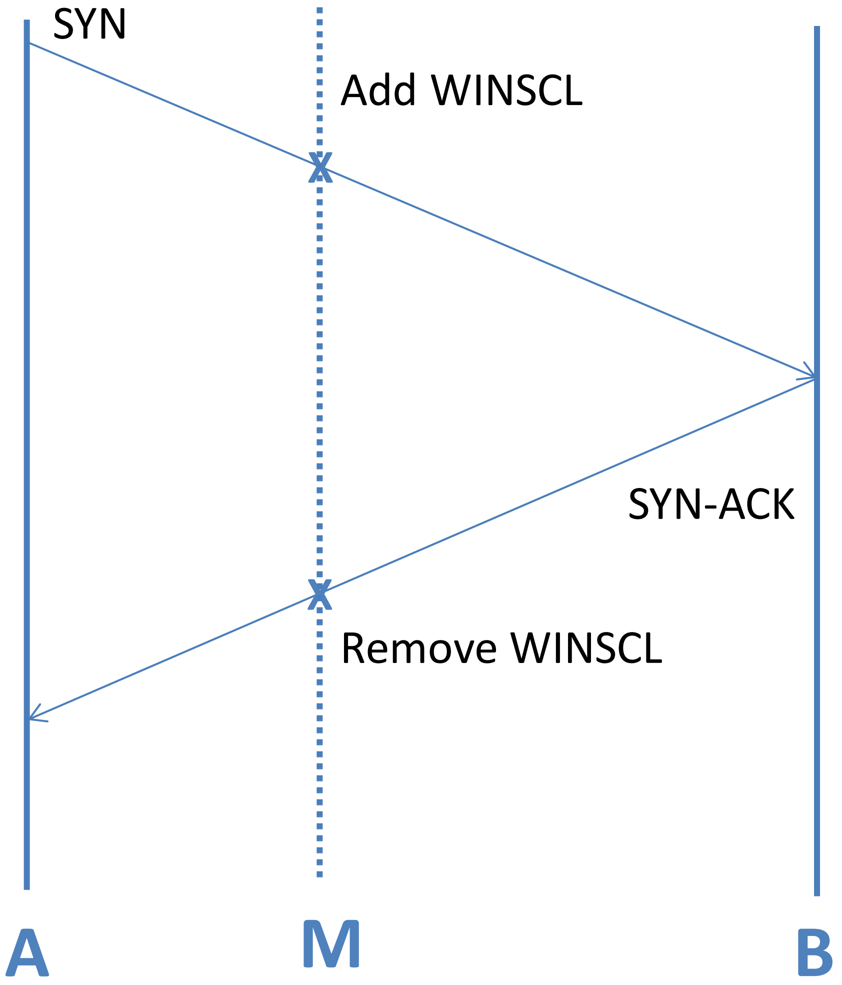
- Israeli PlanetLab node planetlab2.mta.ac.il
- Window scaling option added
- Only when going to ports 80 or 443



Window Scaling

- Israeli PlanetLab node planetlab2.mta.ac.il
- Window scaling option added
- Only when going to ports 80 or 443

Result: bulk transfer is flow-controlled, doubles when WINSCL ignored



Conclusions

- HICCUPS can help TCP infer whether it is being misinterpreted
  - Integrates nicely with TCP, incrementally deployable
  - End-to-end
  - Middlebox-cooperative
- Demonstrated ease of deployment through mass Internet measurements

http://tcphiccups.org