TCP/IP security

CS642: Computer Security

Moving up the network stack

Internet protocol and ICMP

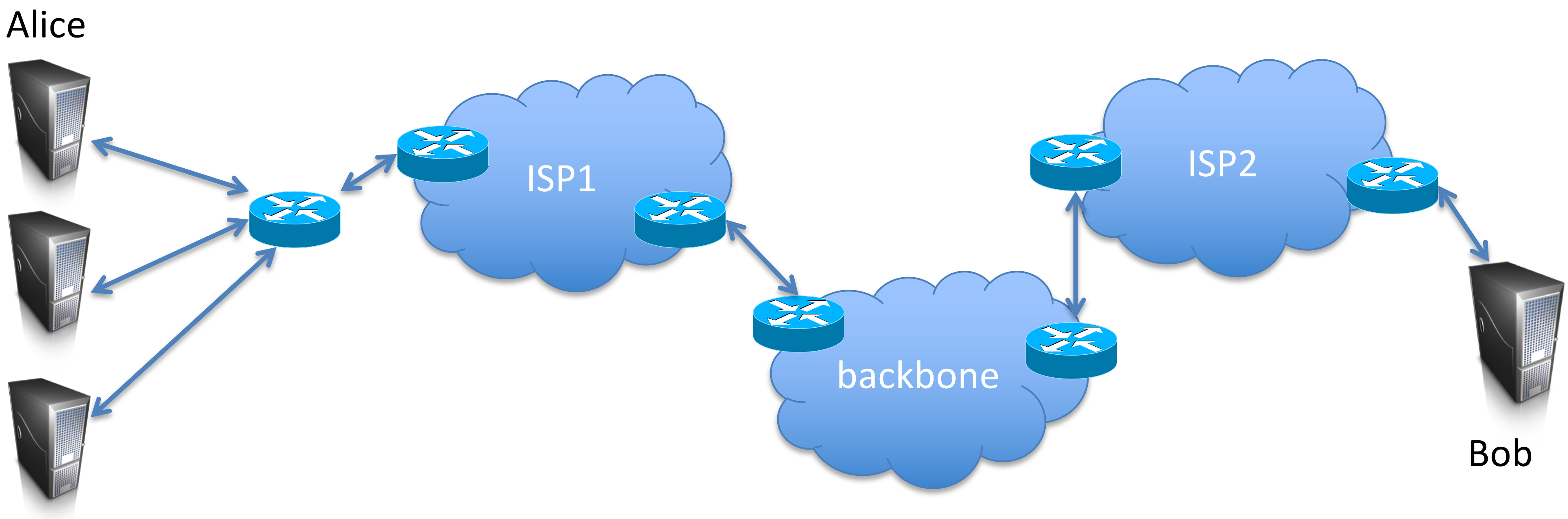
IP spoofing, fragmentation

TCP

Denial of Service

IP traceback, filtering

Internet



Local area network (LAN)

Ethernet

802.11

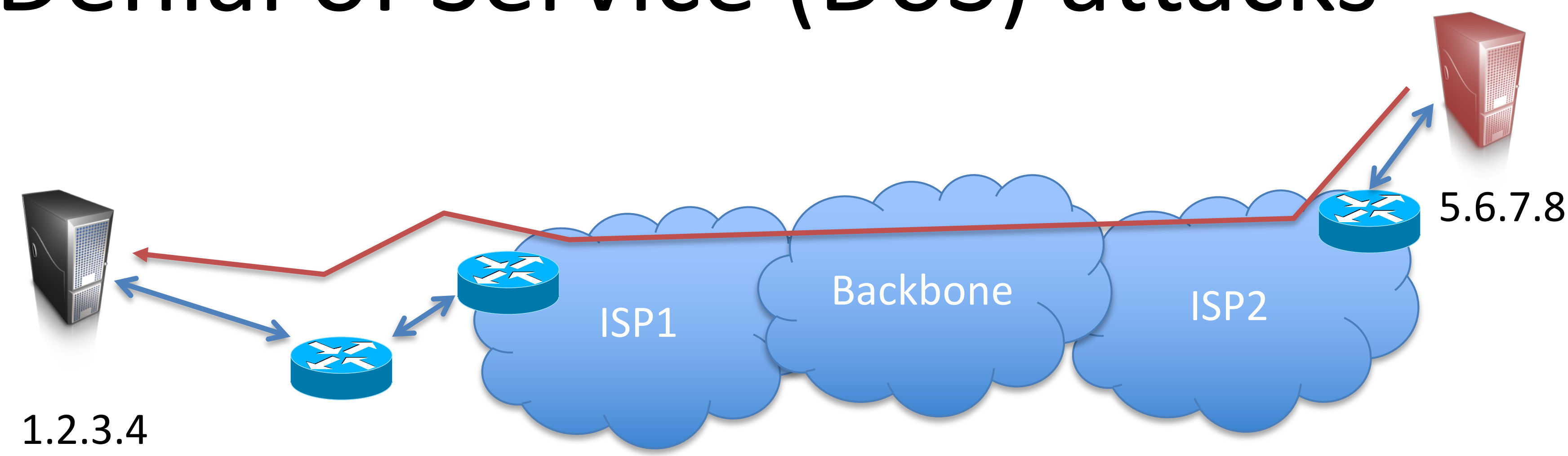
Internet

TCP/IP

BGP (border gateway protocol)

DNS (domain name system)

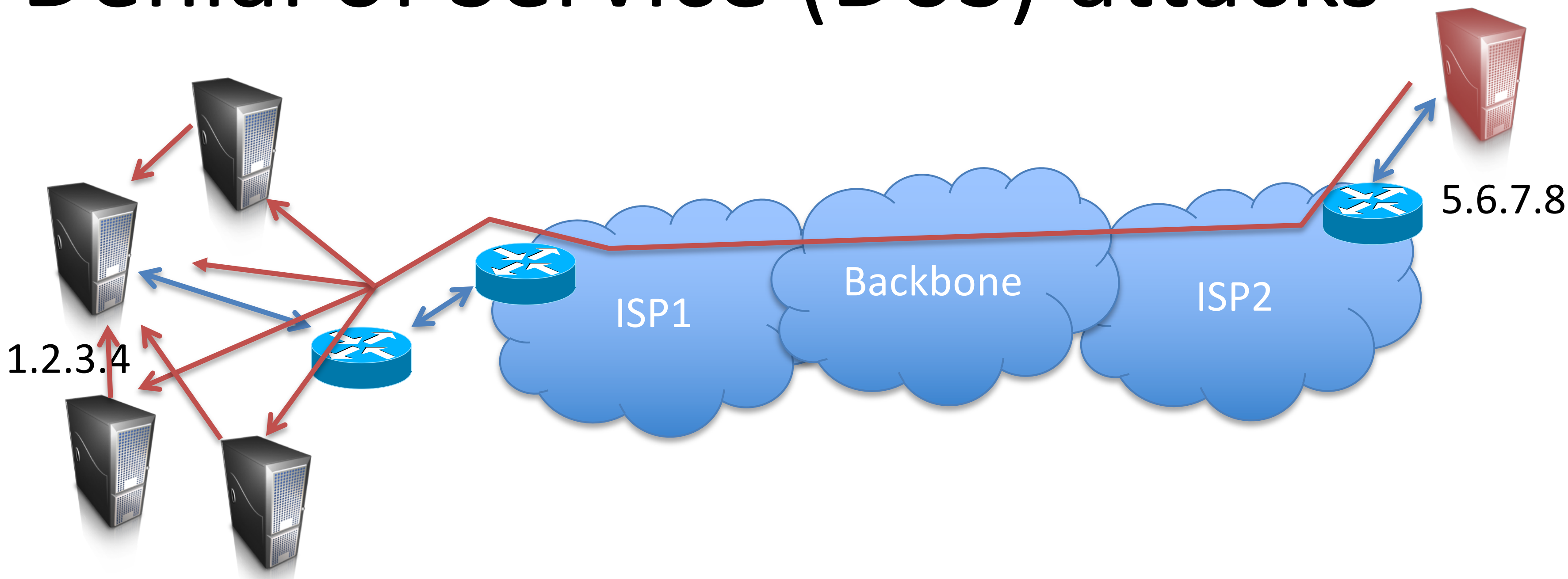
Denial of Service (DoS) attacks



DoS works better when there is ***asymmetry*** between victim and attacker

- Attacker uses few resources to cause
- Victim to consume lots of resources

Denial of Service (DoS) attacks

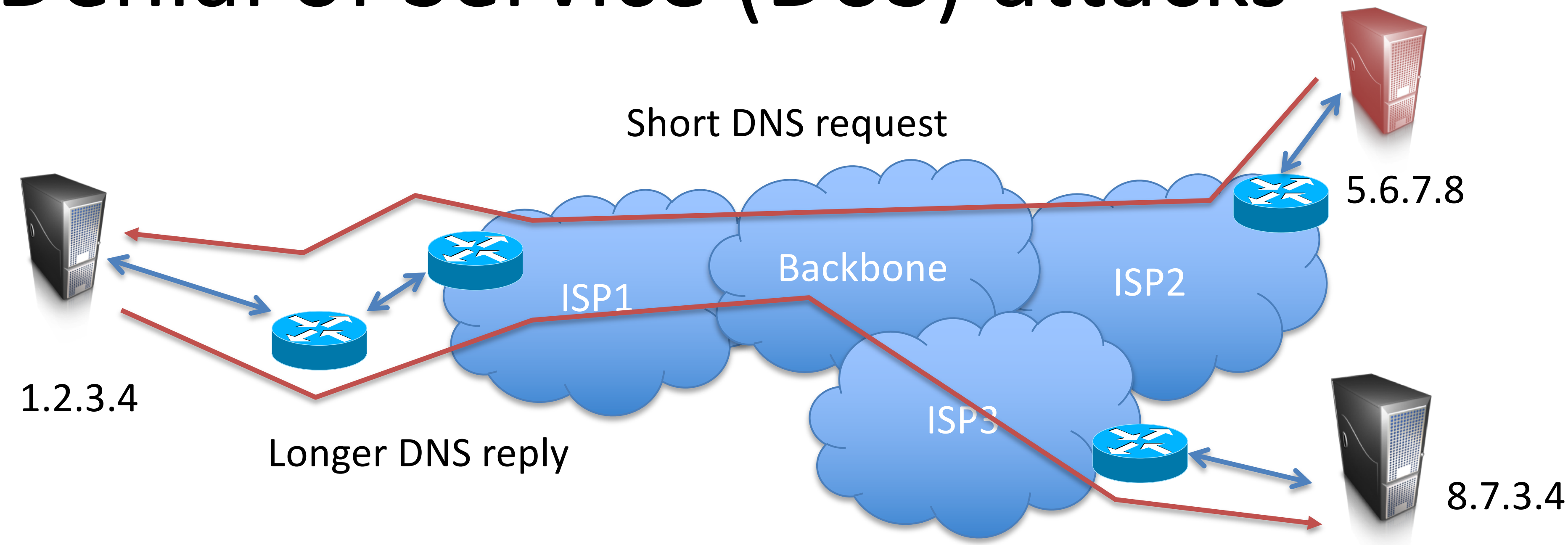


DoS works better when there is ***asymmetry*** between victim and attacker

- Attacker uses few resources to cause
- Victim to consume lots of resources

Old example: Smurf attack
Router allows attacker to send broadcast ICMP ping on network. Attacker spoofs SRC address to be 1.2.3.4

Denial of Service (DoS) attacks



DoS works better when there is ***asymmetry*** between victim and attacker

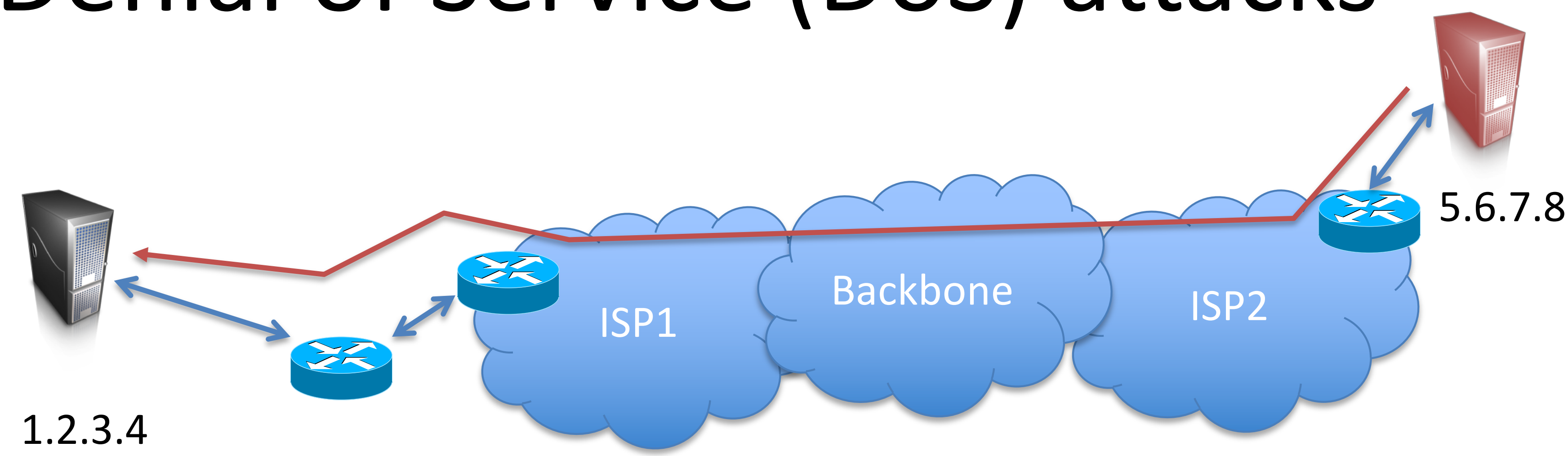
- Attacker uses few resources to cause
- Victim to consume lots of resources

More recent: DNS reflection attacks

Send DNS request w/ spoofed target IP (~65 byte request)

DNS replies sent to target (~512 byte response)

Denial of Service (DoS) attacks



DoS works better when there is ***asymmetry*** between victim and attacker

- Attacker uses few resources to cause
- Victim to consume lots of resources

Big asymmetry: ping of death

A single packet that causes crash on remote system

Early on: ping packet with size > 65,535

IPv4 fragmenting

Ethernet frame
containing
IP datagram

IP allows datagrams of size from
20 bytes up to 65535 bytes

Some link layers only allow MTU of 1500 bytes

IP figures out MTU of next link, and fragments packet if necessary into smaller chunk

IPv4 fragmenting

ENet hdr	IP hdr	data	ENet tlr

Ethernet frame containing IP datagram

4-bit version	4-bit hdr len	8-bit type of service	16-bit total length (in bytes)
16-bit identification		3-bit flags	13-bit fragmentation offset
8-bit time to live (TTL)	8-bit protocol	16-bit header checksum	
32-bit source IP address			
32-bit destination IP address			
options (optional)			

IPv4 fragmenting

ENet hdr	IP hdr	data	ENet tlr

Ethernet frame containing IP datagram

16-bit identification	3-bit flags	13-bit fragmentation offset

Source-specified “unique” number identifying datagram

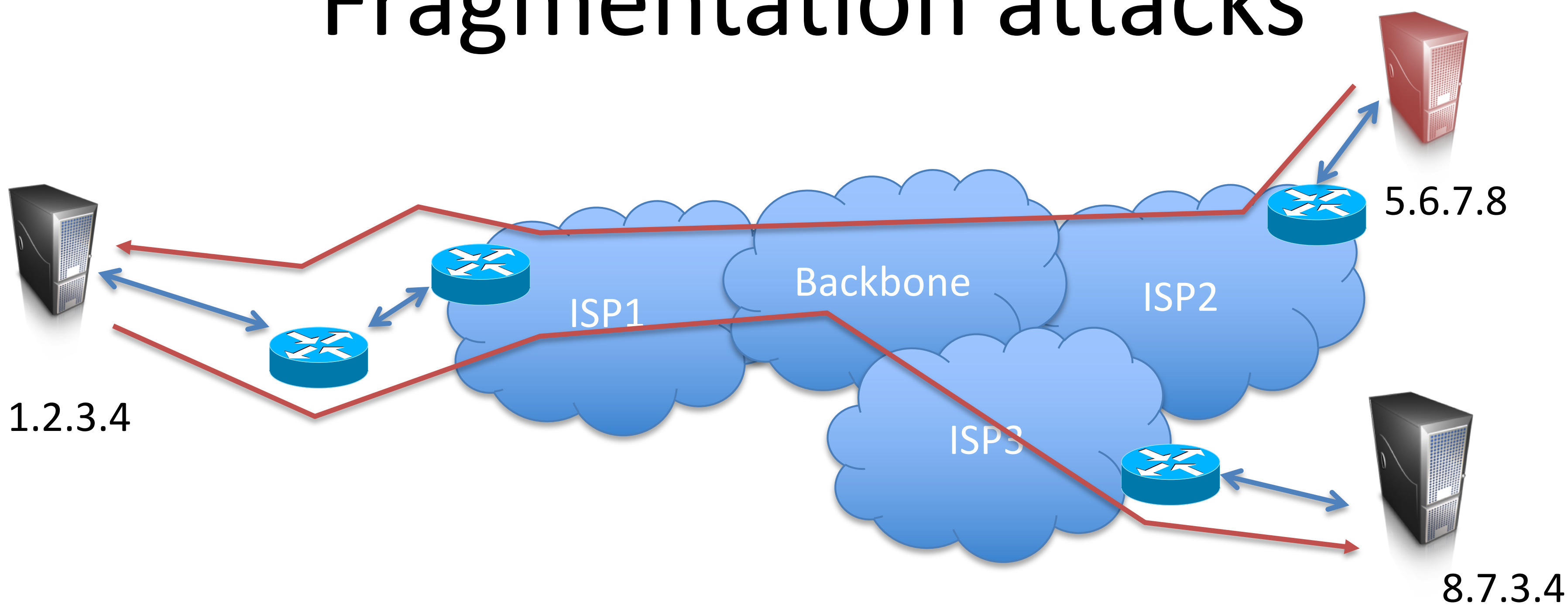
Fragment offset in 8-byte units

Flags:
0 b1 b2

where b1 = May Fragment (0) / Don’t Fragment (1)
where b2 = Last Fragment (0) / More Fragments (1)

What is the problem?

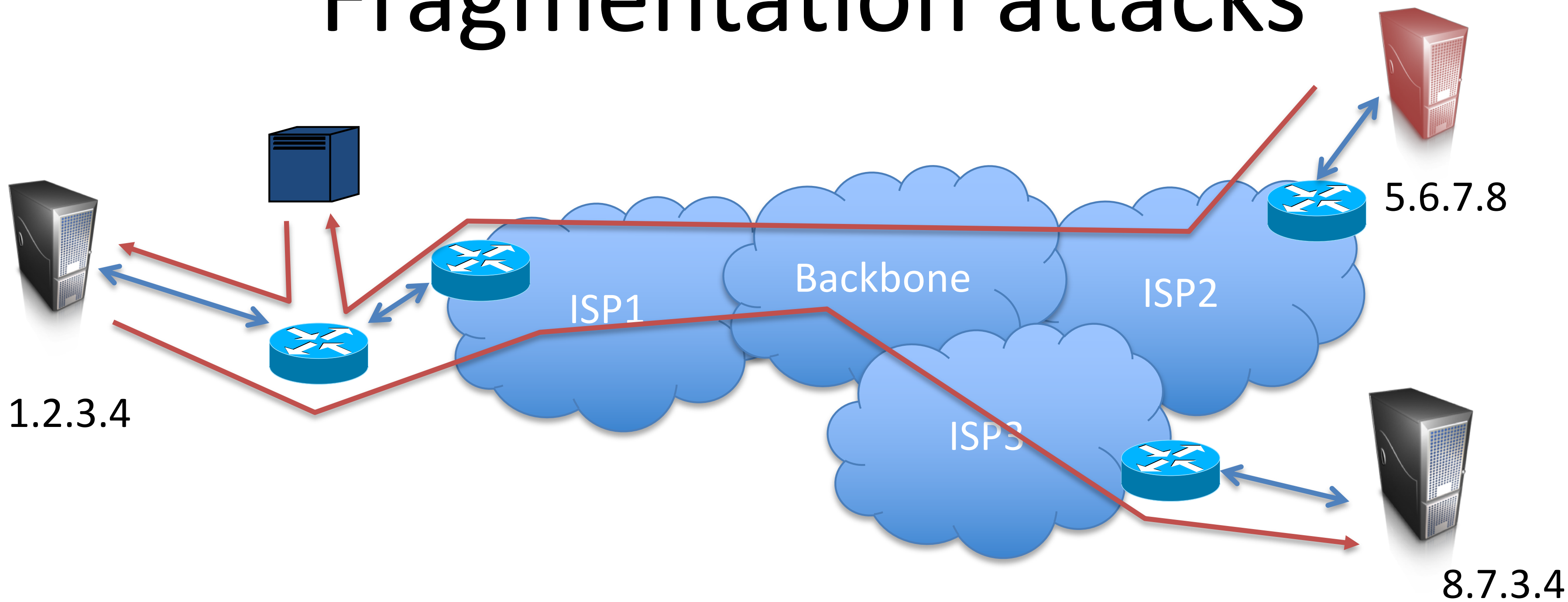
Fragmentation attacks



Fragmentation abused in lots of vulnerabilities:

- Ping of death: allows sending 65,536 byte packet, overflows buffer.
- Teardrop DoS: mangled fragmentation crashes reconstruction code (Set offsets so that two packets have overlapping data)

Fragmentation attacks



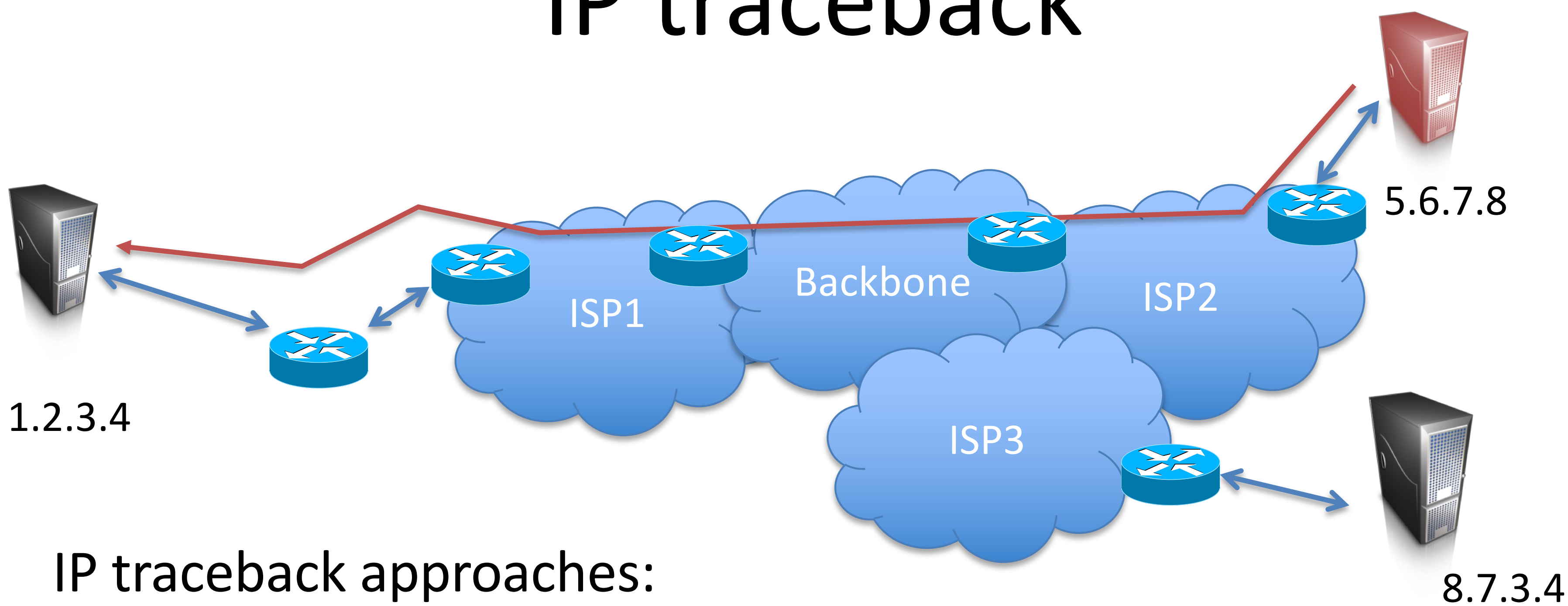
Fragmentation abused in lots of vulnerabilities:

- Ping of death: allows sending 65,536 byte packet, overflows buffer.
- Teardrop DoS: mangled fragmentation crashes reconstruction code (Set offsets so that two packets have overlapping data)
- Avoiding IDS systems: IDS scans packets for exploit strings; add random data into packets, overwrite later during reconstruction due to overlapping fragments

Dealing with spoofing: IP traceback

- Spoofed IPs means we cannot know where packets came from
- IP traceback is problem of determining the origination of one or more packets

IP traceback



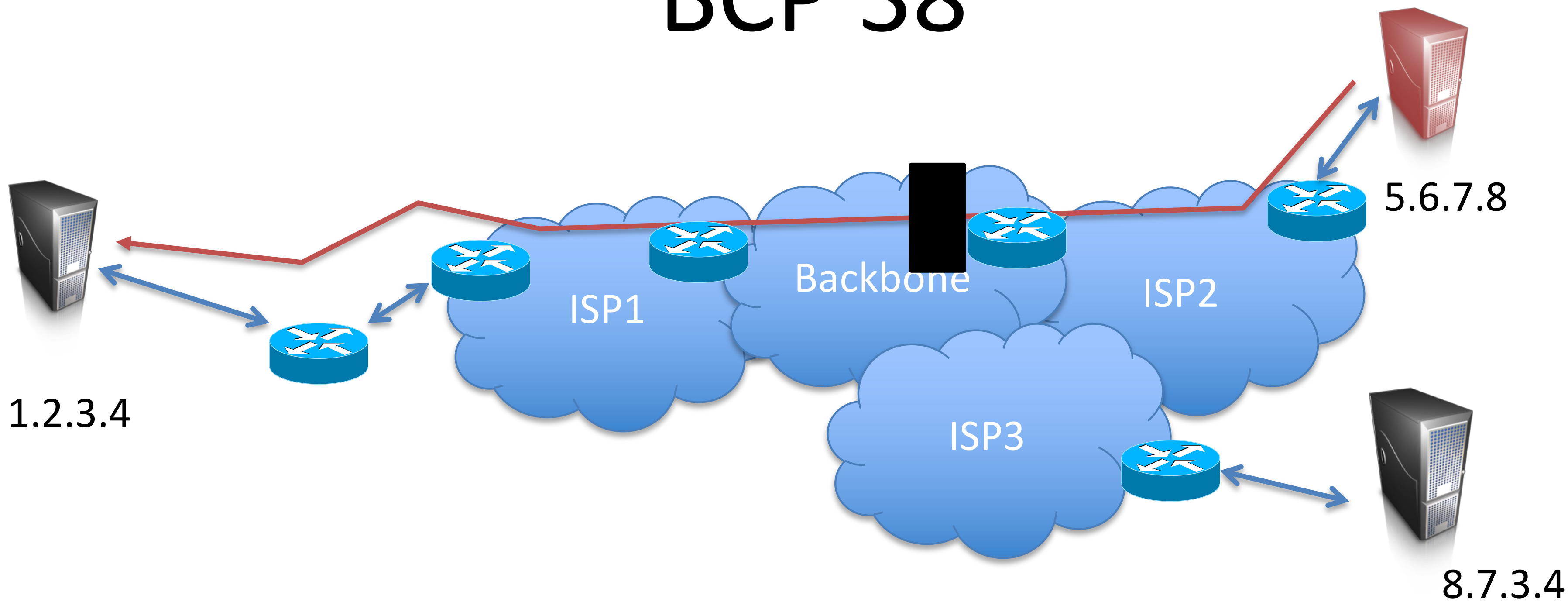
IP traceback approaches:

- Logging – each router keeps logs of packets going by
- Input debugging – feature of routers allowing filtering egress port traffic based on ingress port. Associate egress with ingress
- Controlled flooding – mount your own DoS on links selectively to see how it affects malicious flood
- Marking – router probabilistically marks packets with info
- ICMP traceback – router probabilistically sends ICMP packet with info to destination

Dealing with spoofing: BCP 38

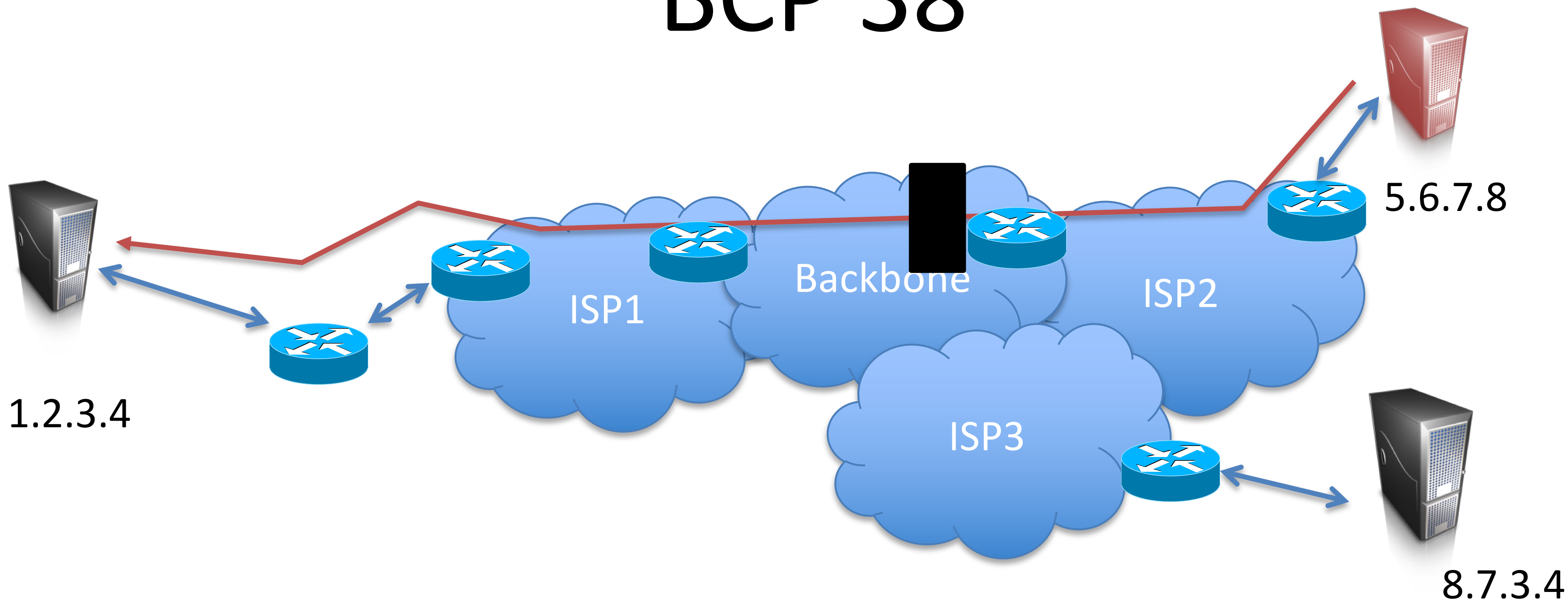
- Spoofed IPs means we cannot know where packets came from
- BCP 38 (RFC 2827): upstream ingress filtering to drop spoofed packets

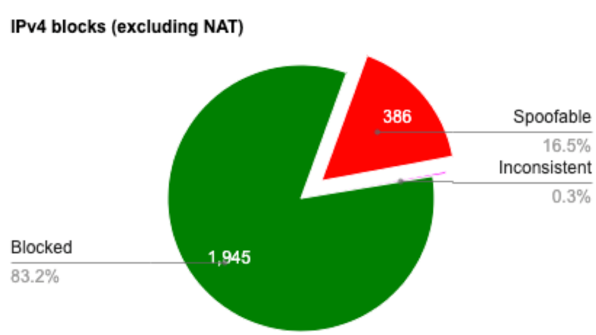
BCP 38



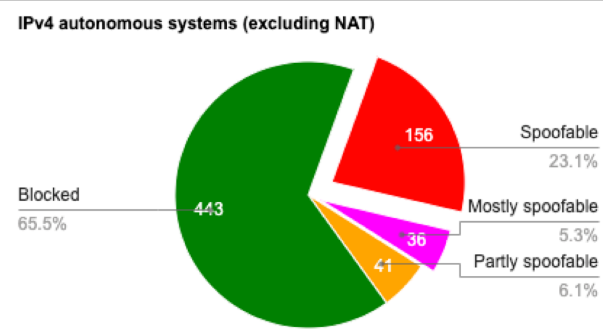
Before forwarding on packets, check at ingress that source IP legitimate

BCP 38





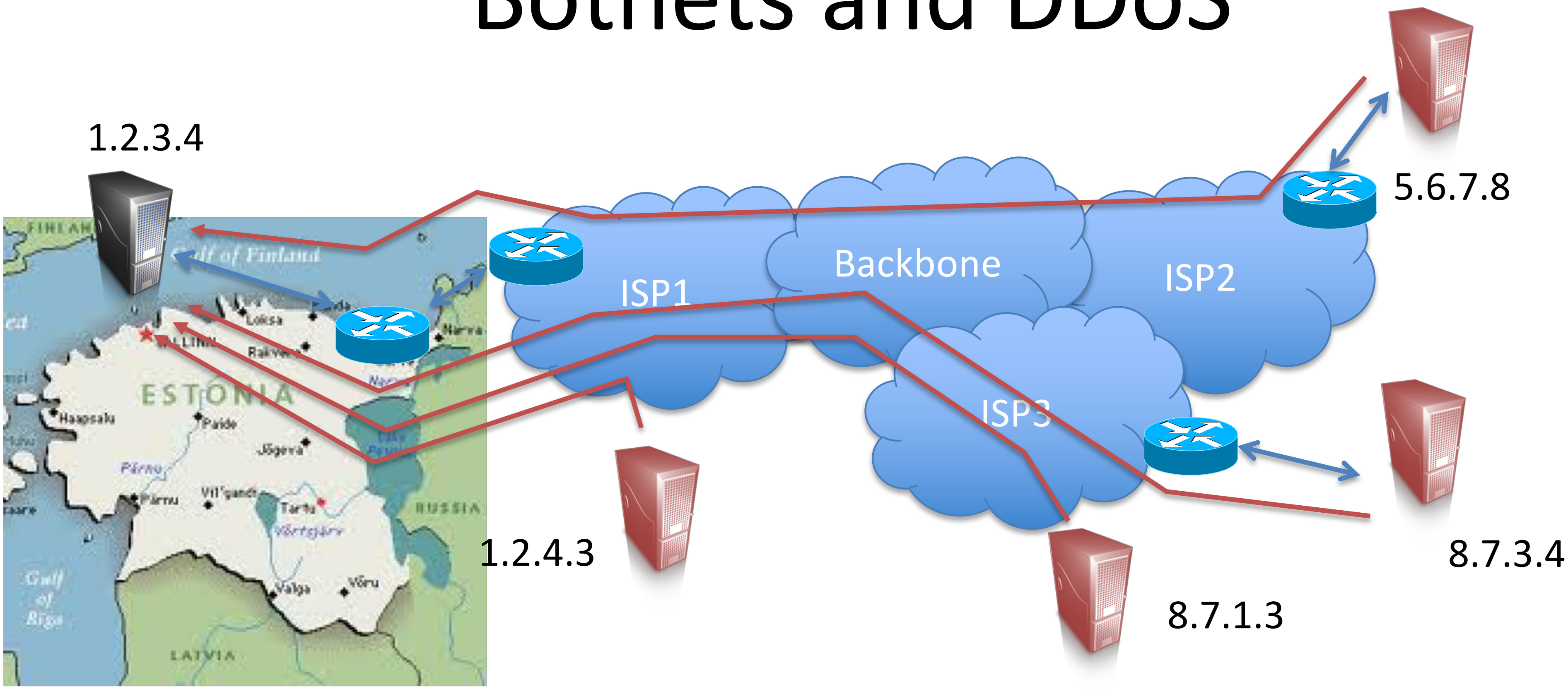
Status	Count
Spoofable	386
Inconsistent	7
Blocked	1945



Status	Count
Spoofable	156
Mostly spoofable	36
Partly spoofable	41
Blocked	443

https://spoofer.caida.org/summary.php

Botnets and DDoS



April 27, 2007

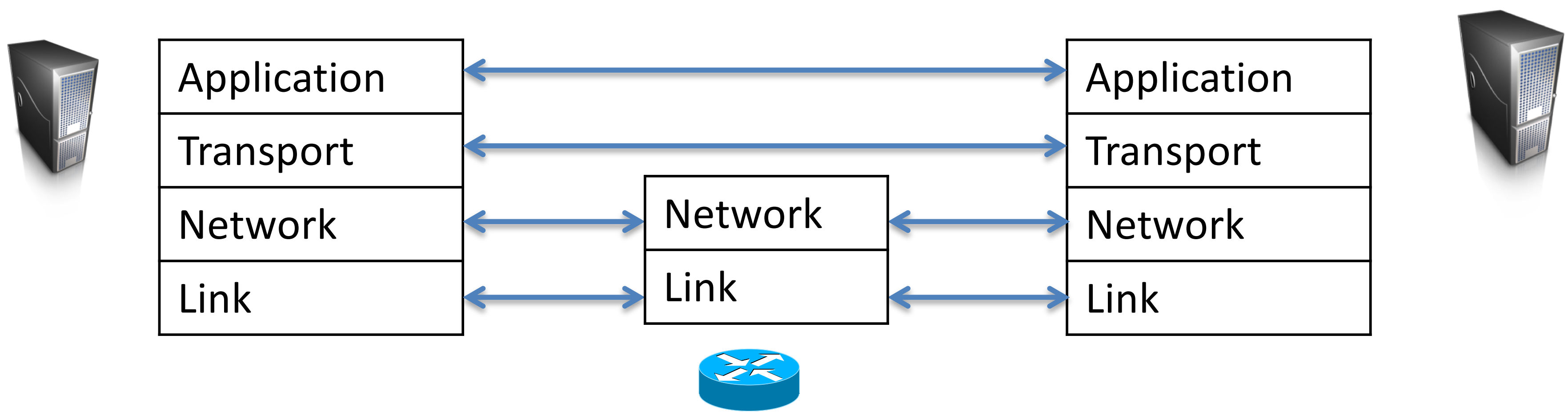
Continued for weeks, with varying levels of intensity

Government, banking, news, university websites

Government shut down international Internet connections

Internet protocol stack

Application	HTTP, FTP, SMTP, SSH, etc.
Transport	TCP, UDP
Network	IP, ICMP, IGMP
Link	802x (802.11, Ethernet)



TCP (transport control protocol)

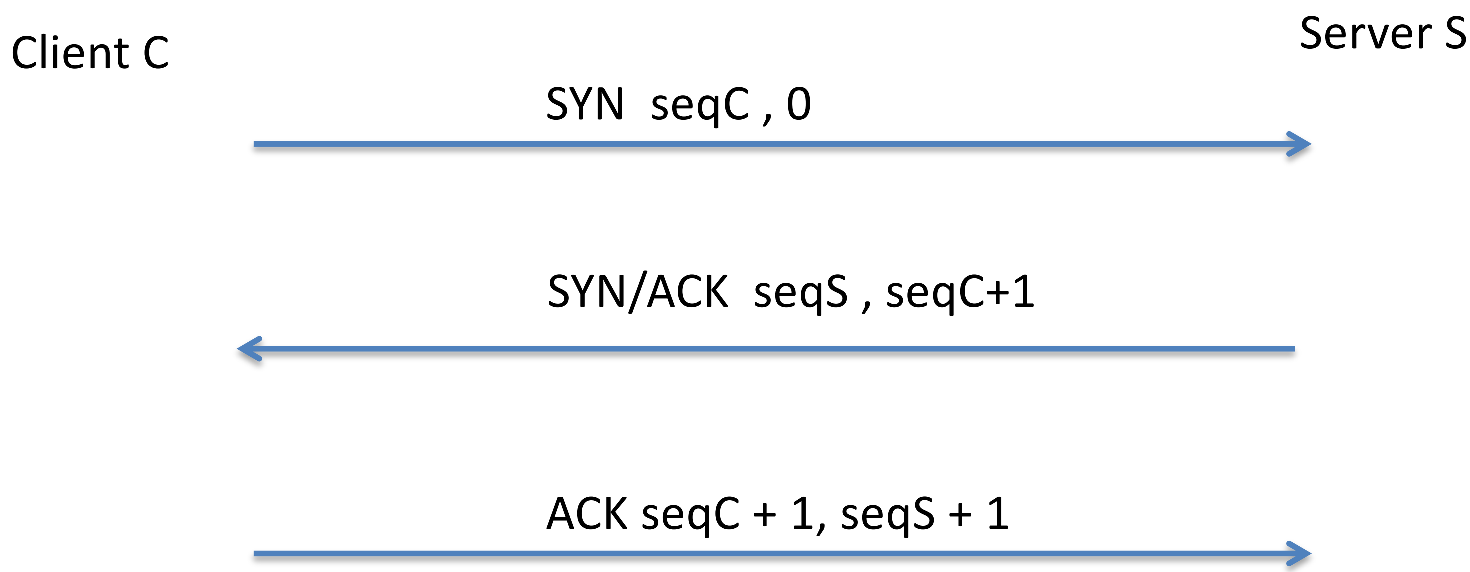
- Connection-oriented
  - state initialized during handshake and maintained
- Reliability is a goal
  - generates segments
  - timeout segments that aren’t ack’d
  - checksums headers,
  - reorders received segments if necessary
  - flow control

TCP (transport control protocol)

IP hdr	TCP hdr	data

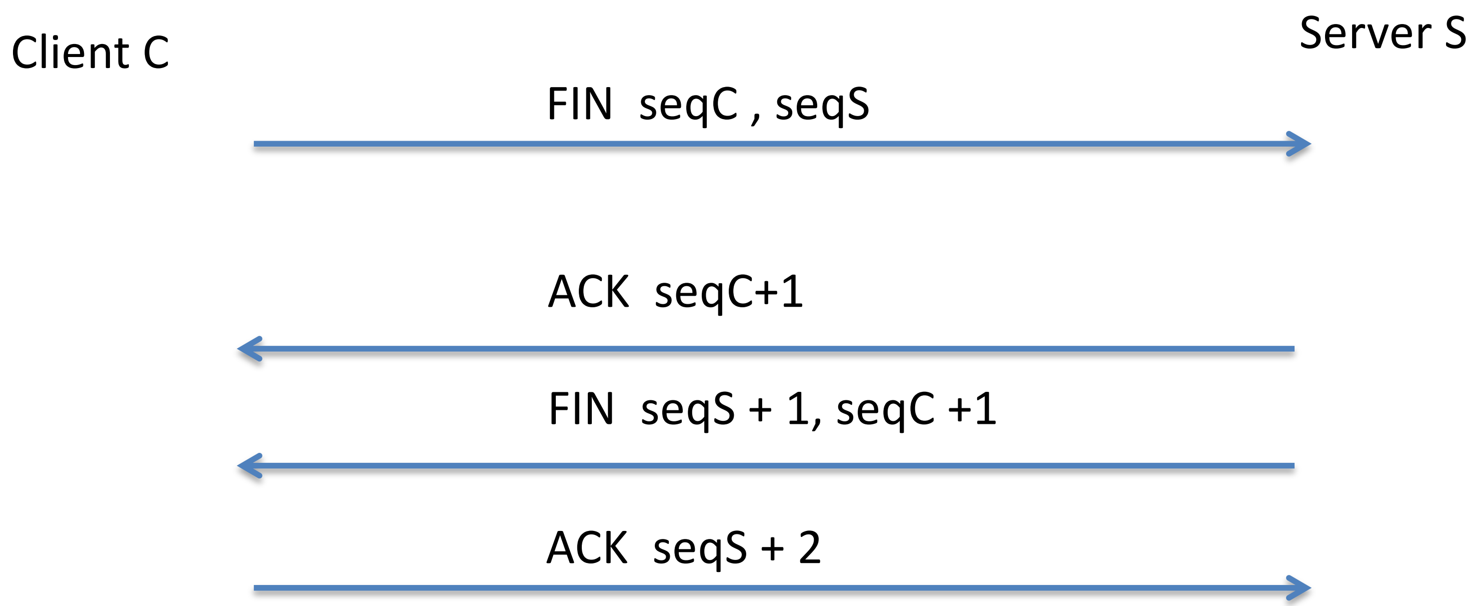
<table>
<tr><td colspan="3">16-bit
source port number</td><td>16-bit
destination port number</td></tr>
<tr><td colspan="4">32-bit
sequence number</td></tr>
<tr><td colspan="4">32-bit
acknowledgement number</td></tr>
<tr><td>4-bit
hdr len</td><td>6-bits
reserved</td><td>6-bits
flags</td><td>16-bit
window size</td></tr>
<tr><td colspan="3">16-bit
TCP checksum</td><td>16-bit
urgent pointer</td></tr>
<tr><td colspan="4">options (optional)</td></tr>
<tr><td colspan="4">data (optional)</td></tr>
</table>

TCP handshake



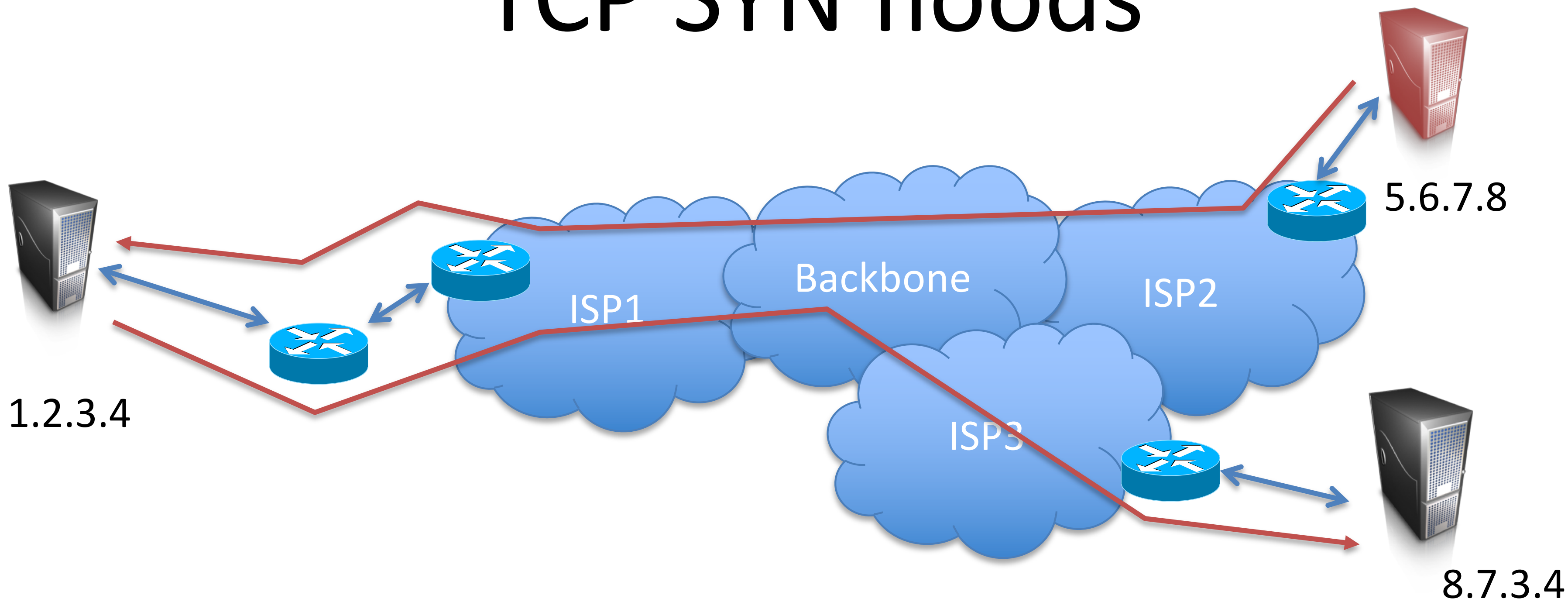
SYN = syn flag set
ACK = ack flag set
x,y = x is sequence #, y is acknowledge #

TCP teardown



SYN = syn flag set
ACK = ack flag set
x,y = x is sequence #, y is acknowledge #

TCP SYN floods



Send lots of TCP SYN packets to 1.2.3.4

- 1.2.3.4 maintains state for each SYN packet for some amount window of time
- If 5.6.7.8 sets SRC IP to be 8.7.3.4, what does 8.7.3.4 receive?

TCP handshake

Client C

Server S

SYN seqC , 0

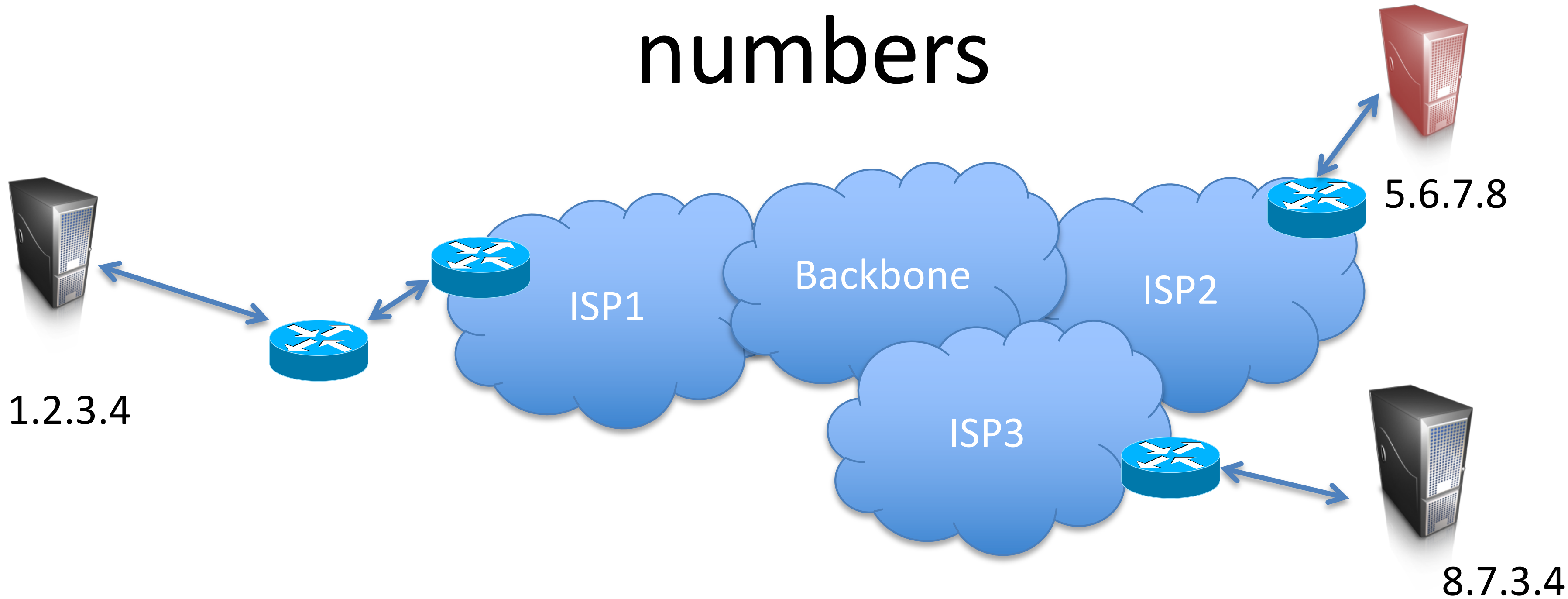
SYN/ACK seqS , secC+1

ACK seqC + 1, seqS + 1

How are secC and seqS selected?

Initial sequence numbers must vary over time so that different connections don't get confused

Predictable sequence numbers

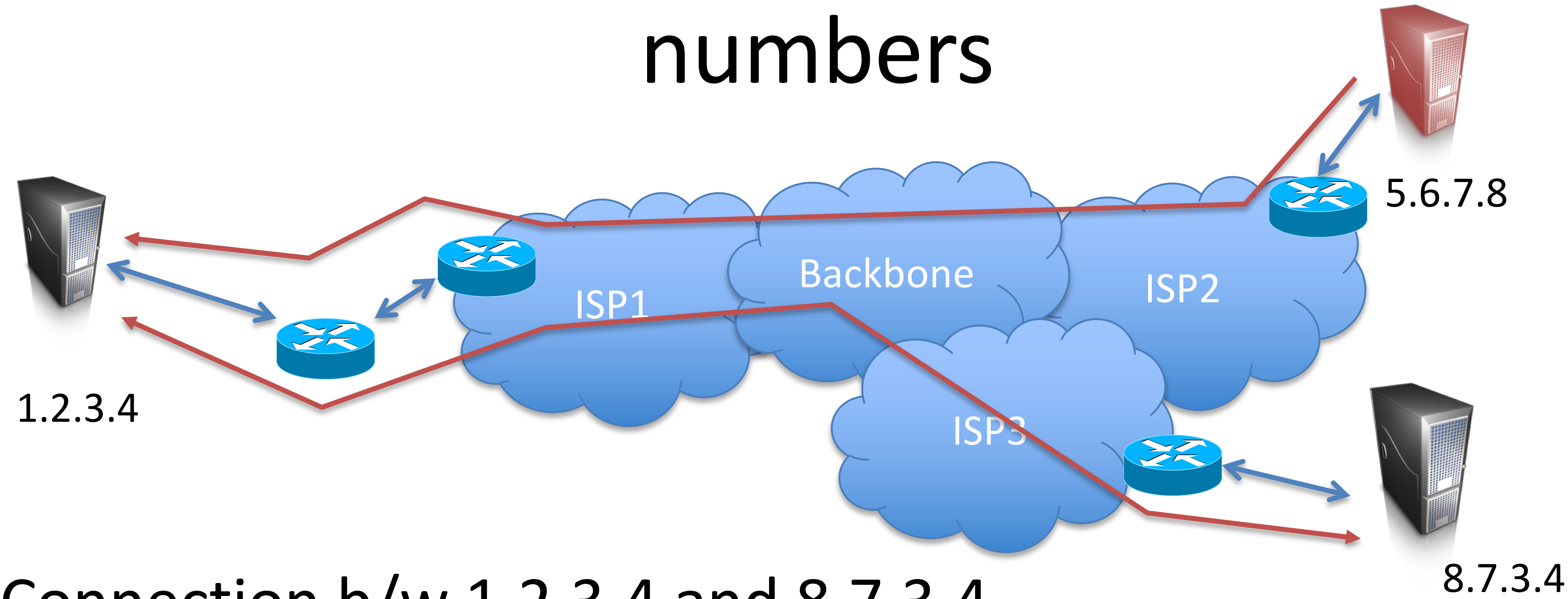


4.4BSD used predictable initial sequence numbers (ISNs)

- At system initialization, set ISN to 1
- Increment ISN by 64,000 every half-second

What can a clever attacker do?

Predictable sequence numbers



Connection b/w 1.2.3.4 and 8.7.3.4

Forge a FIN packet from 8.7.3.4 to 1.2.3.4

src: 8.7.3.4
dst: 1.2.3.4

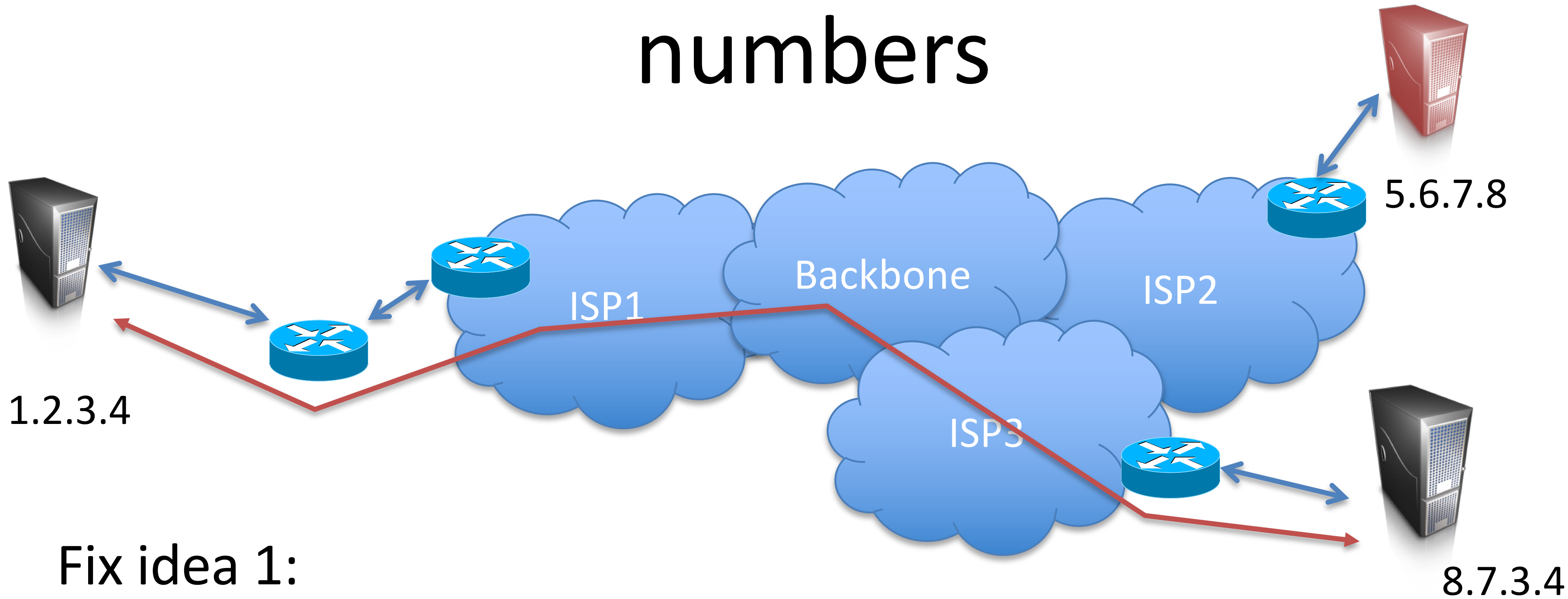
seq#(8.7.3.4)
FIN

Forge some application-layer packet from 8.7.3.4 to 1.2.3.4

src: 8.7.3.4
dst: 1.2.3.4

seq#(8.7.3.4)
“rsh rm –rf /”

Predictable sequence numbers



Fix idea 1:

- Random ISN at system startup
- Increment by 64,000 each half second

Better fix:

- Random ISN for every connection

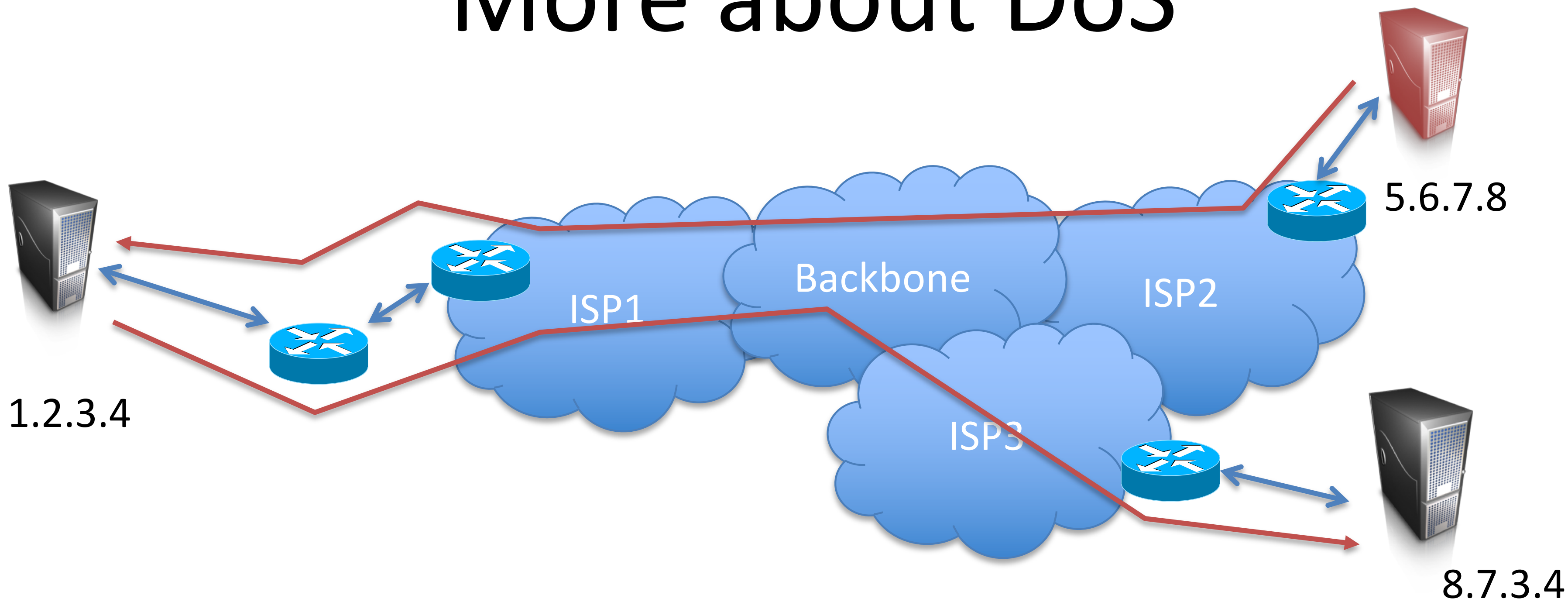
Still issues:

- Any FIN accepted with seq# in receive window: 2^{17} attempts

TCP/IP security: other issues

- Congestion control abuse
  - can allow cheaper DoS
- No crypto
  - We covered TLS
- BGP routing
  - we'll talk about later
- DNS (mapping from IP to domain names)
  - We'll talk about later

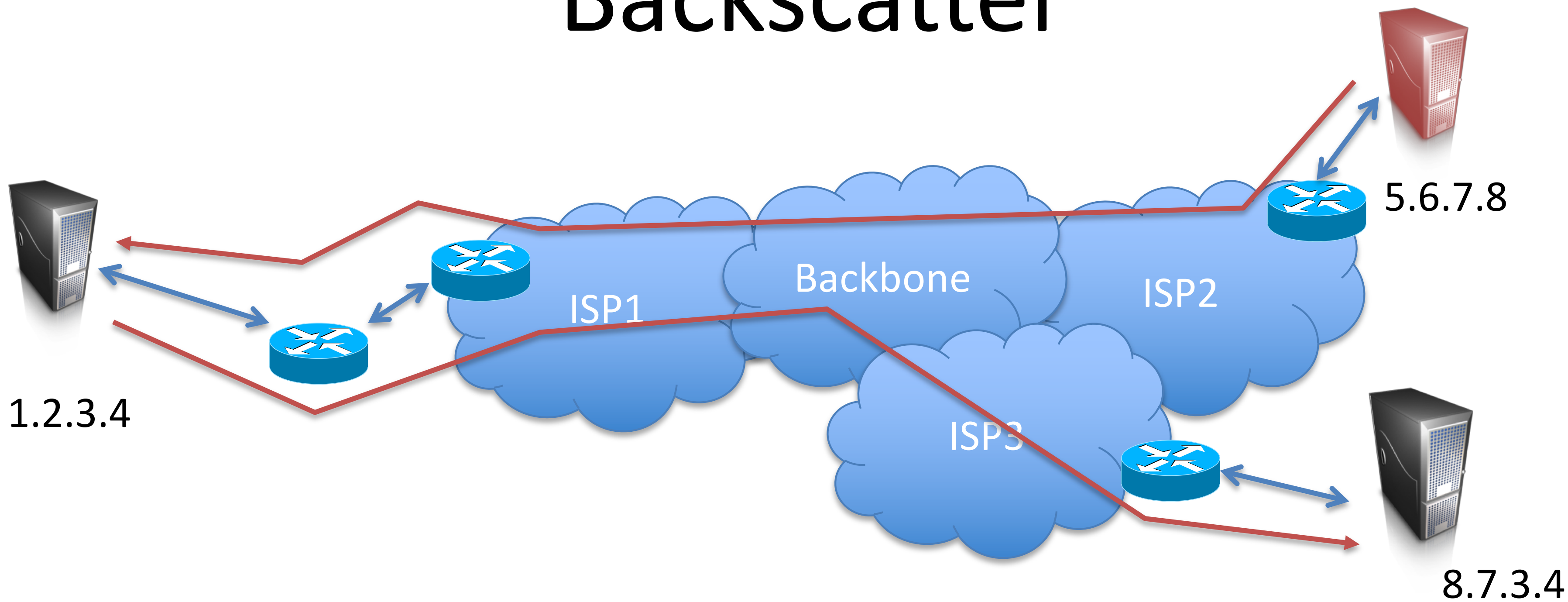
More about DoS



DoS is still a big problem

How big?

Backscatter



Can we measure the level of DoS attacks on Internet?

- If we can measure spurious packets at 8.7.3.4, we might infer something about DoS at 1.2.3.4

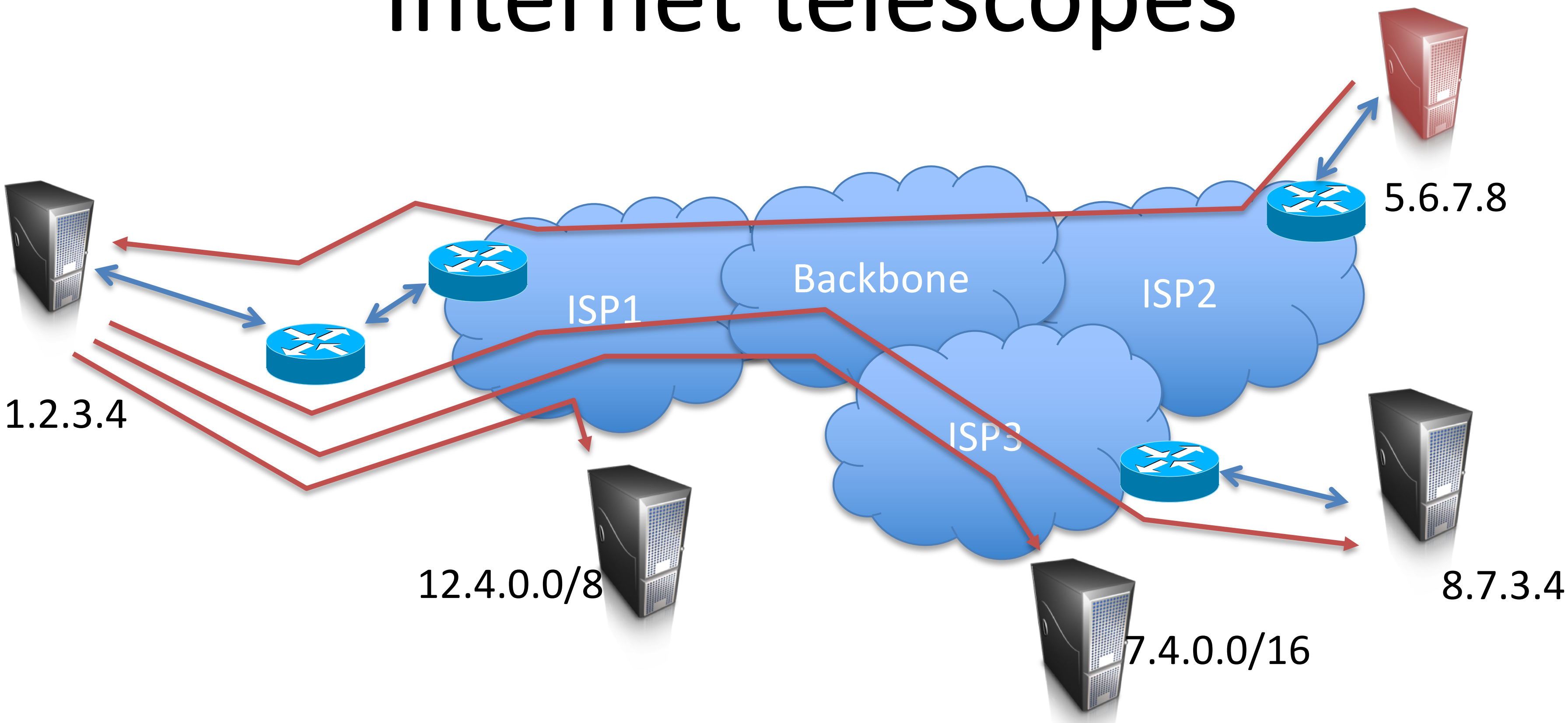
Types of responses to floods

Packet sent	Response from victim
TCP SYN (to open port)	TCP SYN/ACK
TCP SYN (to closed port)	TCP RST (ACK)
TCP ACK	TCP RST (ACK)
TCP DATA	TCP RST (ACK)
TCP RST	no response
TCP NULL	TCP RST (ACK)
ICMP ECHO Request	ICMP Echo Reply
ICMP TS Request	ICMP TS Reply
UDP pkt (to open port)	protocol dependent
UDP pkt (to closed port)	ICMP Port Unreach
...	...

Table 1: A sample of victim responses to typical attacks.

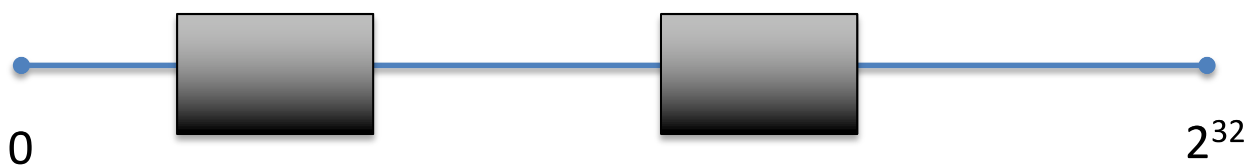
From Moore et al., “Inferring Internet Denial-of-Service Activity”

Internet telescopes



Setup some computers to watch traffic sent to darknets

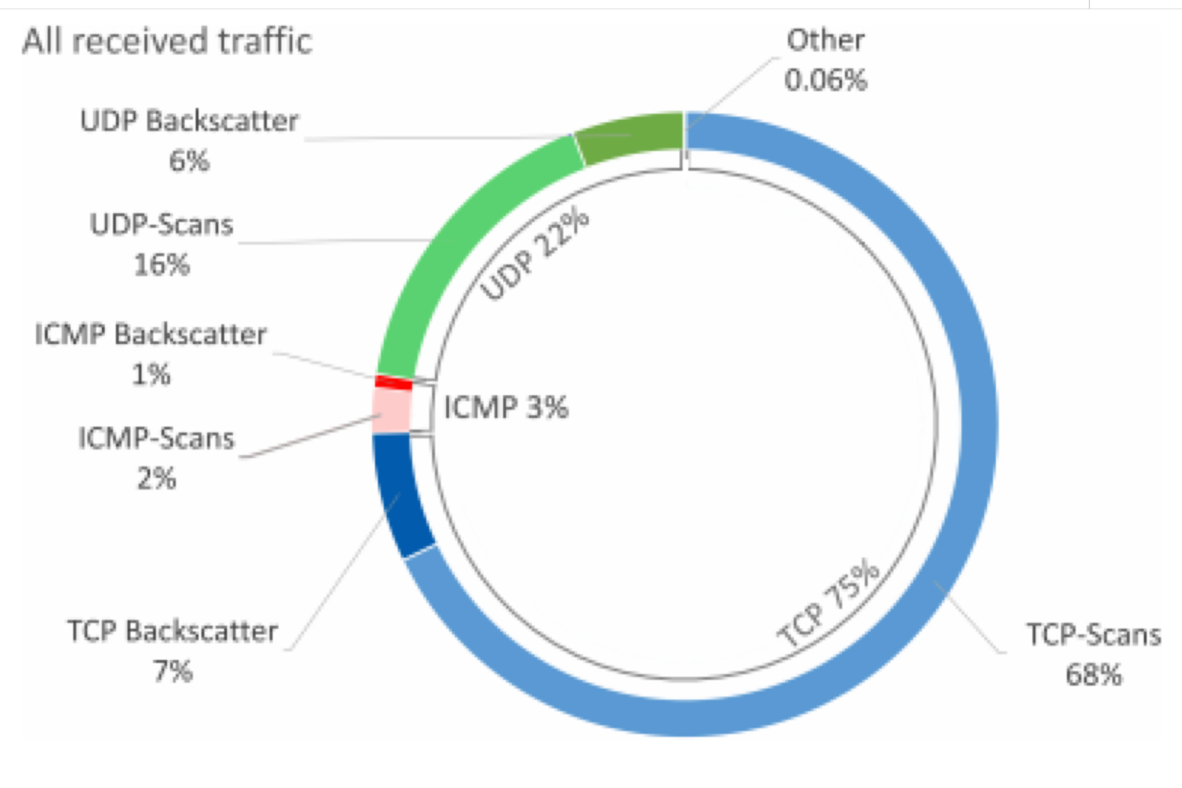
- Darknet = unused routable space

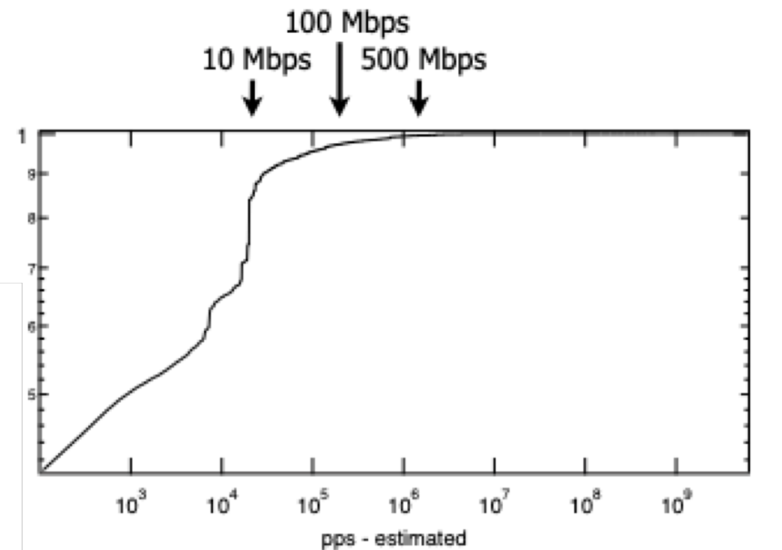


2001: 400 SYN attacks per week

2008: 4425 SYN attacks per 24 hours

Received traffic to idle machine (2017)





ıre 6: Cumulative density function of attack sizes.

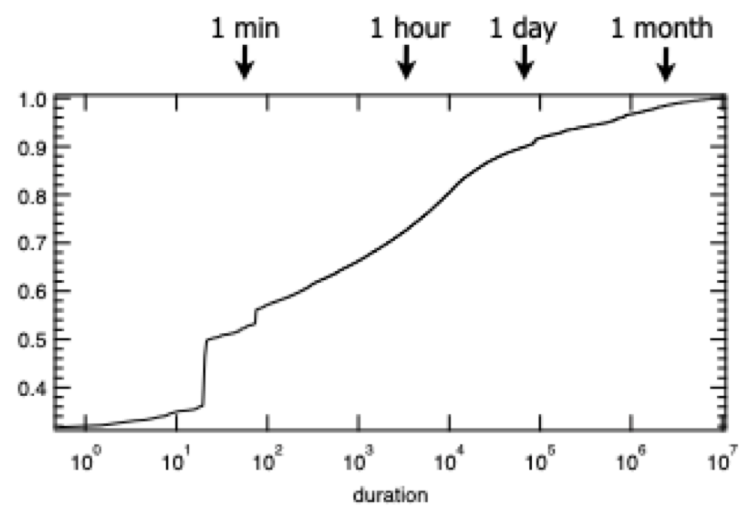
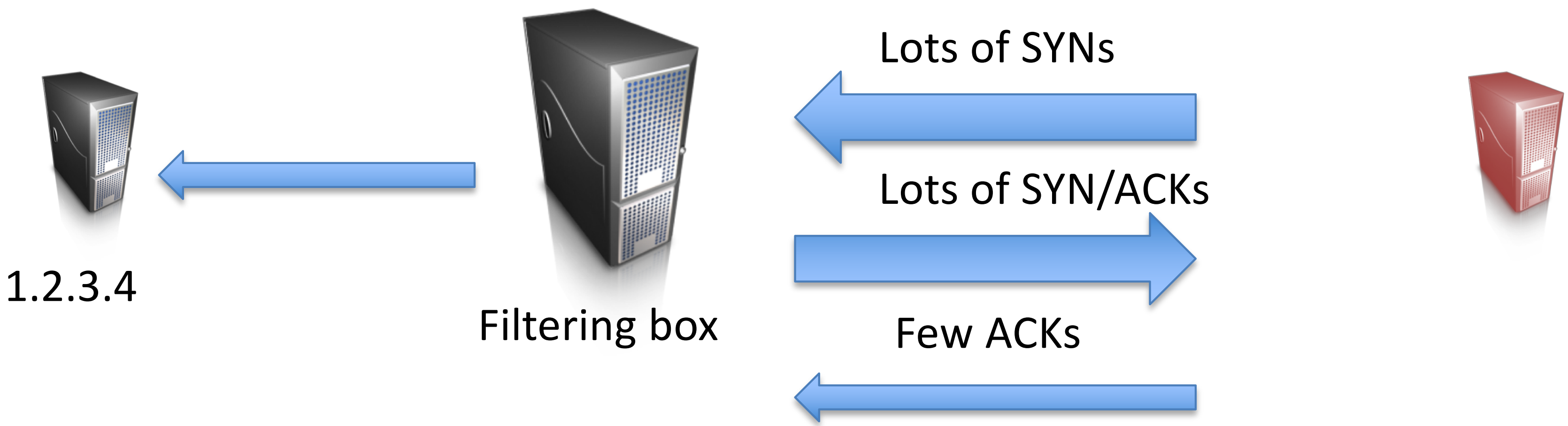


Figure 7: Cumulative density function of attack duration.

Preventing DoS: Akamai approach



Just need a beefy box to help with filtering.
Companies pay Prolexic to do it for them