$40 Hardware Is Enough To Hack $28,000 Police Drones From 2km Away (theregister.co.uk)

94

Posted by **BeauHD** on Saturday April 02, 2016 @01:31AM from the sneak-attack dept.

mask.of.sanity writes:

> Thieves can hijack $28,000 professional drones used widely across the law enforcement, emergency, and private sectors using $40 worth of hardware. The quadcopters can be hijacked from up to two kilometers away thanks to a lack of encryption, which is not present due to latency overheads.

Attackers can commandeer radio links to the drones from up to two kilometers away, and block operators from reconnecting to the craft. With the targeted Xbee chip being very common in drones, IBM security guy Nils Rodday says it is likely many more aircraft are open to compromise.

network security

cs642

computer security

adam everspaugh
ace@cs.wisc.edu

today

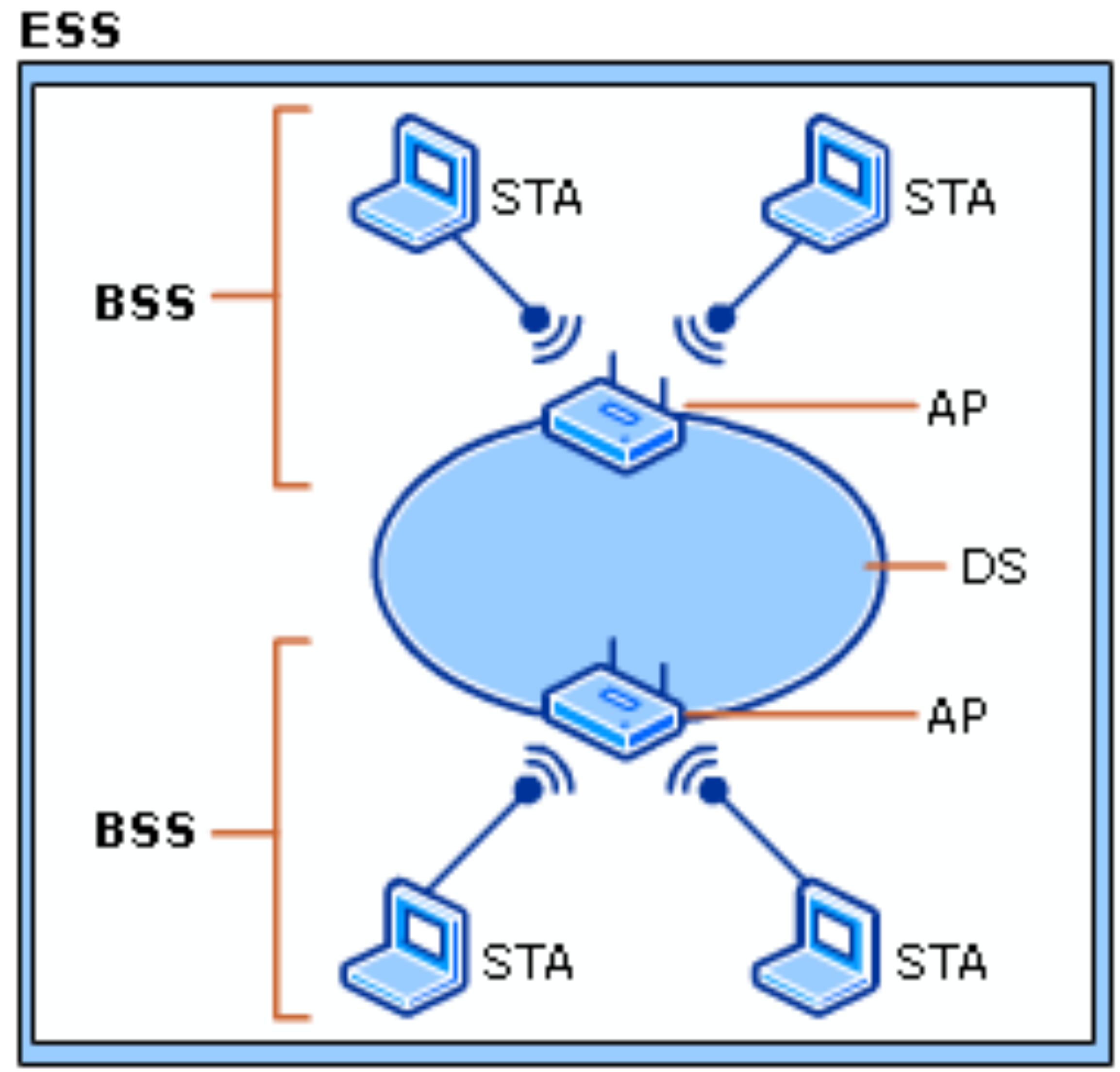
* **Announcement:** HW3 to be released
* WiFi
* IP, TCP
* DoS, DDoS, prevention

802.11 (wifi)

STA = station
AP = access point

BSS = basic service set
DS = distribution service
ESS = extended service set

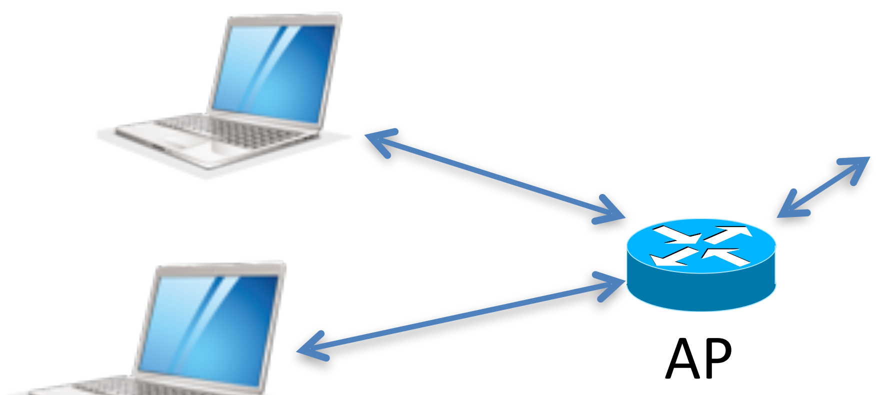
SSID (service set identifier) identifies the 802.11 network



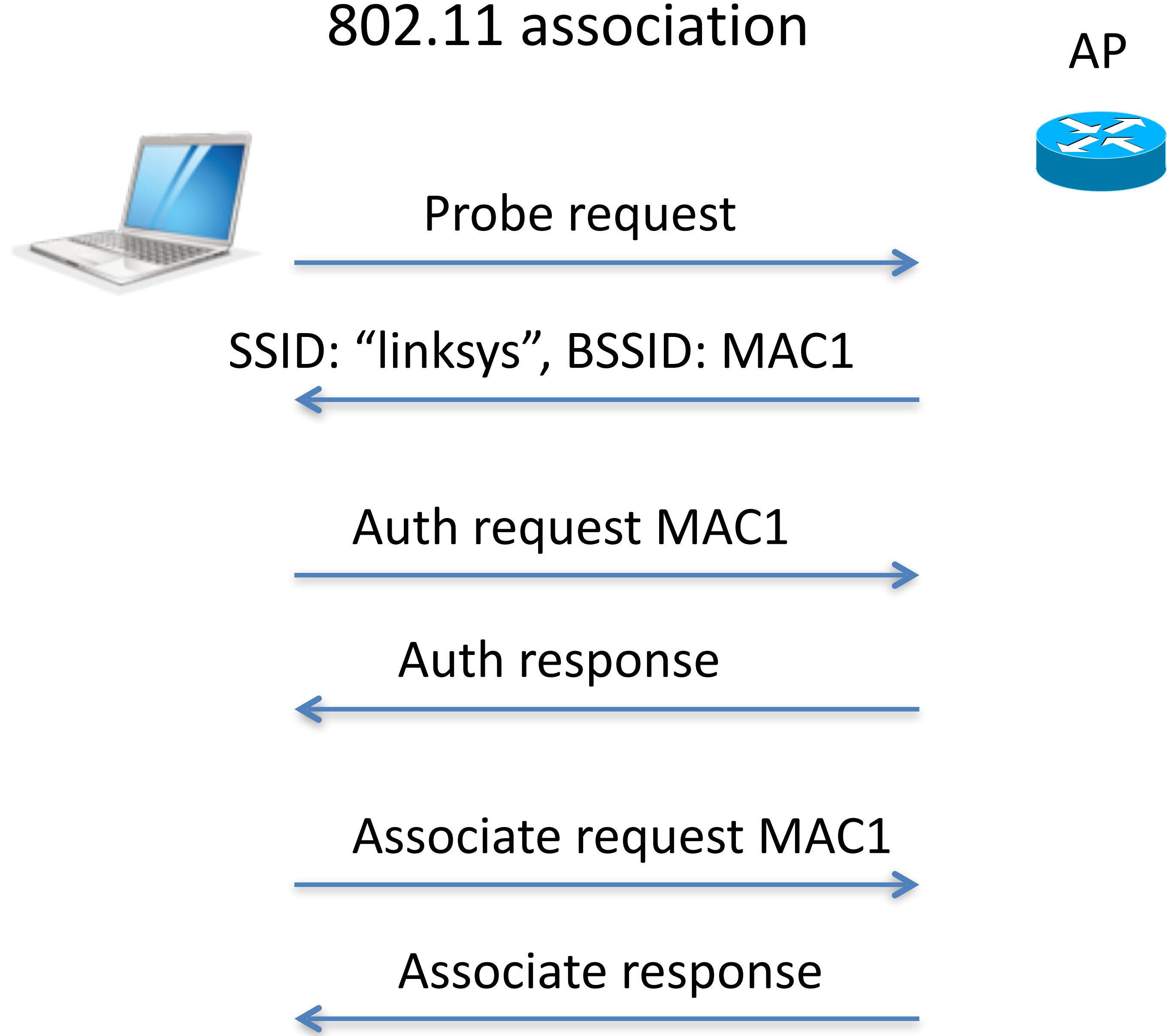
http://technet.microsoft.com/en-us/library/cc757419(WS.10).aspx

Typical WiFi modes:
Unsecured
Wireless Protected Access (WPA2) - password authenticated, encrypted

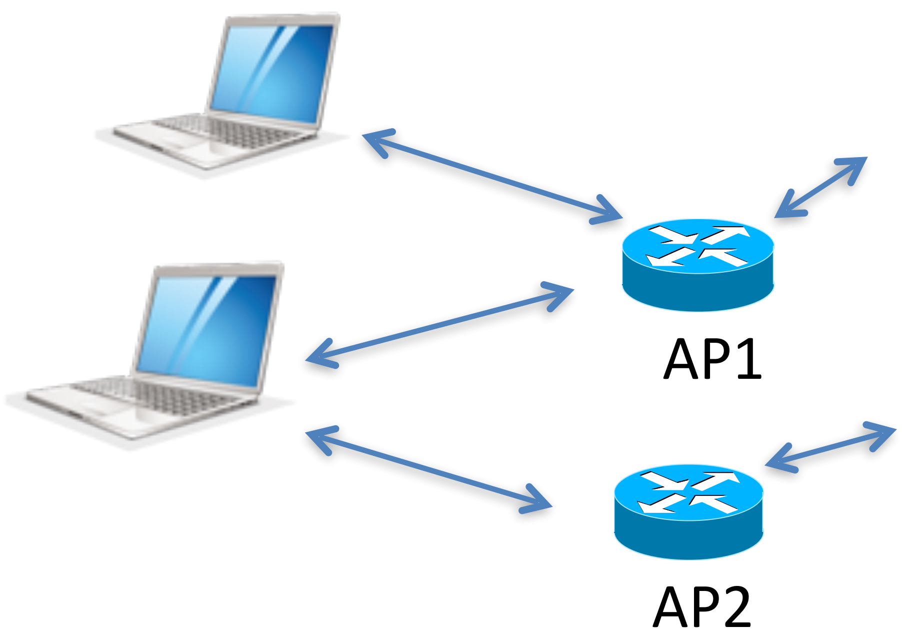
802.11 association



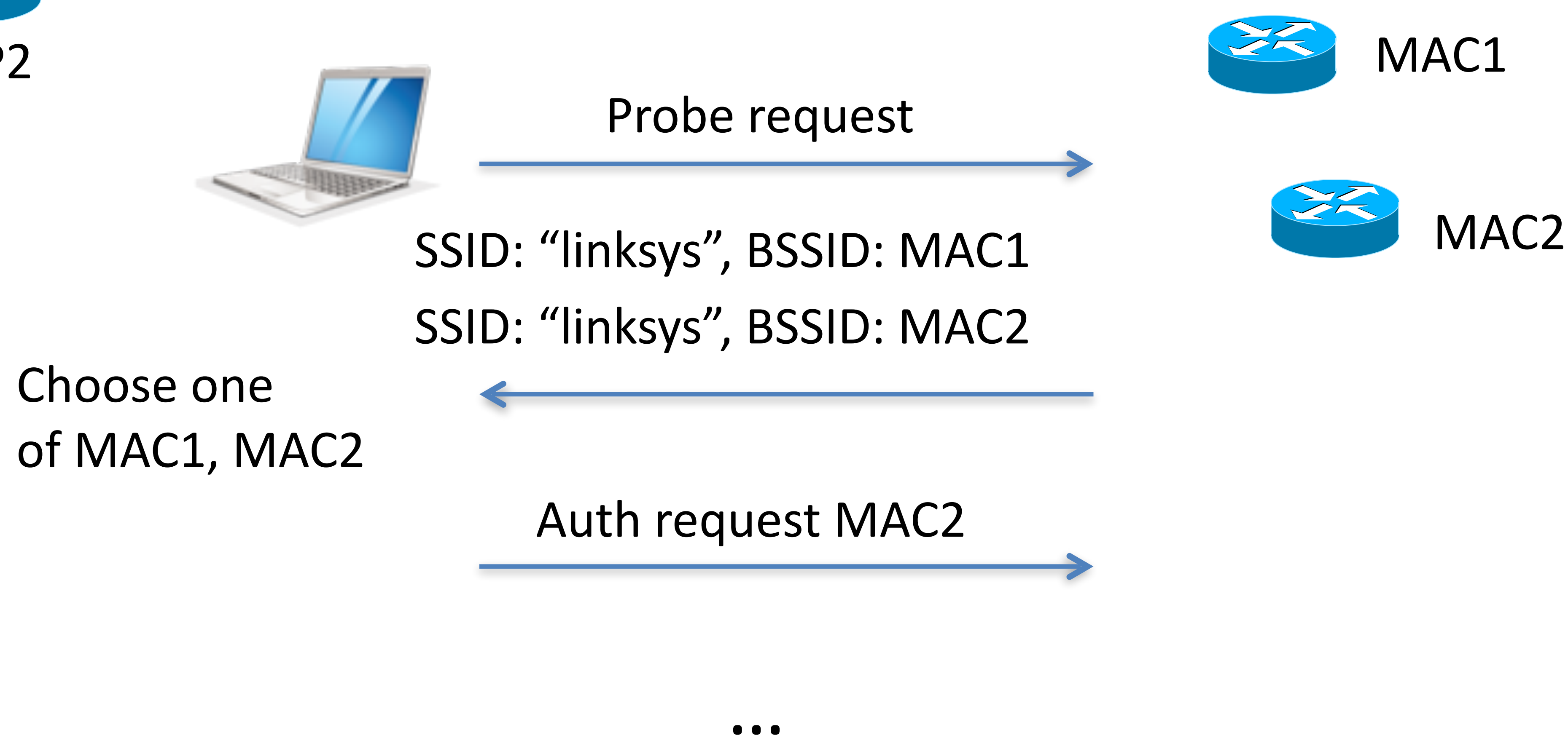
802.11 association



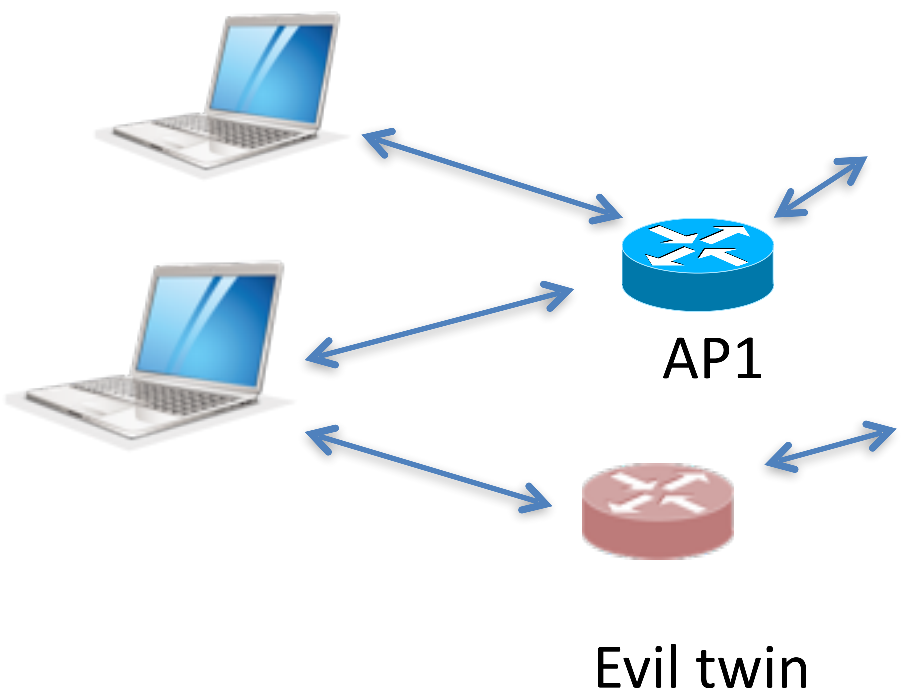
802.11 association



Two APs for same network



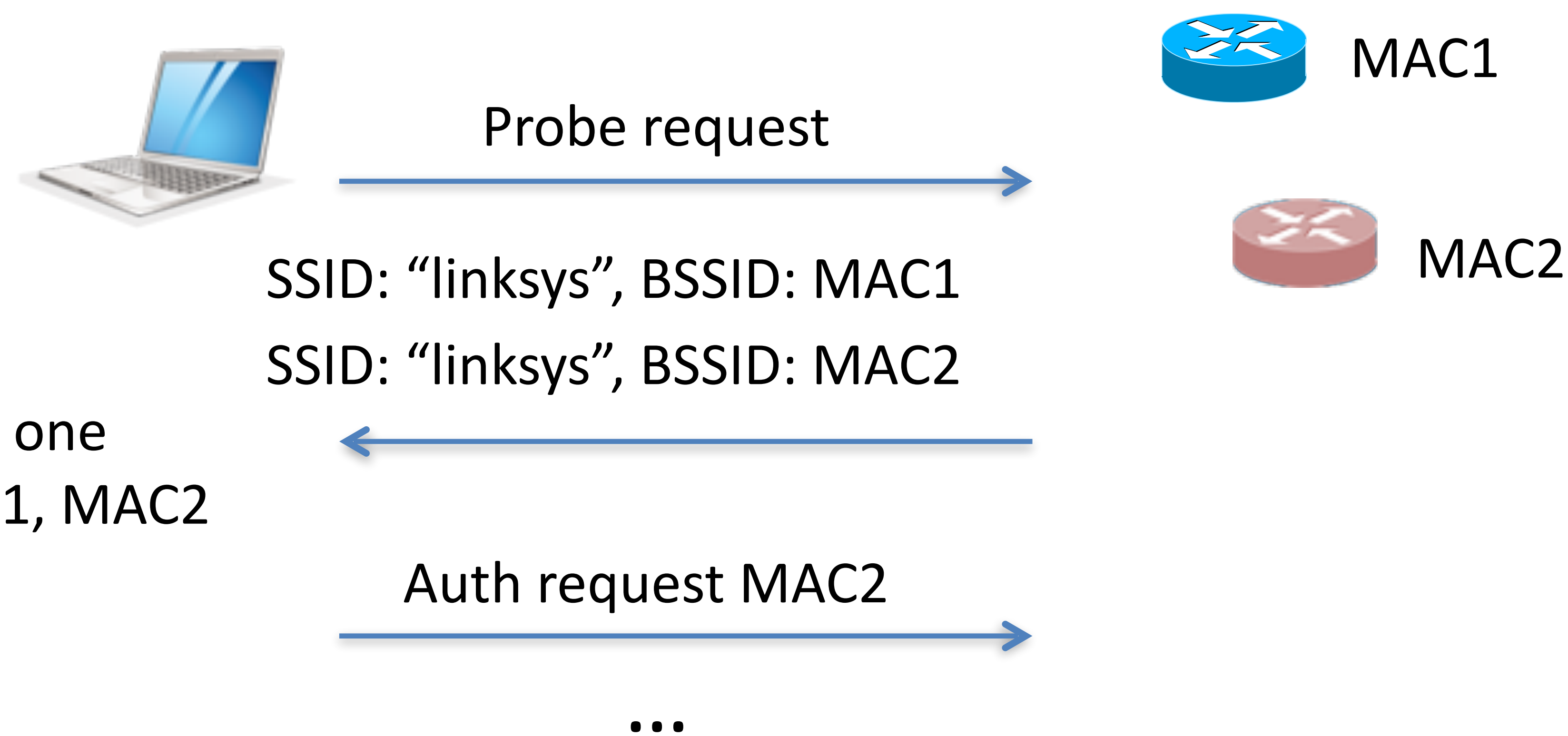
802.11 evil twins



Basic idea:

- Attacker pretends to be an AP to intercept traffic or collect data

Basic attack: rogue AP



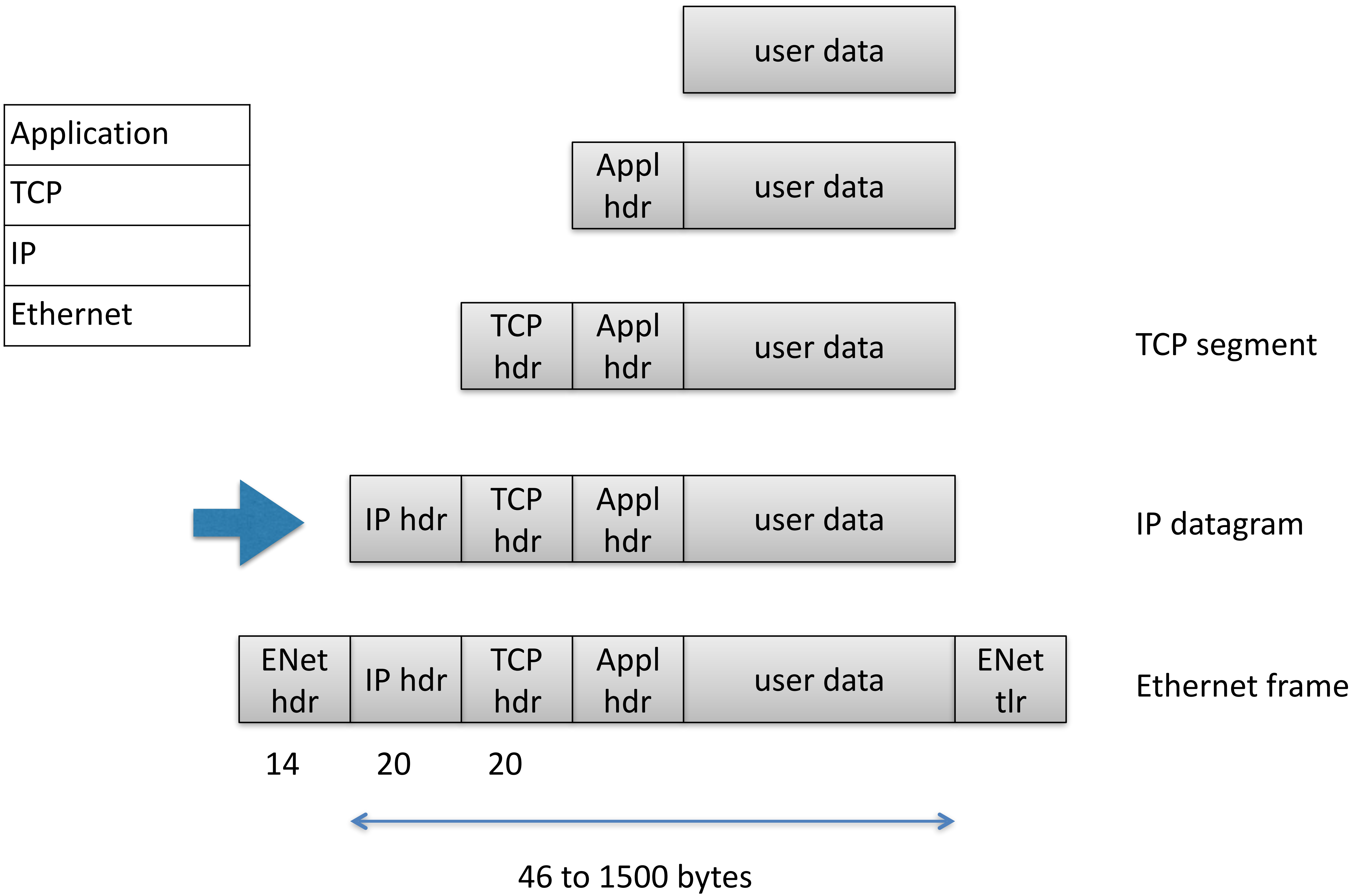
What if client choose MAC1?

Attacker may try to send a forged reset message and force re-connect

Parrot ARdrone

Drone is a WiFi access point
Uses unsecured 802.11 connection (WiFi)
Controlled from iPad or iPhone with an app
Uses MAC address for security

Internet protocol stack



IP protocol (IPv4)

- Connectionless
  - no state
- Unreliable
  - no guarantees
- ICMP (Internet Control Message Protocol)
  - often used by tools such as ping, traceroute

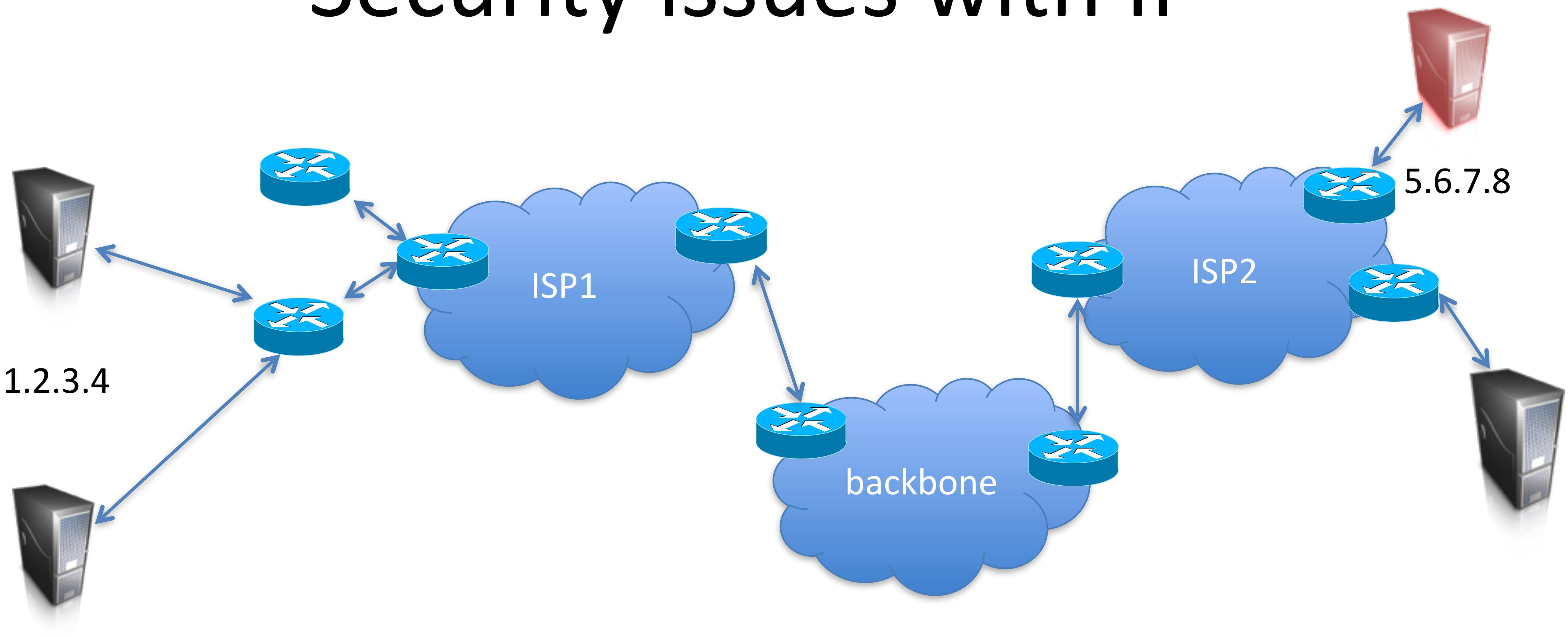
IPv4

ENet hdr	IP hdr	data	ENet tlr

Ethernet frame containing IP datagram

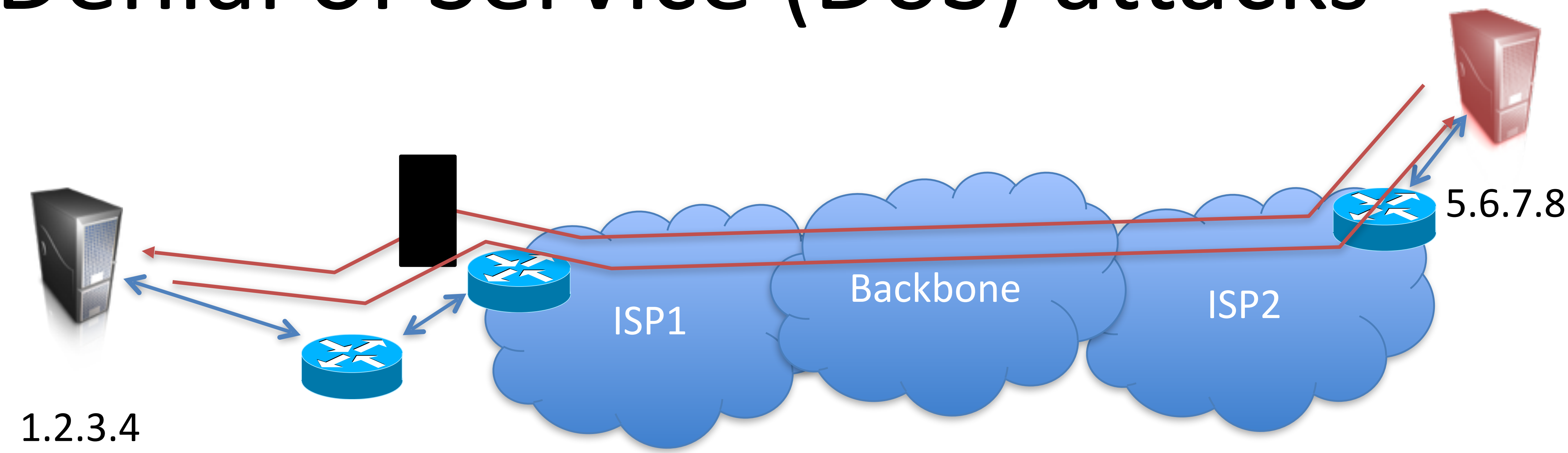
<table>
<tr><td colspan="4">4-bit
version</td><td colspan="4">4-bit hdr
len</td><td colspan="8">8-bit
type of service</td><td colspan="16">16-bit
total length (in bytes)</td></tr>
<tr><td colspan="16">16-bit
identification</td><td colspan="3">3-bit
flags</td><td colspan="13">13-bit
fragmentation offset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

Security issues with IP



Routing has issues, we'll get to that later
What else?
- No source address authentication in general

Denial of Service (DoS) attacks



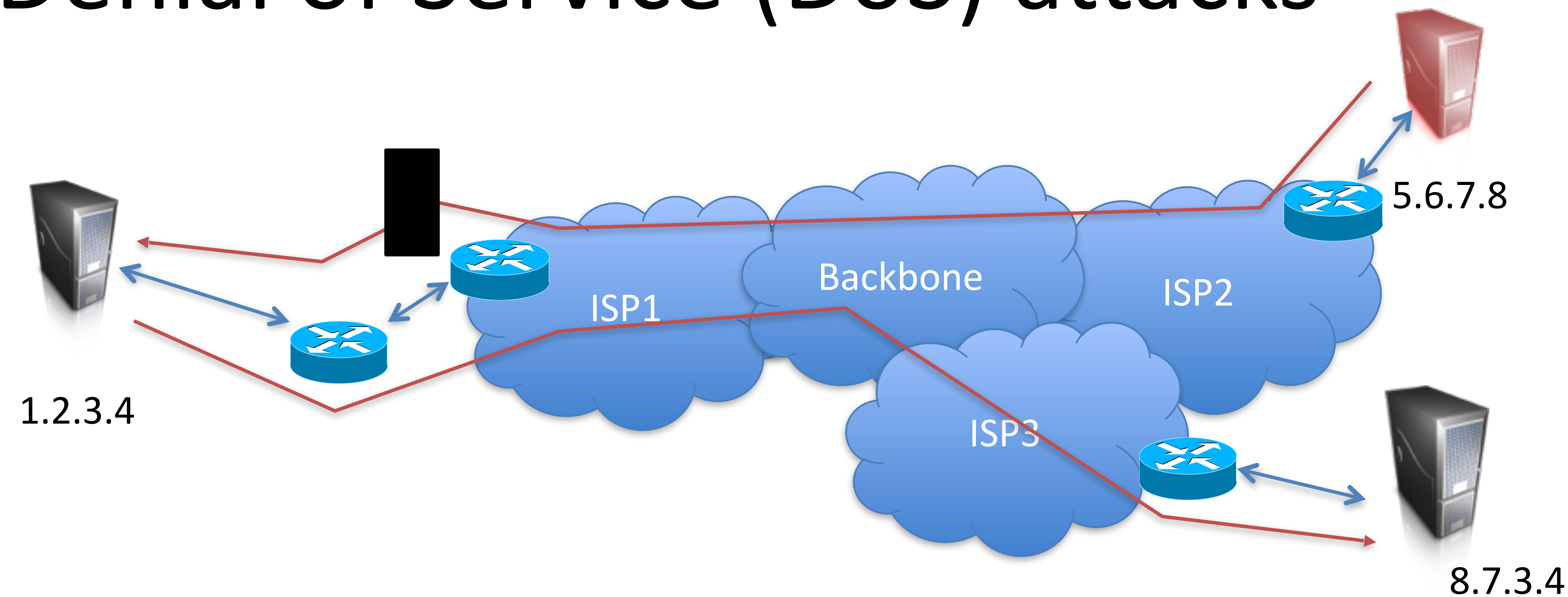
Goal is to prevent legitimate users from accessing victim (1.2.3.4)

think-*pair*-share

ICMP ping flood

- Attacker sends ICMP pings as fast as possible to victim
- When will this work as a DoS? Attacker resources > victim's
- How can this be prevented? Ingress filtering near victim

Denial of Service (DoS) attacks



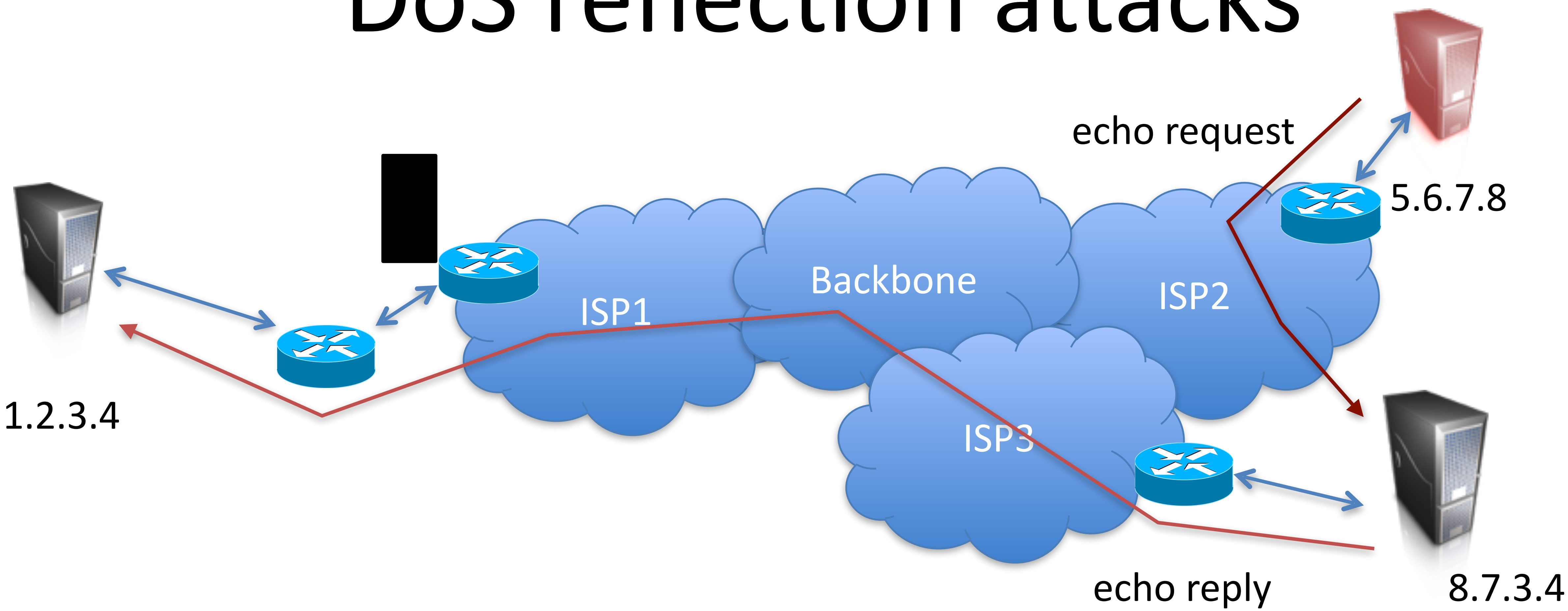
How can attacker avoid ingress filtering?

Attacker can send packet with fake source IP (*packet spoofing*)
Packet will get routed correctly
Replies will not

Send IP packet with source: 8.7.3.4 dest: 1.2.3.4 from 5.6.7.8

Filter based on source may be incorrect

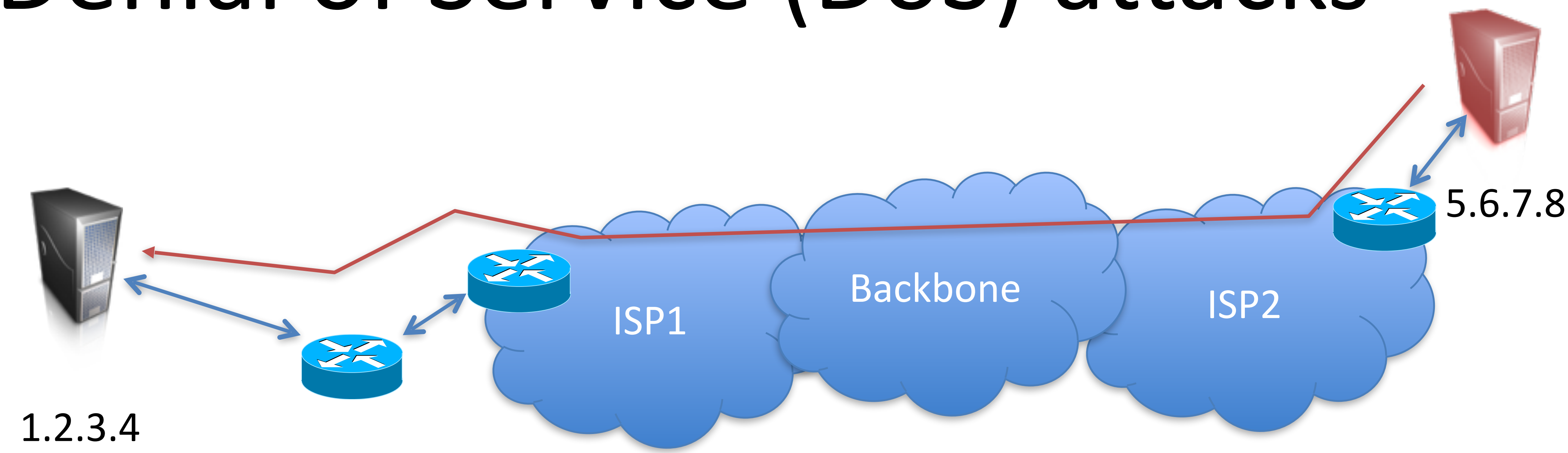
DoS reflection attacks



Note: echo request, DEST IP=8.7.3.4, SRC IP=1.2.3.4

- Attacker can bounce an attack against 1.2.3.4 off 8.7.3.4
- Avoid source filtering

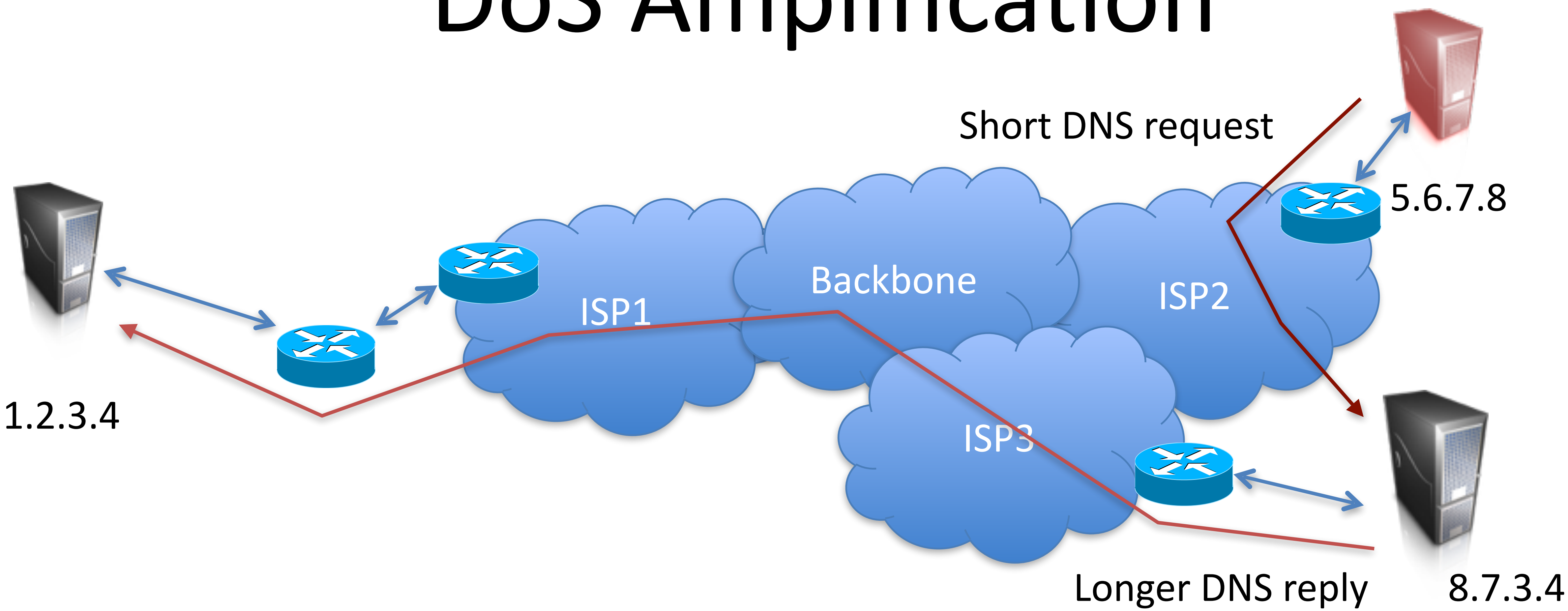
Denial of Service (DoS) attacks



DoS works best when there is asymmetry between victim and attacker

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

DoS Amplification



DoS works best when there is asymmetry between victim and attacker

Example: DNS reflection attacks

Send DNS request with spoofed source IP (~65 byte request)

DNS replies sent to target (~512 byte response)

Reflect + amplify the attack

Estonia attack

Distributed DoS (DDoS)

- April 2007
- Used army of bots
- Attacks continued for weeks with varying intensities
- Targeted government, banks, news, university web sites

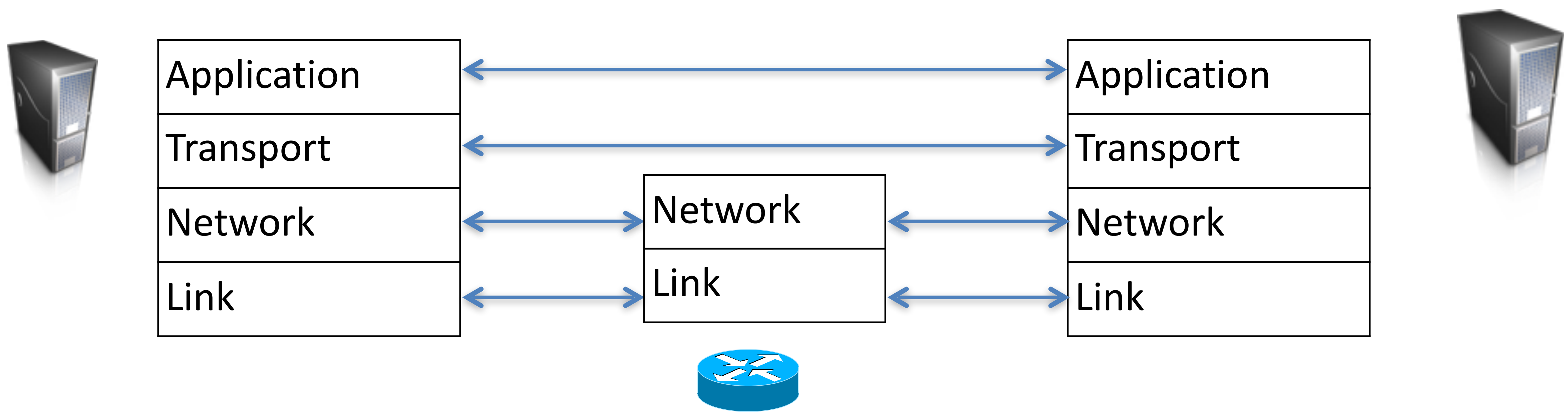
[ATLAS 2007]

From analysis of 2 weeks of attack traffic

- 120+ distinct attacks
- 115 ICMP floods, 4 TCP SYN floods
- 12 attacks: 70-95 Mbps for 10+ hrs
- All attack traffic from outside Estonia
- Solution: Block all foreign traffic until attacks subsided

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
- Goal: reliable, ordered, error-checked delivery of a stream of bytes
  - generates segments
  - timeout segments that aren't acknowledged
  - reorders received segments when necessary

TCP (transport control protocol)

IP hdr	TCP hdr	data

16-bit source port number		16-bit destination port number	
32-bit sequence number			
32-bit acknowledgement number			
4-bit hdr len	6-bits reserved	6-bits flags	16-bit window size
16-bit TCP checksum		16-bit urgent pointer	
options (optional)			
data (optional)			

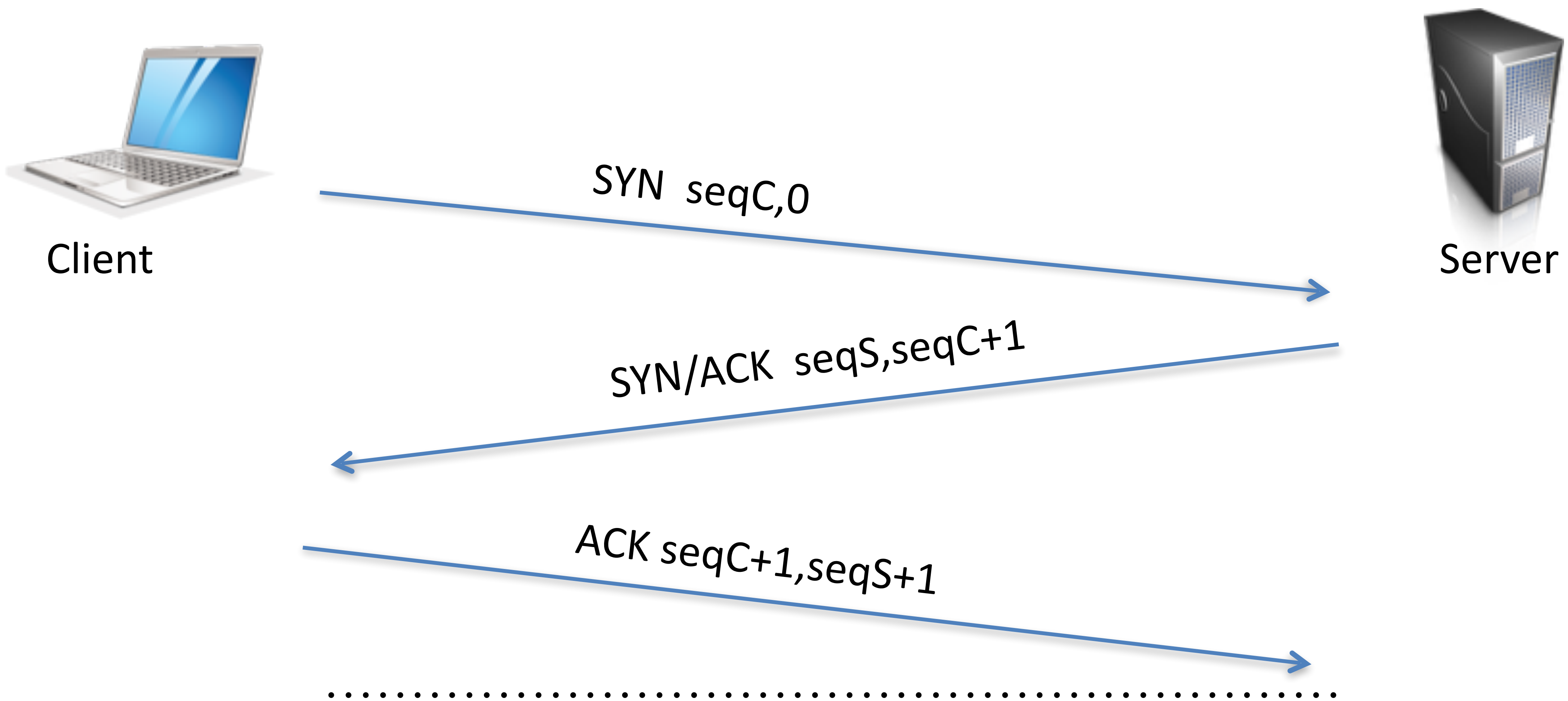
TCP (transport control protocol)

IP hdr	TCP hdr	data

TCP flags

URG	urgent pointer valid
ACK	**acknowledgement number valid**
PSH	pass data to app ASAP
RST	reset connection
SYN	**synchronize sequence #'s**
FIN	finished sending data

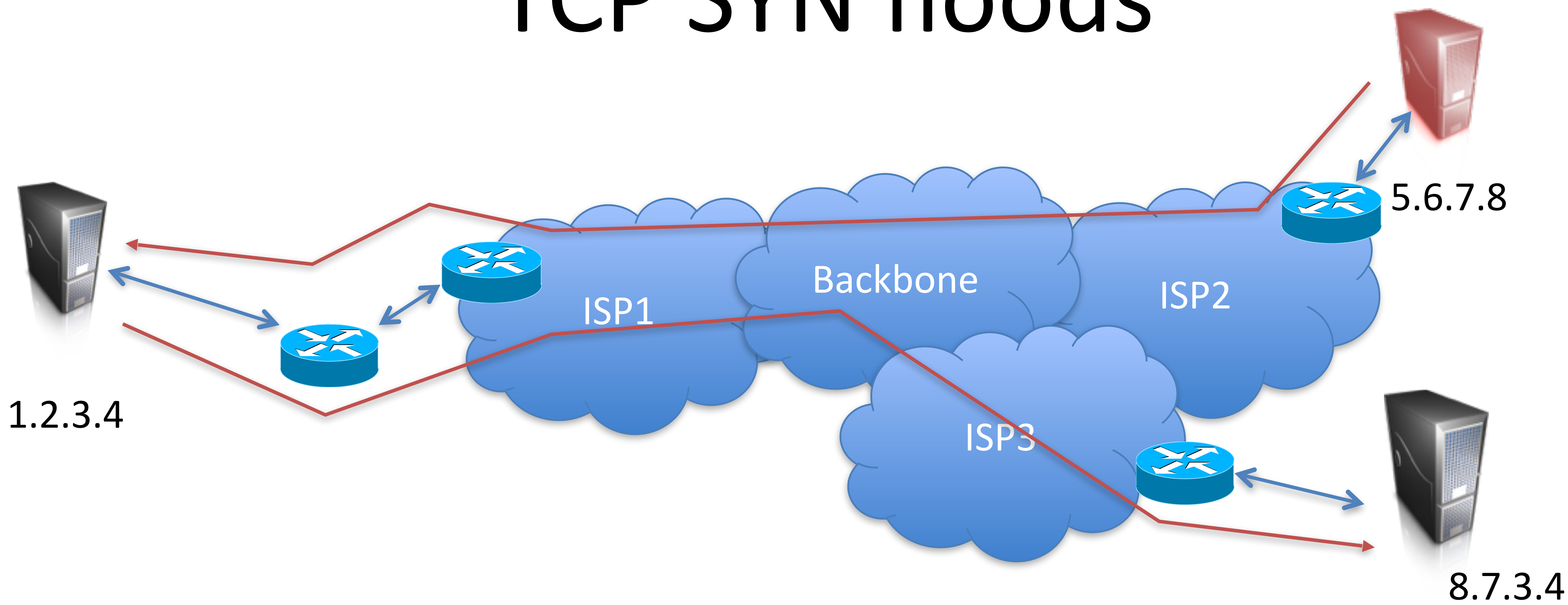
TCP handshake



TCP connection established

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

TCP SYN floods

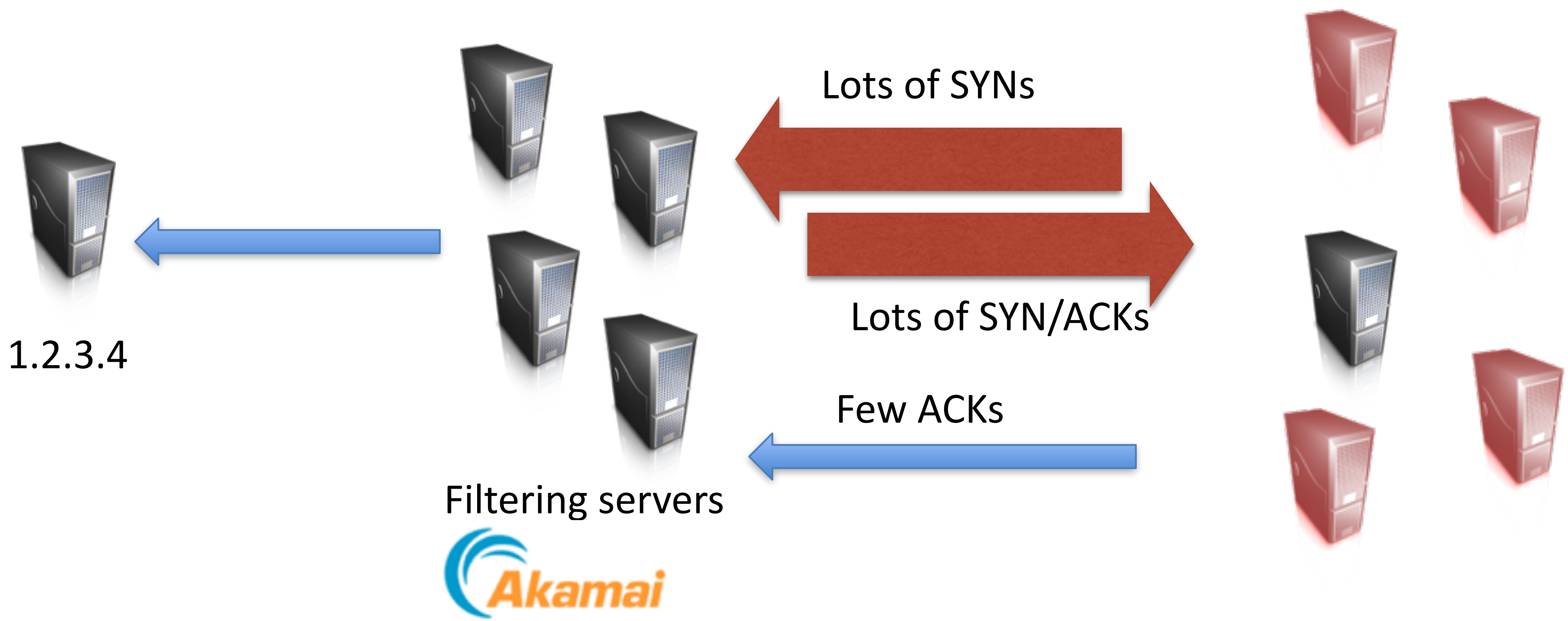


Send lots of TCP SYN packets to 1.2.3.4, no ACK

- 1.2.3.4 maintains state for each SYN packet for some time window

- What asymmetry is being abused?
- What SRC IP does attacker use?
- If attackers sets SRC IP=8.7.3.4, what does 8.7.3.4 receive?

Preventing DDoS



Large number of front-end servers absorb traffic
Forward legitimate-looking traffic to back-end servers

Companies and web sites pay for this: CloudFlare, Arbor Networks, Akamai, and many others

recap

* WiFi Evil Twins
* DoS
  / ICMP Flood
  / DDoS
  / DNS reflection, amplification
  / TCP SYN Flooding
  / Preventing DDoS
* Exit slips
  / 1 thing you learned
  / 1 thing you didn't understand