

REVEALING MIDDLEBOXES INTERFERENCE WITH TRACEBOX

Gregory Detal^{*}, Benjamin Hesmans^{*}, Olivier Bonaventure^{*}, Yves Vanaubel[°] and Benoit Donnet[°].

^{*}Université catholique de Louvain

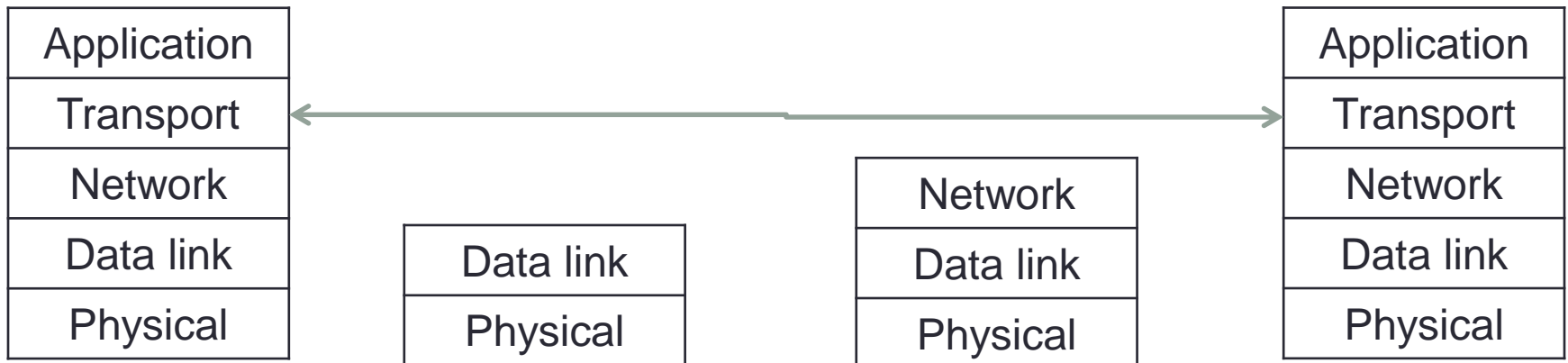
[°]Université de Liège

<http://www.tracebox.org>

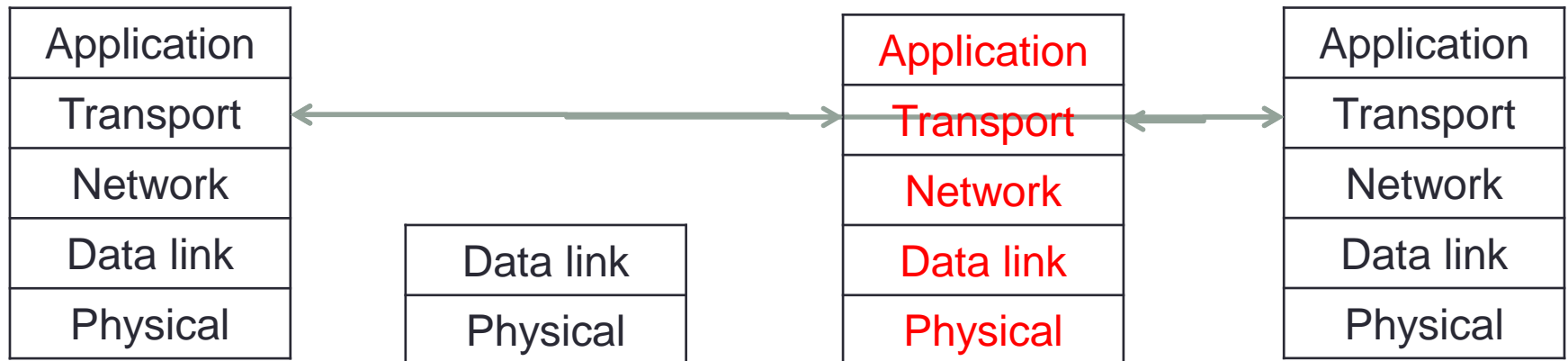
Outline

- **Middleboxes interference**
- Detect packet modification with ICMP
- Tracebox
- Measurements results

The end-to-end principle ...



... does not hold ☹️



In reality

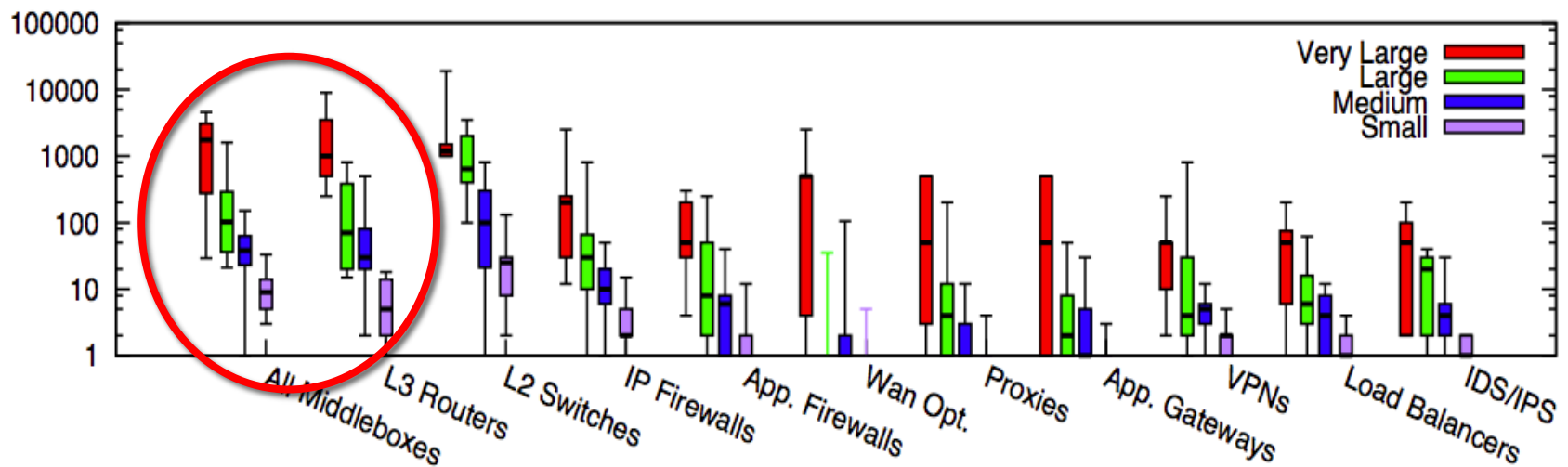
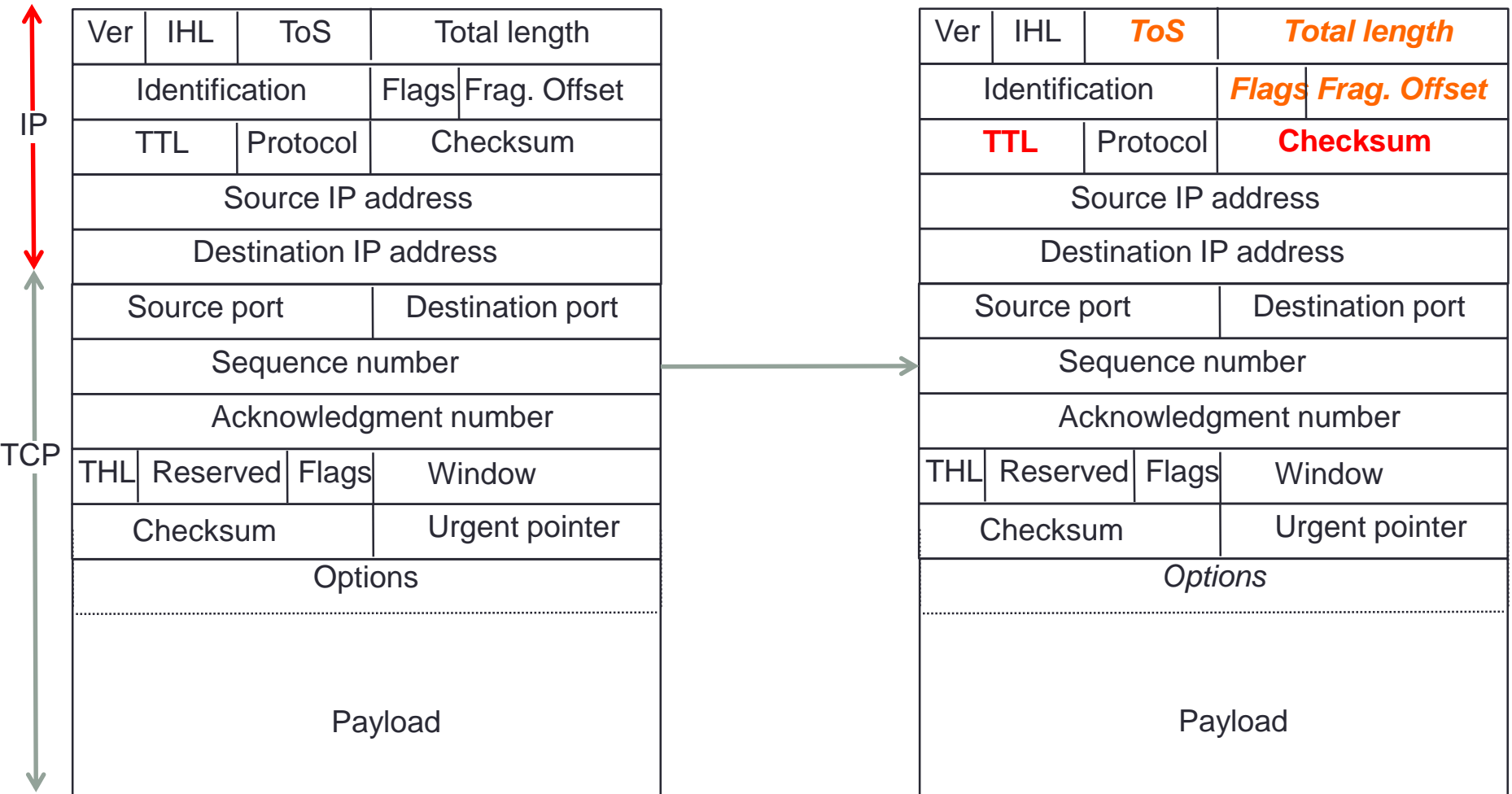


Figure 1: Box plot of middlebox deployments for small (fewer than 1k hosts), medium (1k-10k hosts), large (10k-100k hosts), and very large (more than 100k hosts) enterprise networks. Y-axis is in log scale.

TCP Segment processed by a router



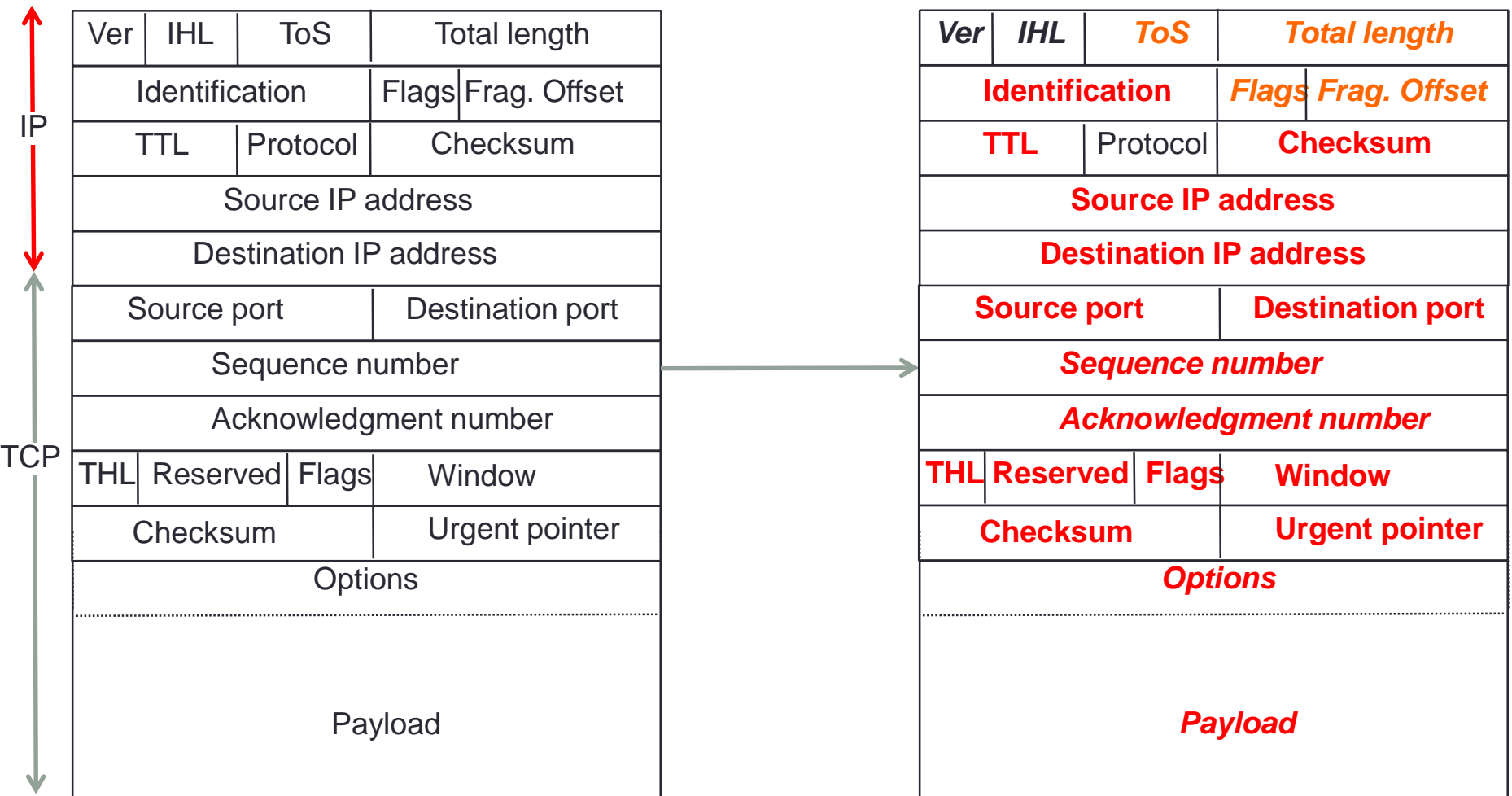
How transparent is the Internet ?

- 25th September 2010 to 30th April 2011
- 142 access networks
- 24 countries
- Craft TCP segments using custom scripts
- Sent specific TCP segments from client to a server in Japan

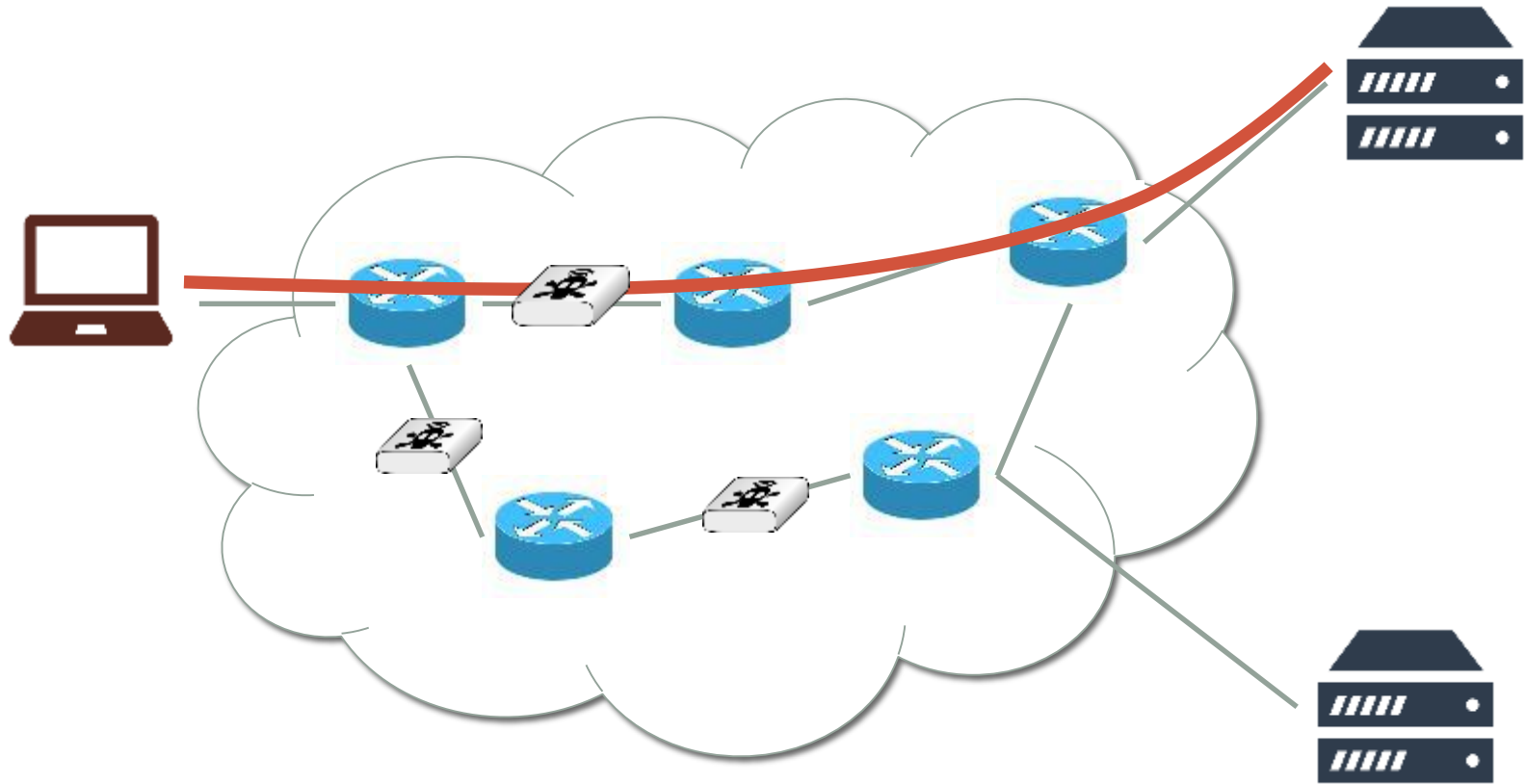
Table 2: Experiment Venues

Country	Home	Hotspot	Cellular	Univ	Ent	Hosting	Total
Australia	0	2	0	0	0	1	3
Austria	0	0	0	0	1	0	1
Belgium	4	0	0	1	0	0	5
Canada	1	0	1	0	1	0	3
Chile	0	0	0	0	1	0	1
China	0	7	0	0	0	0	7
Czech	0	2	0	0	0	0	2
Denmark	0	2	0	0	0	0	2
Finland	1	0	0	3	2	0	6
Germany	3	1	3	4	1	0	12
Greece	2	0	1	0	0	0	3
Indonesia	0	0	0	3	0	0	3
Ireland	0	0	0	0	0	1	1
Italy	1	0	0	0	1	0	2
Japan	19	10	7	3	2	0	41
Romania	1	0	0	0	0	0	1
Russia	0	1	0	0	0	0	1
Spain	0	1	0	1	0	0	2
Sweden	1	0	0	0	0	0	1
Switzerland	2	0	0	0	0	0	2
Thailand	0	0	0	0	2	0	2
U.K.	10	4	4	2	1	1	22
U.S.	3	4	4	0	4	2	17
Vietnam	1	0	0	0	1	0	2
Total	49	34	20	17	17	5	142

TCP Segments on the today's Internet



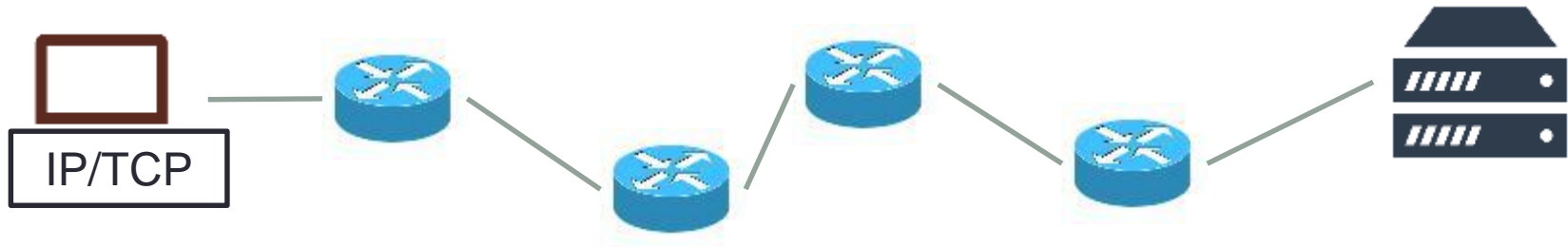
Potentially miss a lot of middleboxes



Outline

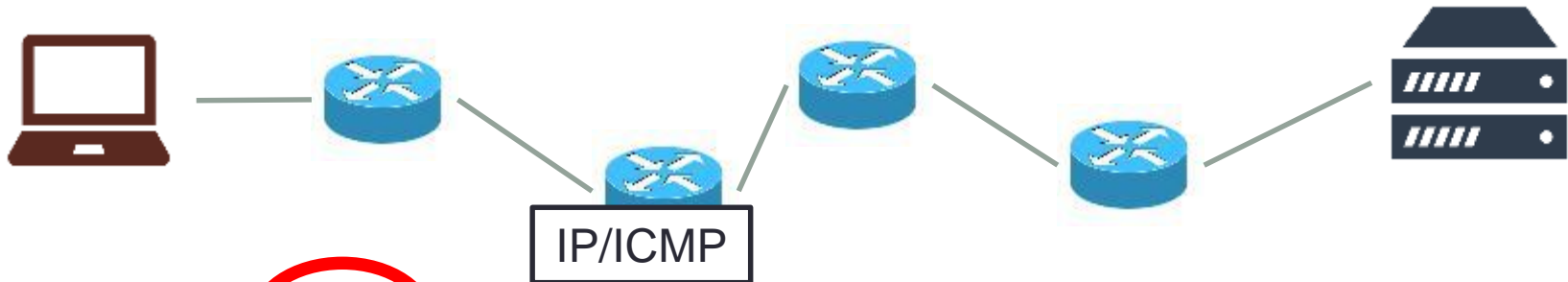
- Middleboxes interference
- **Detect packet modification with ICMP**
- Tracebox
- Measurements results

Traceroute with ICMP in a nutshell



Ver	IHL	ToS	Total length	
Identification		Flags	Frag. Offset	
TTL=2		Protocol	Checksum	
Source IP address				
Destination IP address				
Source port		Destination port		
Sequence number				
Acknowledgment number				
THL	Reserved	Flags	Window	
Checksum		Urgent pointer		

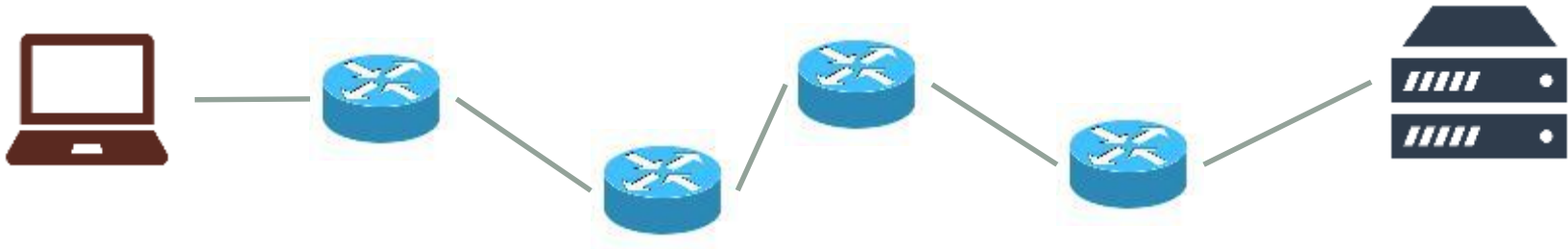
Traceroute with ICMP in a nutshell



IP			
type = 11	code = 0	checksum	
0 (unused)			
Ver	IHL	ToS	Total length
Identification		Flags	Frag. Offset
1	Protocol	Checksum	
Source IP address			
Destination IP address			
Source port		Destination port	
Sequence number			

Use the IP source to identify routers

Middlebox detection with ICMP



Ver	IHL	ToS	Total length	
Identification		Flags	Frag. Offset	
1	Protocol	Checksum		
Source IP address				
Destination IP address				
Source port		Destination port		
Sequence number				

Compare

Ver	IHL	ToS	Total length	
Identification		Flags	Frag. Offset	
2	Protocol	Checksum		
Source IP address				
Destination IP address				
Source port		Destination port		
Sequence number				
Acknowledgment number				
THL	Reserved	Flags	Window	
Checksum			Urgent pointer	

ICMP-based modification detection

- RFC792 requires ICMP to include only the first 8 bytes of the transport header.

- In 1995 RFC1812 requires that routers should q

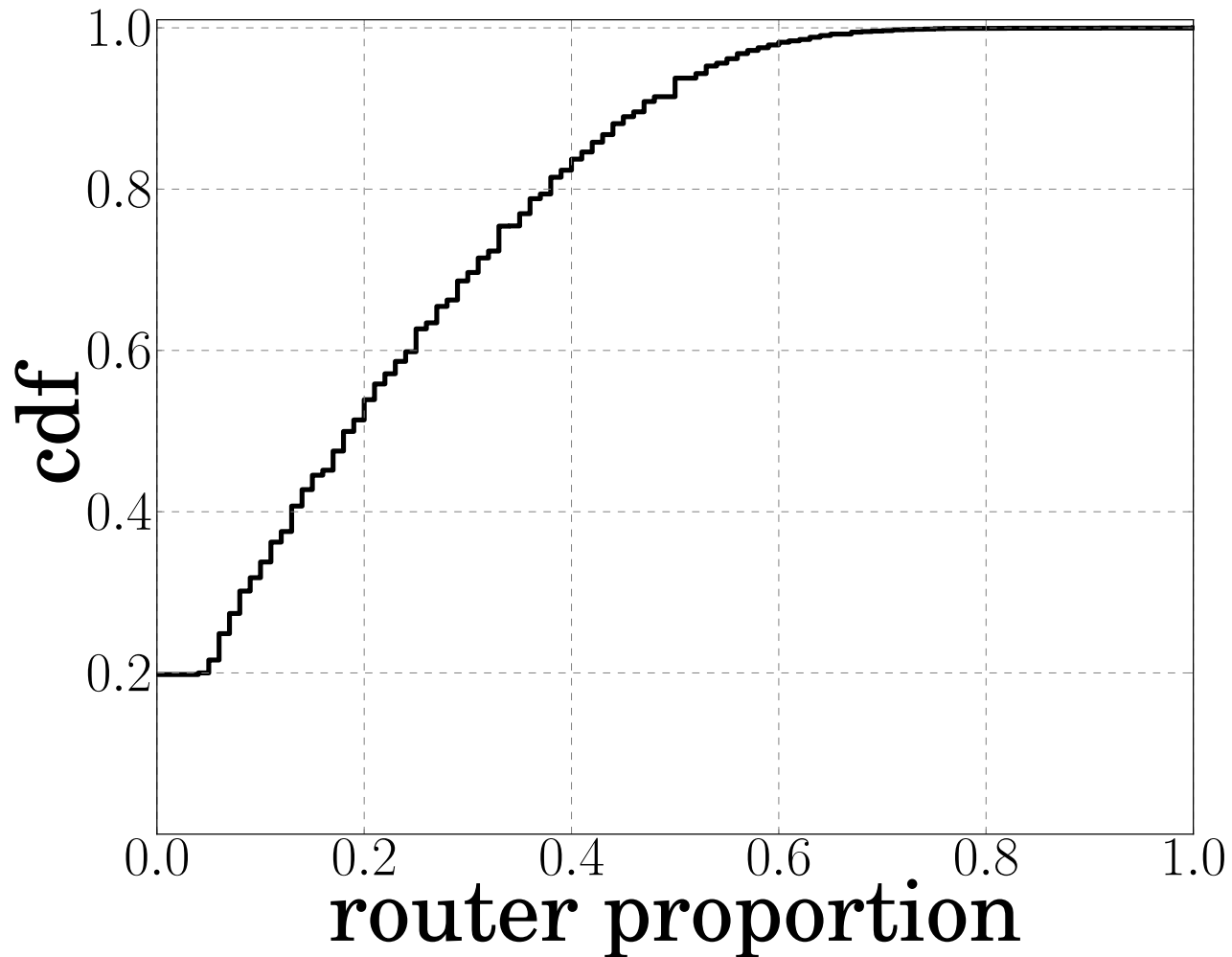
- By default on Linux routers, PaloAlto

Ver	IHL	ToS	Total length	
Identification		Flags	Frag. Offset	
TTL	Protocol	Checksum		
Source IP address				
Destination IP address				
Source port		Destination port		
Sequence number				
Acknowledgment number				
THL	Reserved	Flags	Window	
Checksum		Urgent pointer		
Options				
Payload				

requires that packet.

s, Alcatel

80 % of Internet paths contains at least one RFC1812-capable router



ICMP detection limitations

- Similar to traceroute:
 - Filtering of ICMP
 - Routers throttle or does not send ICMP
- To detect middlebox in front of server, the latter should generate an ICMP.

Outline

- Middleboxes interference
- Detect packet modification with ICMP
- **Tracebox**
- Measurements results

Tracebox

- Uses the previous mechanism to detect middleboxes.
- Implemented in C++ with Lua embedded.
- Libcrafter allows to efficiently describe probes as Scapy.
- Open source and available at <http://www.tracebox.org>
- Supports Linux and Mac OSX

Tracebox

Usage:

```
tracebox [ OPTIONS ] host
```

Options are:

-h	Display this help and exit
-n	Do not resolve IP addresses
-6	Use IPv6 for static probe generated
-u	Use UDP for static probe generated
-d port	Use the specified port for static probe generated. Default is 80.
-i device	Specify a network interface to operate with
-m hops_max	Set the max number of hops (max TTL to be reached). Default is 30
-p probe	Specify the probe to send.
-s script	Run a script.

Probe definition

- SYN probe that contains the window scale option
 - ip{} / tcp{flags=0x2,dst=80} / WSCALE
 - IP / TCP / wscale(9) / NOP
- IPv6/UDP probe with payload
 - IPv6 / udp{dst=5678} / raw('this is a payload')
- Multiple options:
 - ip{} / RR(8) / tcp{dst=80} / mss(1400) / WSCALE / TS

Output example

```
# tracebox -n -p "IP/TCP/MSS/MPCAPABLE/WSCALE" bahn.de
tracebox to 81.200.198.6 (bahn.de): 64 hops max
1: 130.104.228.126 IP::Checksum
2: 130.104.254.229 IP::TTL IP::Checksum
3: 193.191.3.85 IP::TTL IP::Checksum
4: 193.191.16.21 IP::TTL IP::Checksum
5: 195.69.144.123 IP::TTL IP::Checksum
6: 145.254.5.158 IP::TTL IP::Checksum
7: 88.79.13.62 IP::TTL IP::Checksum
8: 81.200.194.234 IP::TTL IP::Checksum
9: 81.200.197.9 IP::TTL IP::Checksum
10: 81.200.198.6 TCP::Checksum IP::TTL IP::Checksum
    TCPOptionMaxSegSize::MaxSegSize
    -TCPOptionMPTCPCapable -TCPOptionWindowScale
```

Output example

```
# tracebox -n -p IP/TCP/MSS/MPCAPABLE/WSCALE bahn.de
tracebox to 81.200.198.6 (bahn.de): 64 hops max
1: 130.104.228.126 IP::Checksum
2: 130.104.254.229 IP::TTL IP::Checksum
3: 193.191.3.85 IP::TTL IP::Checksum
4: 193.191.16.21 IP::TTL IP::Checksum
5: 195.69.144.123 IP::TTL IP::Checksum
6: 145.254.5.158 IP::TTL IP::Checksum
7: 88.79.13.62 IP::TTL IP::Checksum
8: 81.200.194.234 IP::TTL IP::Checksum
9: 81.200.197.9 IP::TTL IP::Checksum
10: 81.200.198.6 TCP::Checksum IP::TTL IP::Checksum
    TCPOptionMaxSegSize::MaxSegSize
    -TCPOptionMPTCPCapable -TCPOptionWindowScale
```

Output example

```
# tracebox -n -p IP/TCP/MSS/MPCAPABLE/WSCALE bahn.de
tracebox to 81.200.198.6 (bahn.de): 64 hops max
1: 130.104.228.126 IP::Checksum
2: 130.104.254.229 IP::TTL IP::Checksum
3: 193.191.3.85 IP::TTL IP::Checksum
4: 193.191.16.21 IP::TTL IP::Checksum
5: 195.69.144.123 IP::TTL IP::Checksum
6: 145.254.5.158 IP::TTL IP::Checksum
7: 88.79.13.62 IP::TTL IP::Checksum
8: 81.200.194.234 IP::TTL IP::Checksum
9: 81.200.197.9 IP::TTL IP::Checksum
10: 81.200.198.6 TCP::Checksum IP::TTL IP::Checksum
TCPOptionMaxSegSize::MaxSegSize
-TCPOptionMPTCPCapable -TCPOptionWindowScale
```

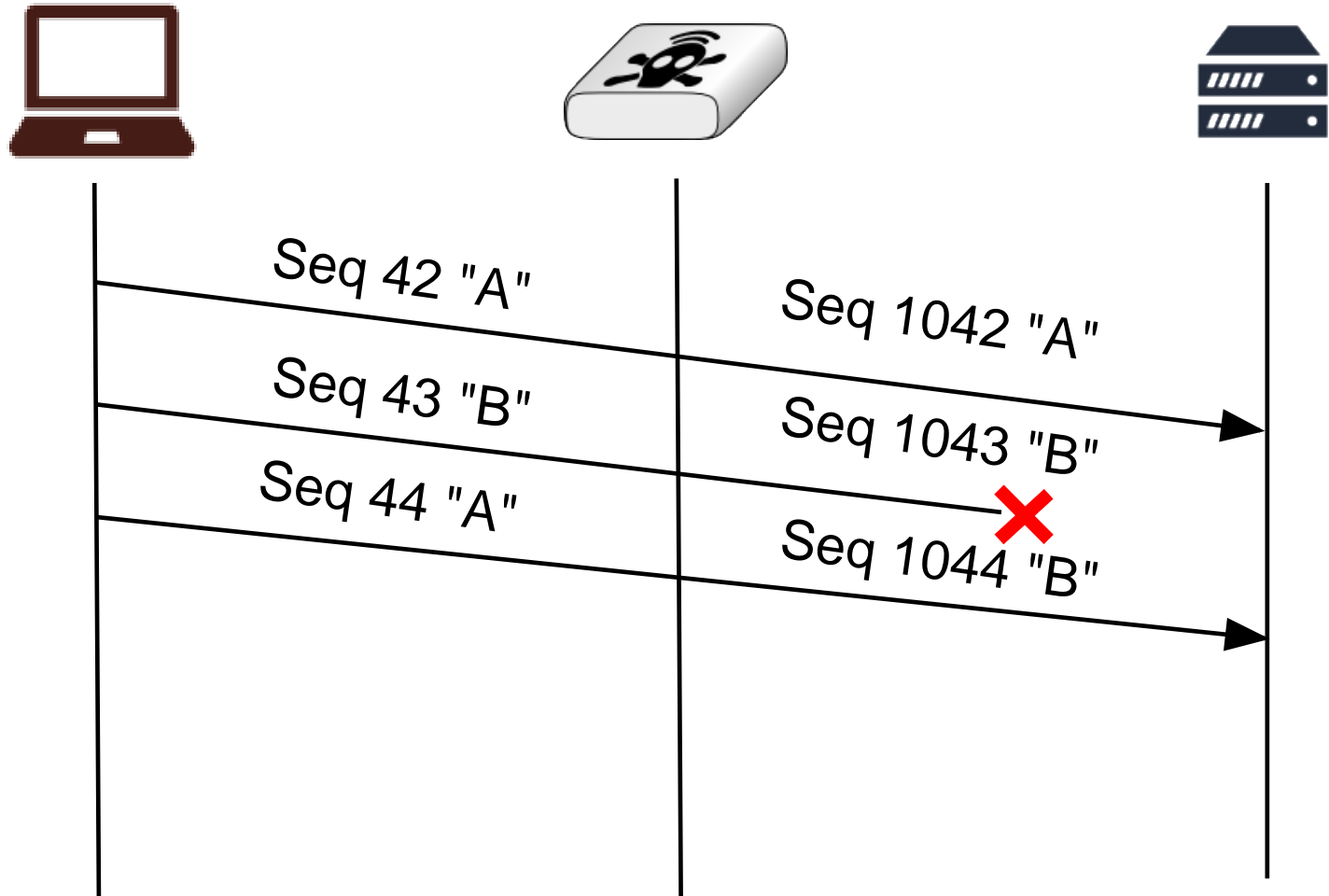
Outline

- Middleboxes interference
- Detect packet modification with ICMP
- Tracebox
- **Measurements results**

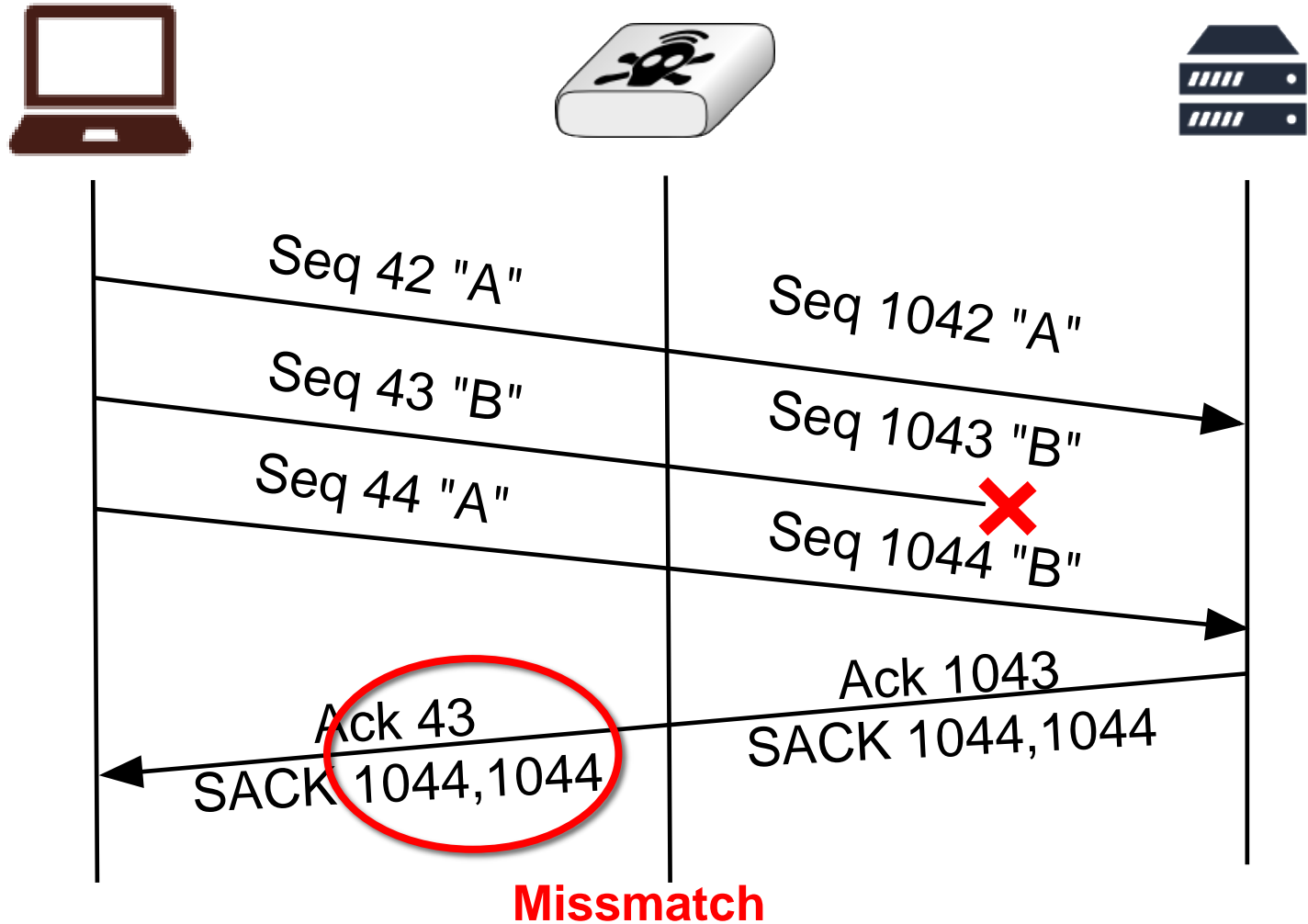
Measurements

- Used PlanetLab to perform experiments
- PlanetLab nodes are supposed to be directly connected to the Internet.
- Sources: 70 vantage points
- Destinations: Top 5000 Alexa

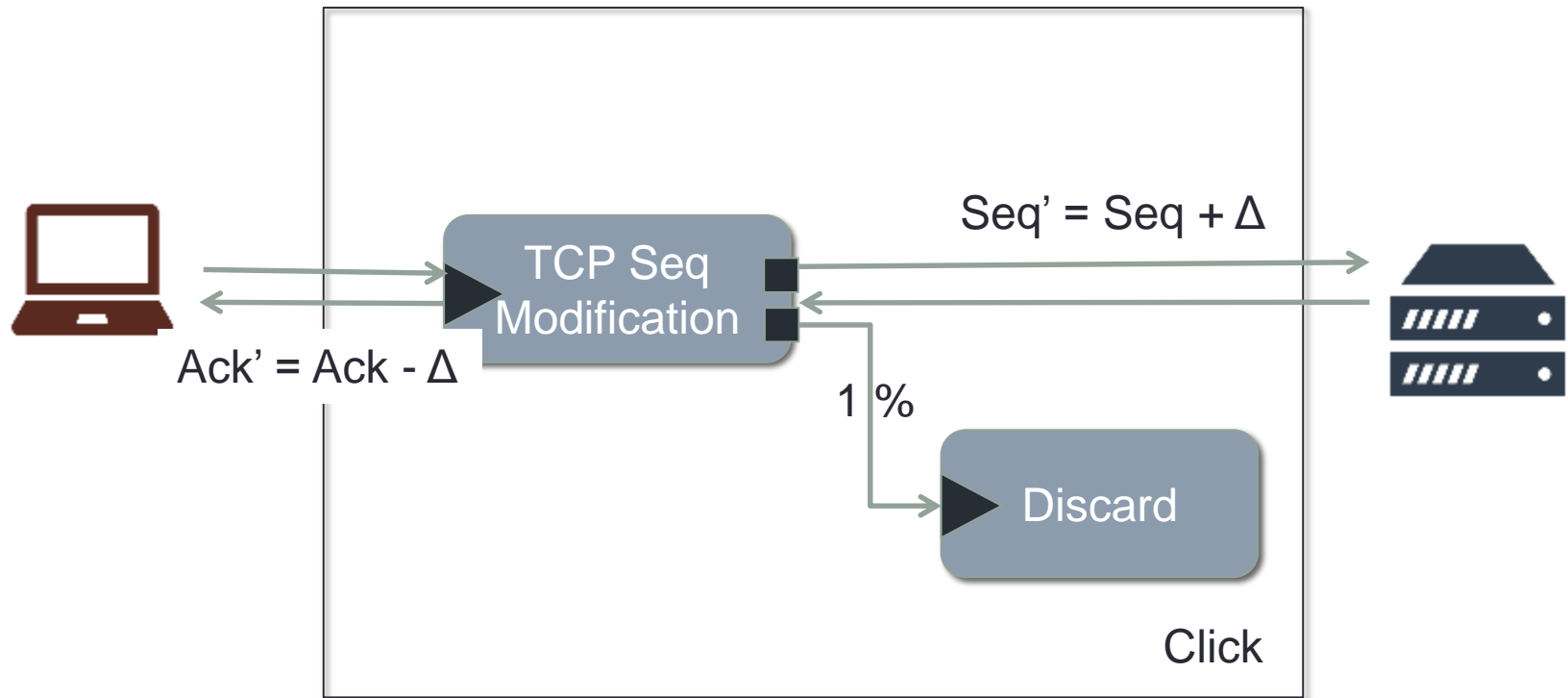
Some middleboxes randomize the TCP sequence number ...



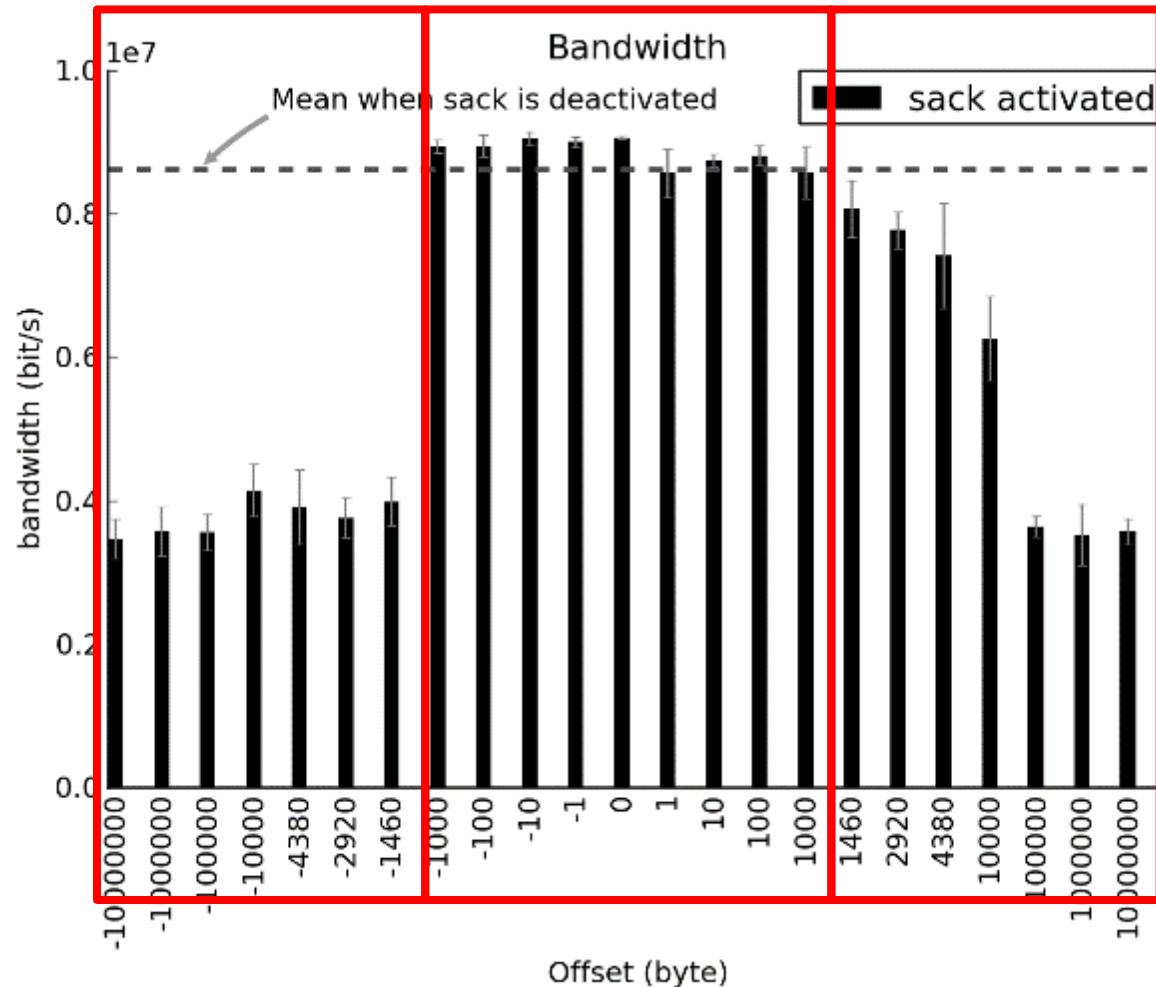
... but does not modify the SACK blocks



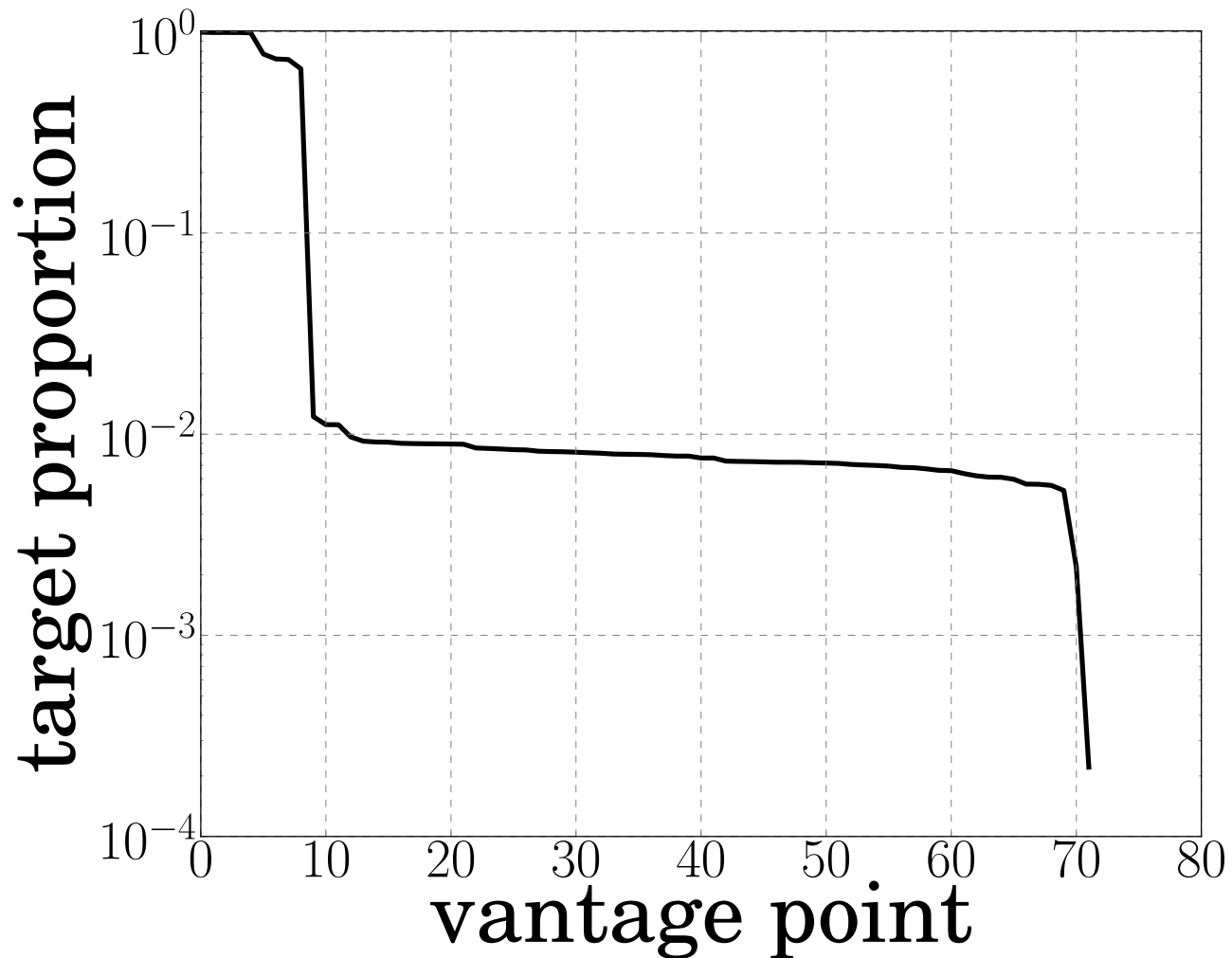
Evaluation of the impact



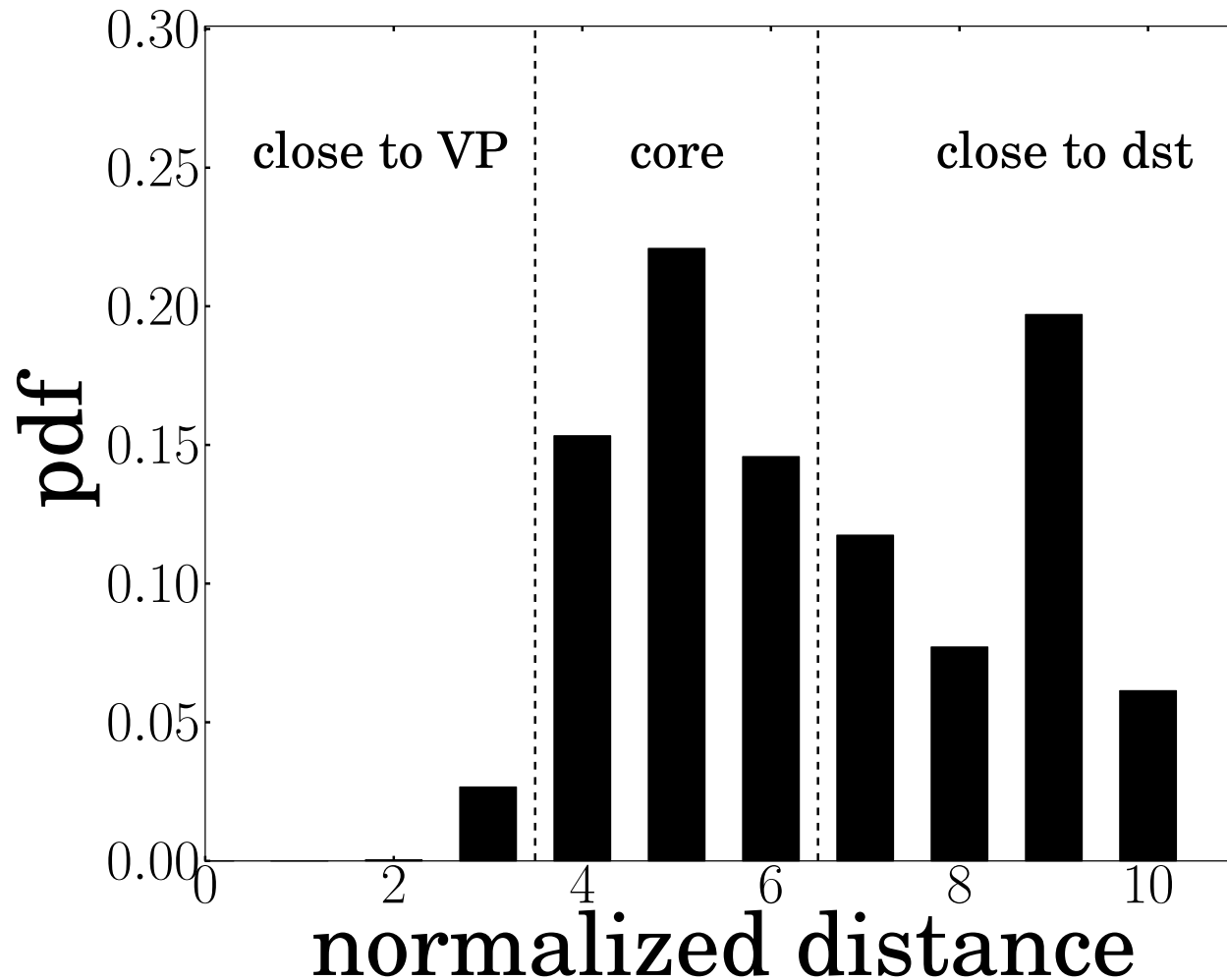
Linux performance significantly drops



Firewall at source modified the MSS



Core network also look at the MSS option and modifies it



Lessons learned

- There exists middleboxes that affect performances and network operators are not always aware of them.
- Tracebox can detect some middleboxes.
- Tracebox could help network operators to debug their network even better with more routers that are RFC1812-capable.

Thank you. Questions ?

gregory.detal@uclouvain.be

<http://www.tracebox.org>