Network Security
Introduction to Large Scale Attacks

ET6540

PGS.TS. Trần Quang Vinh
BM. Kỹ thuật Thông tin
Viện Điện tử - Viễn thông
Đại học Bách Khoa Hà Nội
vinhtq@mail.hut.edu.vn

DOS Taxonomy: Quiz One

Match the DOS attack classification with its description.

Attacks:	
4	Random Scanning
2	Permutation Scanning
3	Signpost Scanning
1	Hitlist Scanning

Descriptions:

1. A portion of a list of targets is supplied to a compromised computer.
2. All compromised computers share a common pseudo-random permutation of the IP address space.
3. Uses the communication patterns of the compromised computer to find new target.
4. Each compromised computer probes random addresses.

DOS Taxonomy: Quiz Two

Match the DOS attack classification with its description.

Attacks:

2 Subnet Spoofing

1 Random Spoofing

3 Fixed Spoofing

Descriptions:

1. Generate 32-bit numbers and stamp packets with them.
2. Generate random addresses within a given address space.
3. The spoofed address is the address of the target.

DOS Taxonomy: Quiz Three

Match the DOS attack classification with its description.

Attacks:	Descriptions:
2 Server Application	1. The motivation of this attack is a crucial service of a global internet operation, for example core router
3 Network Access	2. The attack is targeted to a specific application on a server
1 Infrastructure	3. The attack is used to overload or crash the communication mechanism of a network.

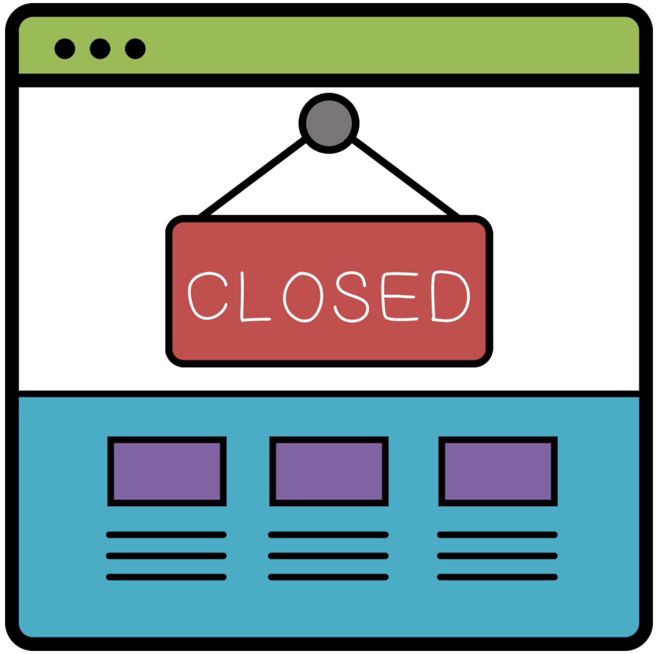
Network DoS:

Goal: take out a large site with little computing

How:

- Amplification
  - Small number of packets

BIG EFFECT



Network DoS:

Two types of amplification attacks:

DoS bug:

- Design flaw allowing one Machine to disrupt a service

DoS flood:

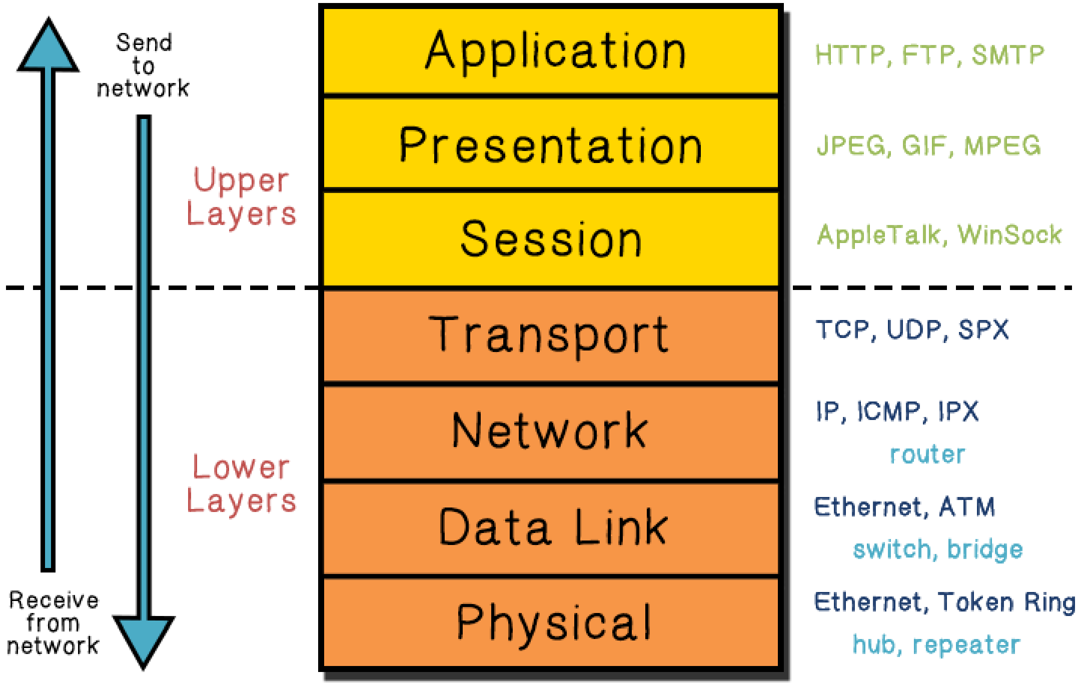
- Command botnet to Generate flood of requests

Network DoS:

DoS can happen at any layer

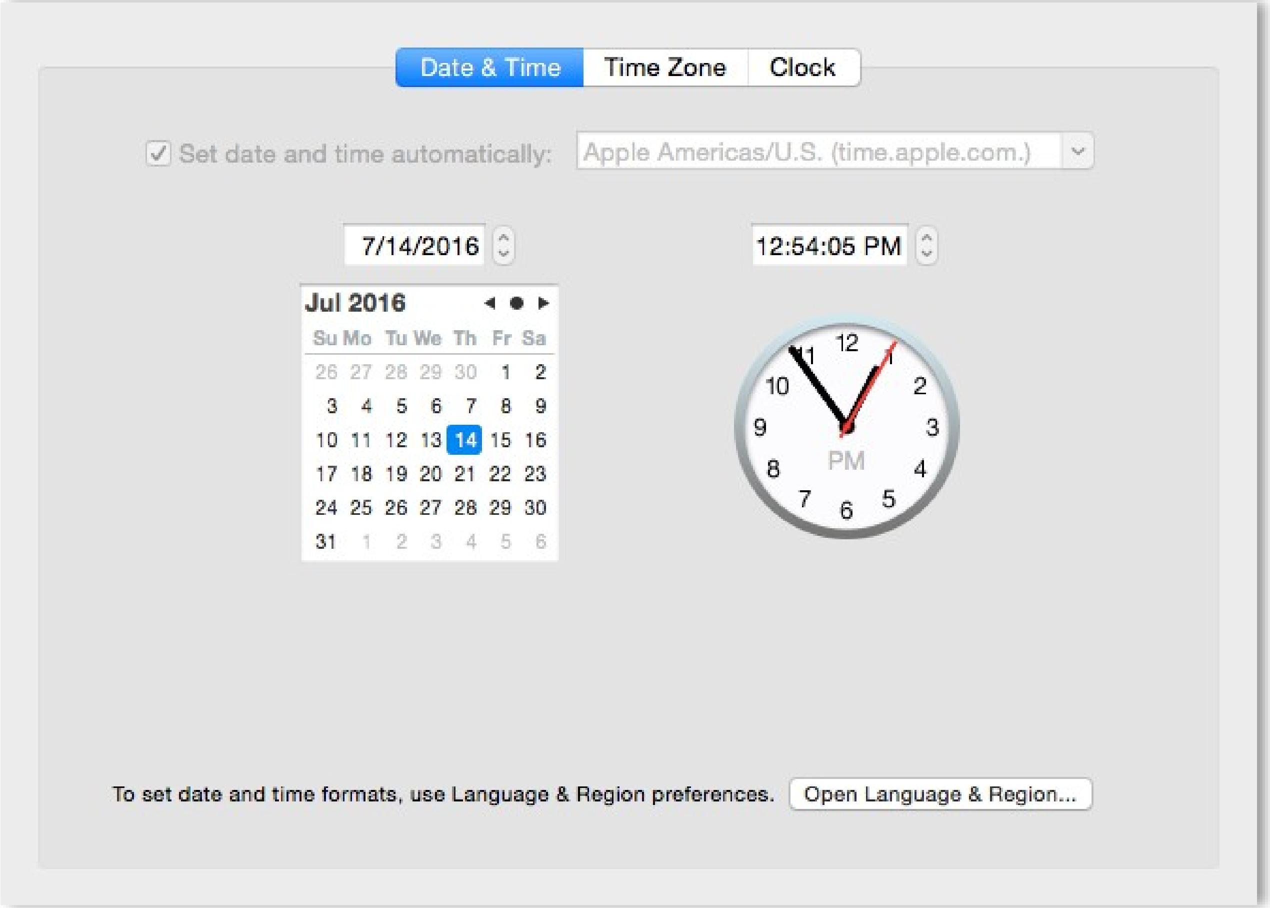
Sample DoS at different layers:

- **Link**
- **TCP/UDP**
- **Application**



Sad truth: Current internet not designed to handle DDoS attacks

Amplification Quiz



NTP – Network Time Protocol

- Used to synchronize machines and their clocks.

Amplification Quiz

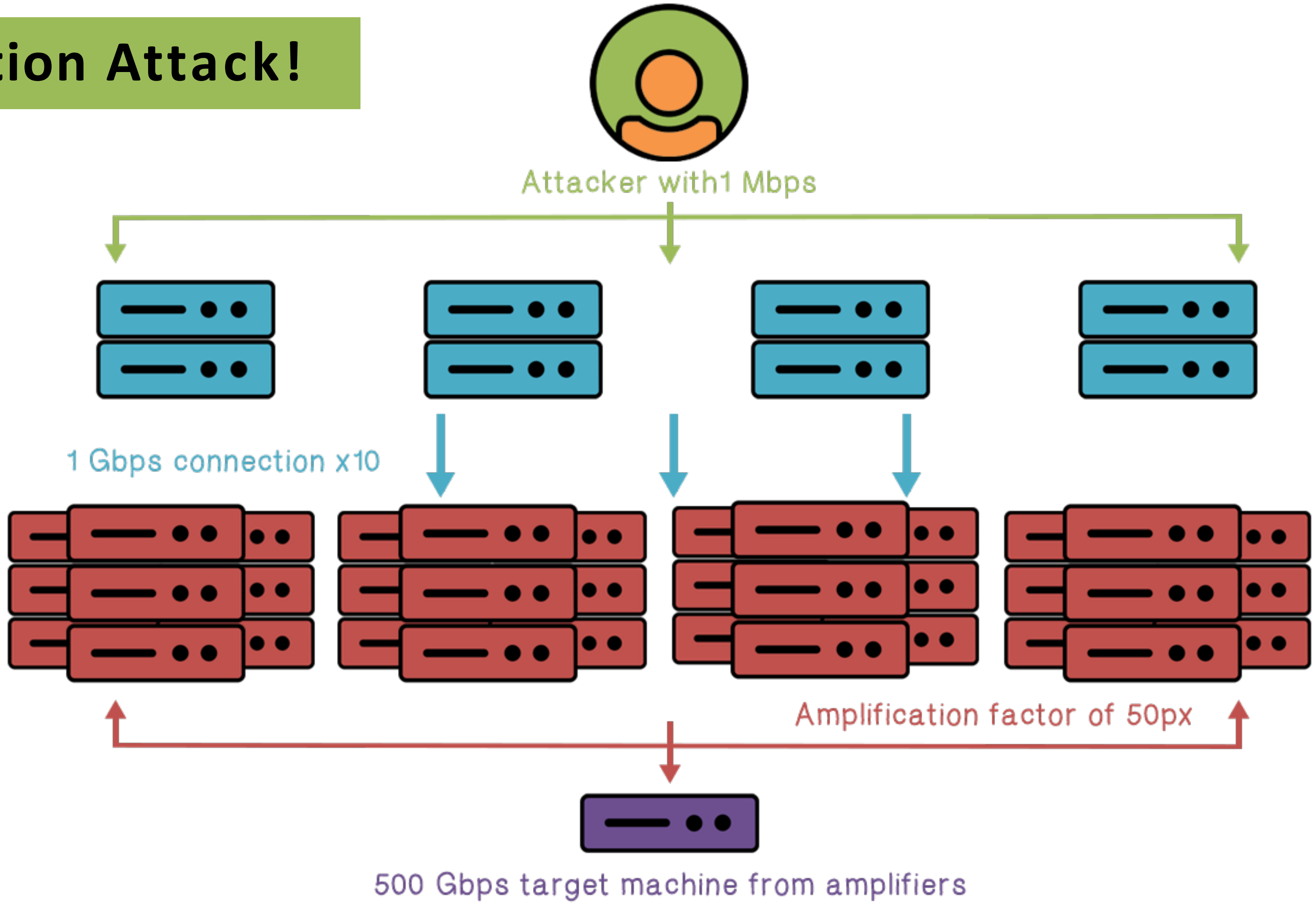
Which of these are reasons why the UDP-based NTP protocol is particularly vulnerable to amplification attacks?

Select all that are true.

- [x] A small command can generate a large response.
- [x] Vulnerable to source IP spoofing.
- [x] It is difficult to ensure computers communicate only with legitimate NTP severs.

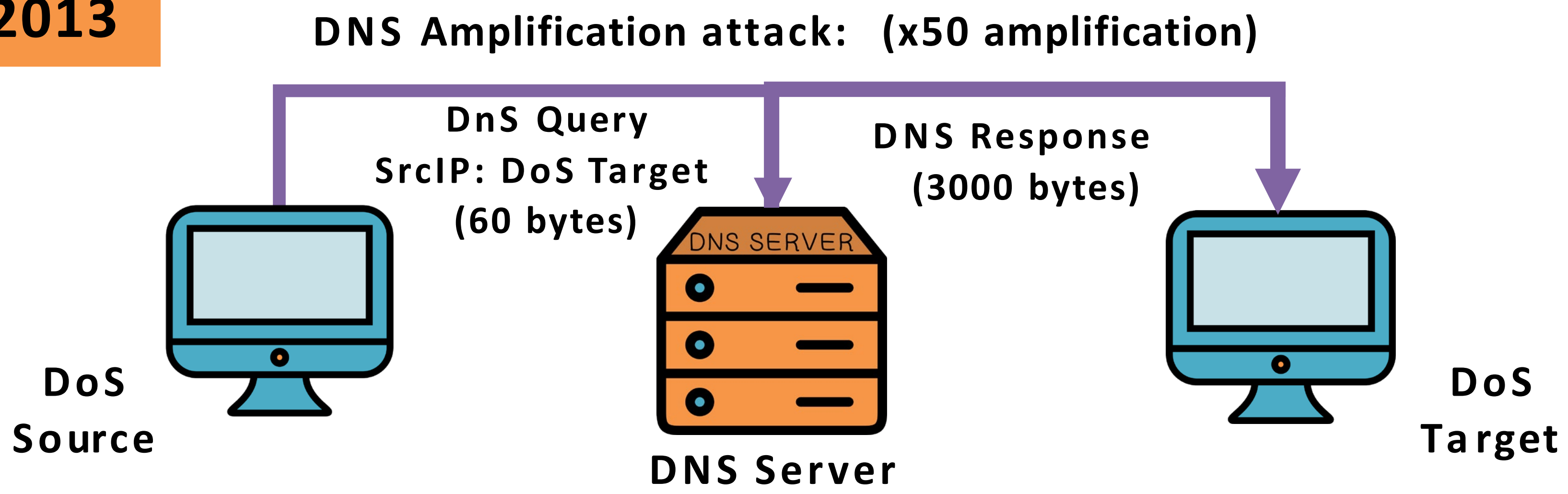
Amplification Example

Amplification Attack!



Amplification Example

March 2013



2006: 0.58M open resolvers on Internet *(Kaminsky-Shiffman)*

2014: 28M open resolvers *(openresolverproject.org)*

- March 2013: DDoS attack generating 309 Gbps for 28 mins

IP Header Format

Connectionless

Unreliable

Best Effort

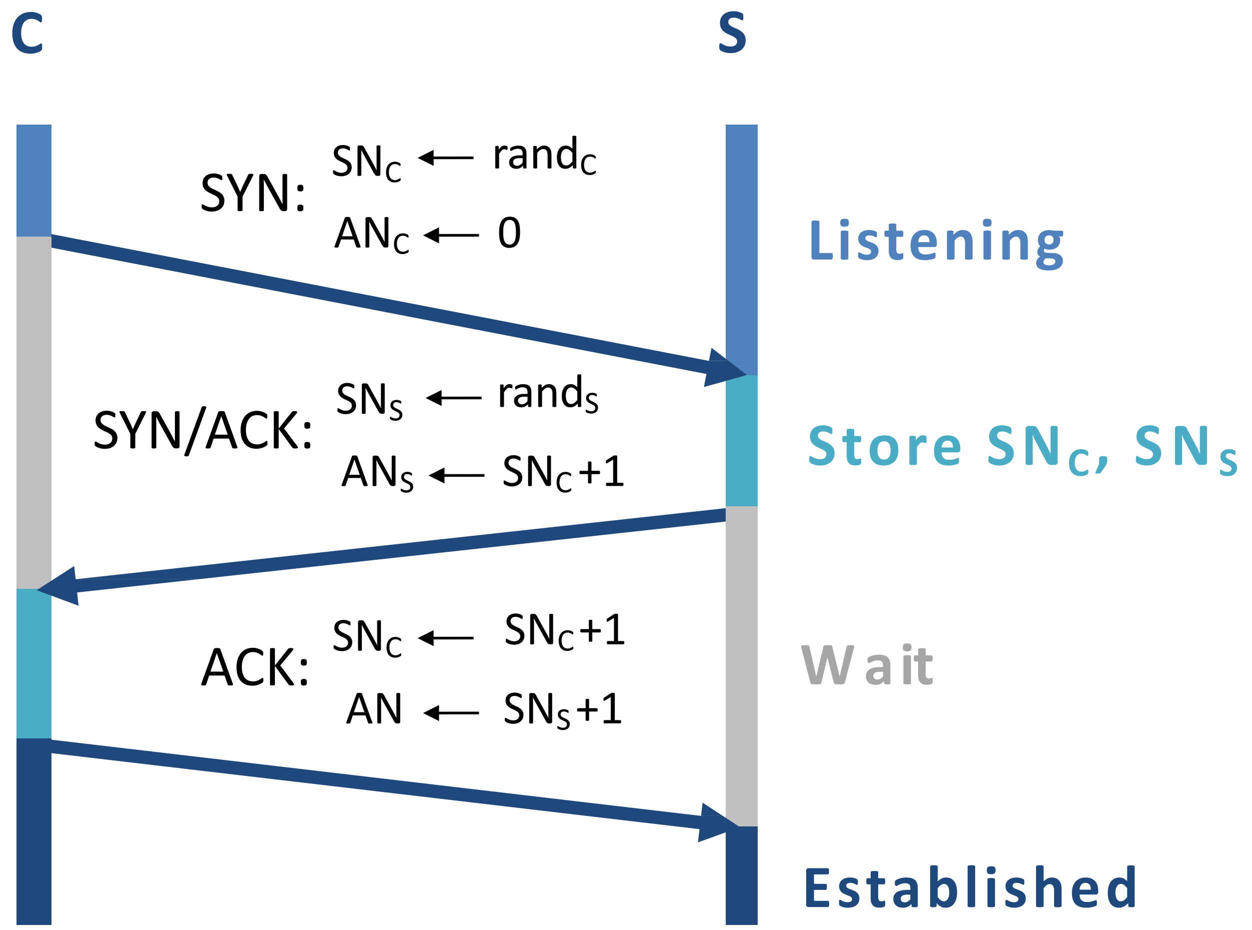
Version	Header Length
Type of Service	
Total Length	
Identification	
Flags	Fragment Offset
Time to Live	
Protocol	
Header Checksum	
Source Address of Originating Host	
Destination Address of Target Host	
Options	
Padding	
IP Data	

TCPHeader Format

- Session Based
- Congestion control
- In order delivery

Source Port						Dest Port
SEQ Number						
ACK Number						
	URG	ACK	PSH	RST	SYN	FIN
Other stuff						

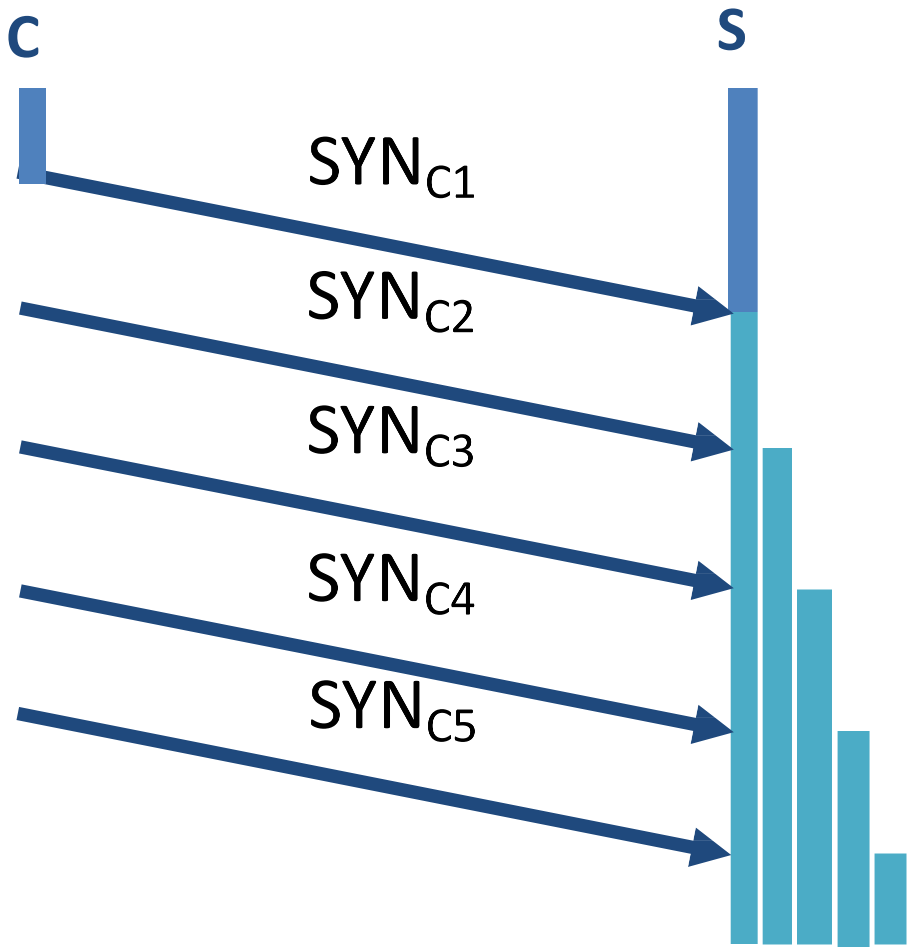
TCP Handhake



TCP SYN Flood I: low rate (DoS Bug)

Single machine:

- SYN Packets with random source IP addresses
- Fills up backlog queue on server
- No further connections possible



TCP SYN Flood I

A classic SYN flood example

MS Blaster worm (2003)

- Infected machines at noon on Aug 16th:
- SYN flood on port 80 to windowsupdate.com
- 50 SYN packets every second
  - each packet is 40 bytes
- Spoofed source IP: a.b.X.Y where X,Y random

MS Solution

- new name: windowsupdate.microsoft.com

TCP SYN Flood I

Low rate SYN flood defenses

Non-solution:

- Increase backlog queue size
- Decrease timeout

Correct Solution:

- Syncookies: remove state from server
- Small performance overhead

SYN COOKIES

Idea: use secret key and data in packet to generate server SN

Server responds to Client with SYN-ACK cookie:

- T = 5 (bit): **counter incremented every 64 secs.**
- L = MAC_{key} (SAddr, SPort, DAddr, DPort, SN_C, T) [24 bits]

key: picked at random during boot

- SN_S = (T . mss . L) (|L| = 24 bits)
- **Server does not save state**

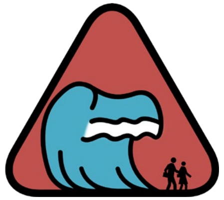
Honest client responds with
ACK (AN=SN_S +1, SN=SN_C+1):

- **Server allocates space for socket only if valid SN_S**

Syn Cookies Quiz

Select all the true statements:

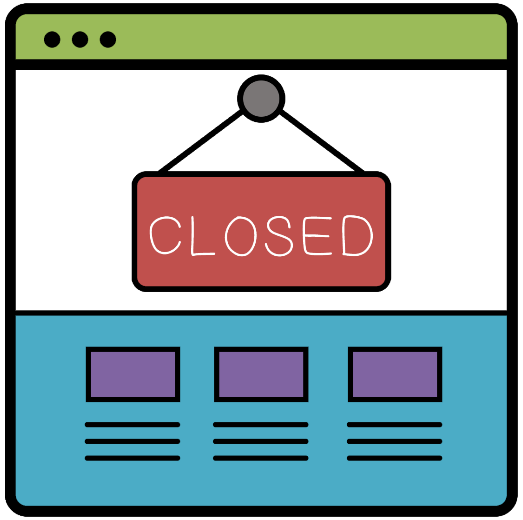
- [ ] SYN cookies require modified versions of TCP
- [ ] SYN cookies lead to overall slower performance
- [x] The server must reject all TCP options because the server discards the SYN queue entry



SYN Floods II: Massive flood

Command bot army to flood specific target: (DDoS)

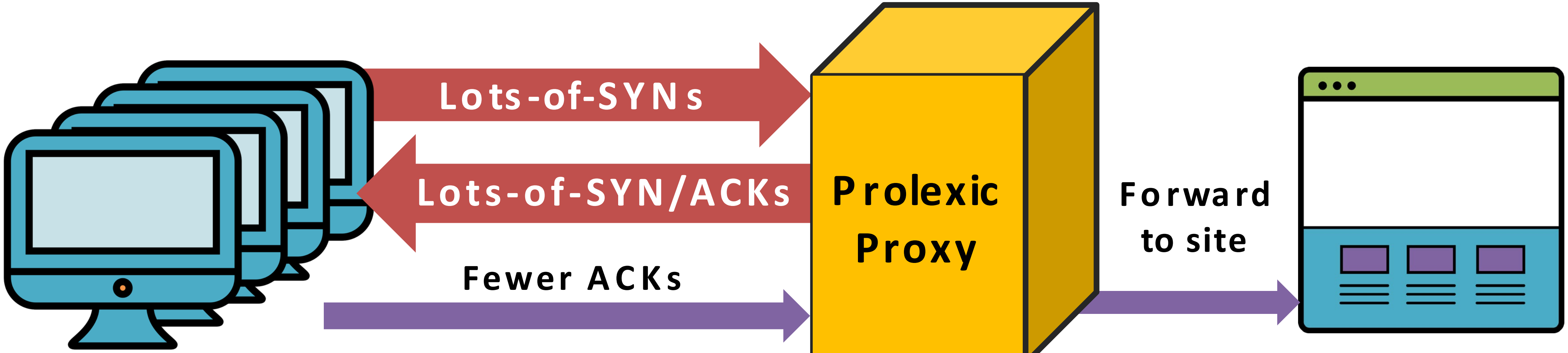
- 20,000 bots can generate 2Gb/sec of SYNs (2003)
- At web site:
  - Saturates network uplink or network router
  - Random source IP → attack SYNs look the same as real SYNs



SYN Floods II: Massive flood

Prolexic / CloudFlare

Idea: only forward established TCP connections to site



Stronger attacks: TCP connection flood

Command bot army:

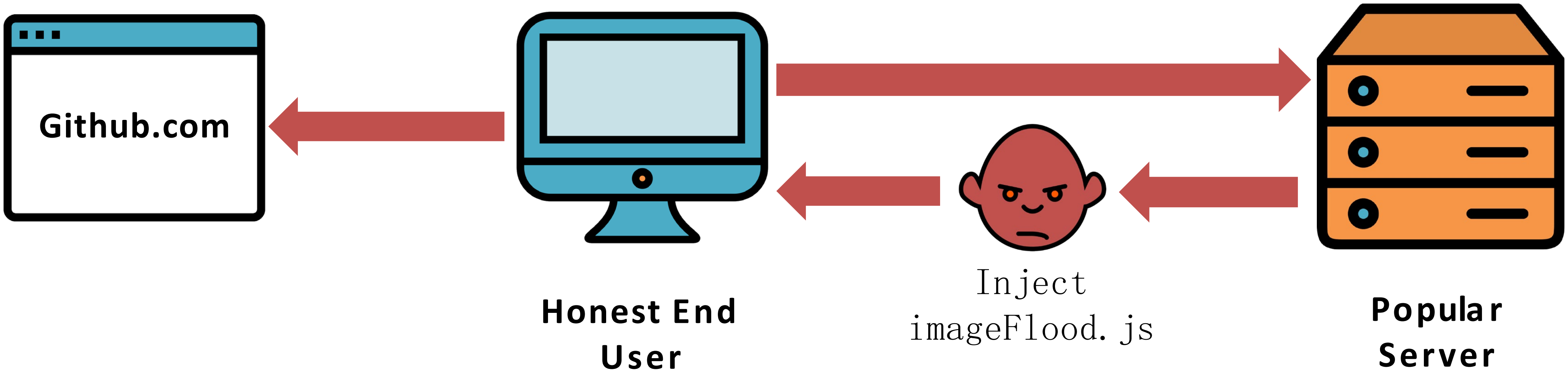
- Complete TCP connection to web site
- Send short HTTP HEAD request
- Repeat

Will bypass SYN flood protection proxy but:

- Attacker can no longer use random source IPs
- Reveals location of bot zombies
- Proxy can now block or rate-limit bots

A real-world example: GitHub(3/2015)

Javascript-based DDoS:



A real-world example: GitHub(3/2015)

imageFlood.js

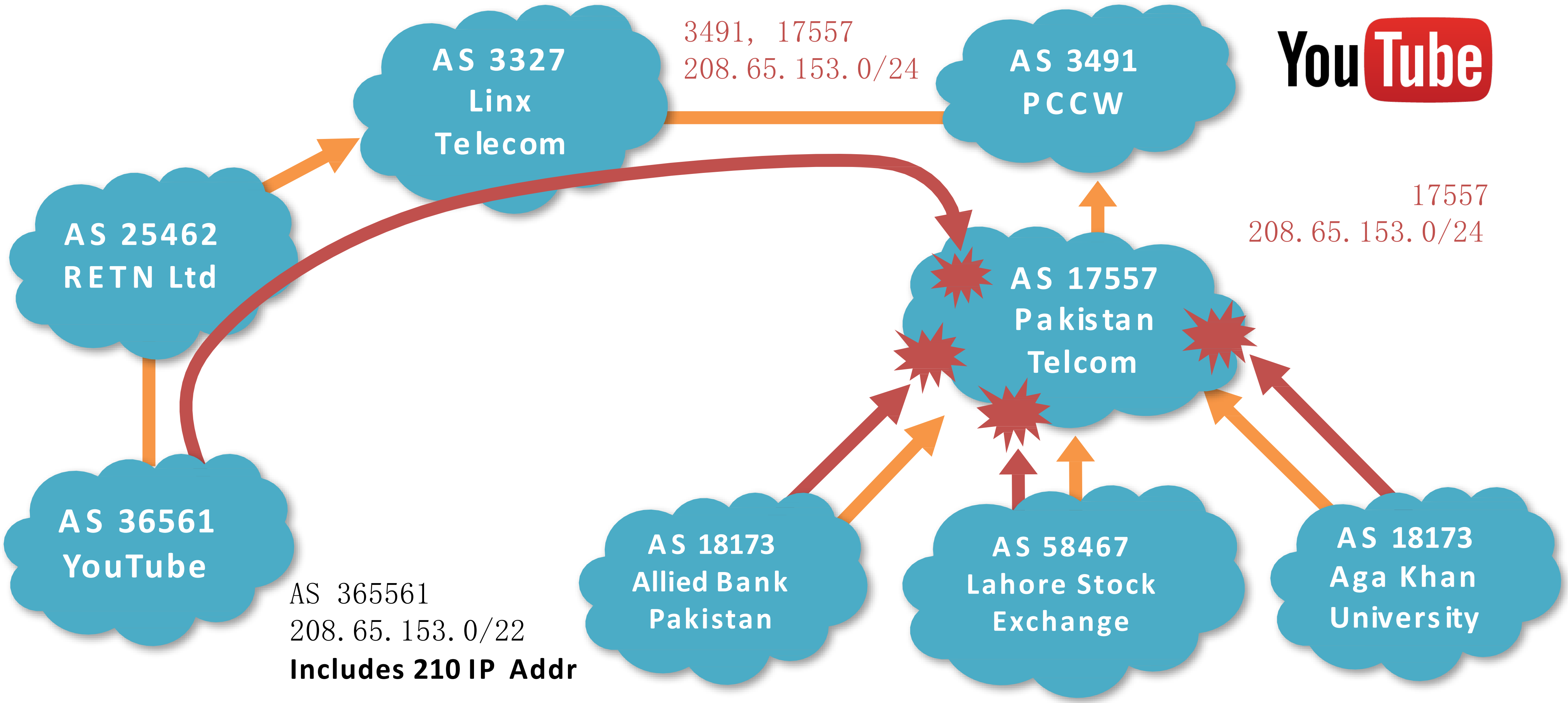
```
Function  imgflood()  {
 var TARGET  =  ‘victim-website.com/index.php?’
 var rand= Math.floor(Math.random()    * 1000)
 var pic = new Image()
Pic.src  =  ‘http://’ +TARGET+rand+’ =val’
}
setInterval(imgflood,10)
```

Flood Attack Quiz

With regards to a UDP flood attack, which of the following statements are true:

- [x] Attackers can spoof the IP address of their UDP packets
- [ ] The attack can be mitigated using firewalls
- [x] Firewalls cannot stop a flood because the firewall is susceptible to flooding.

DoS via route hijacking

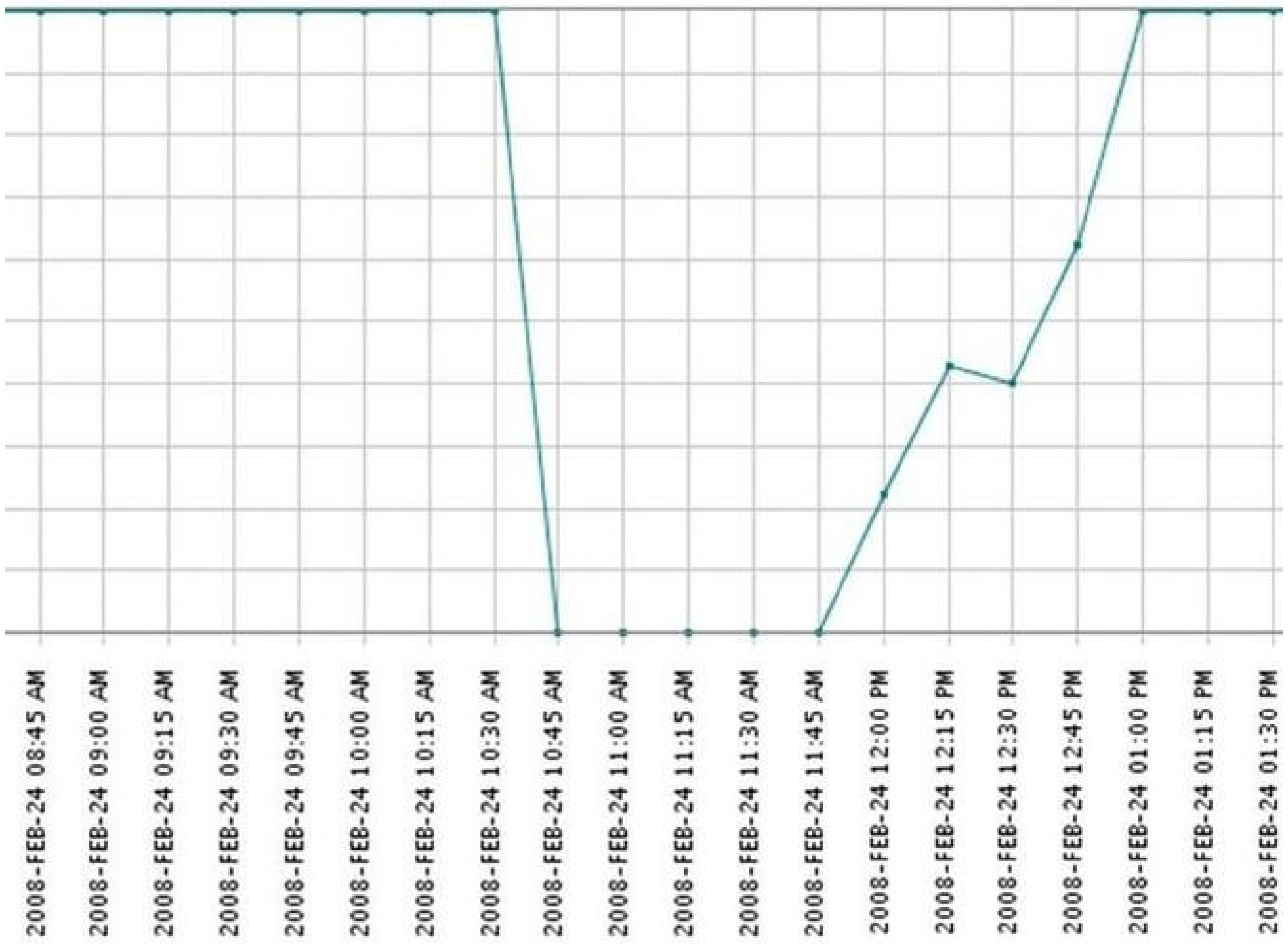


DoS via route hijacking

DETOUR

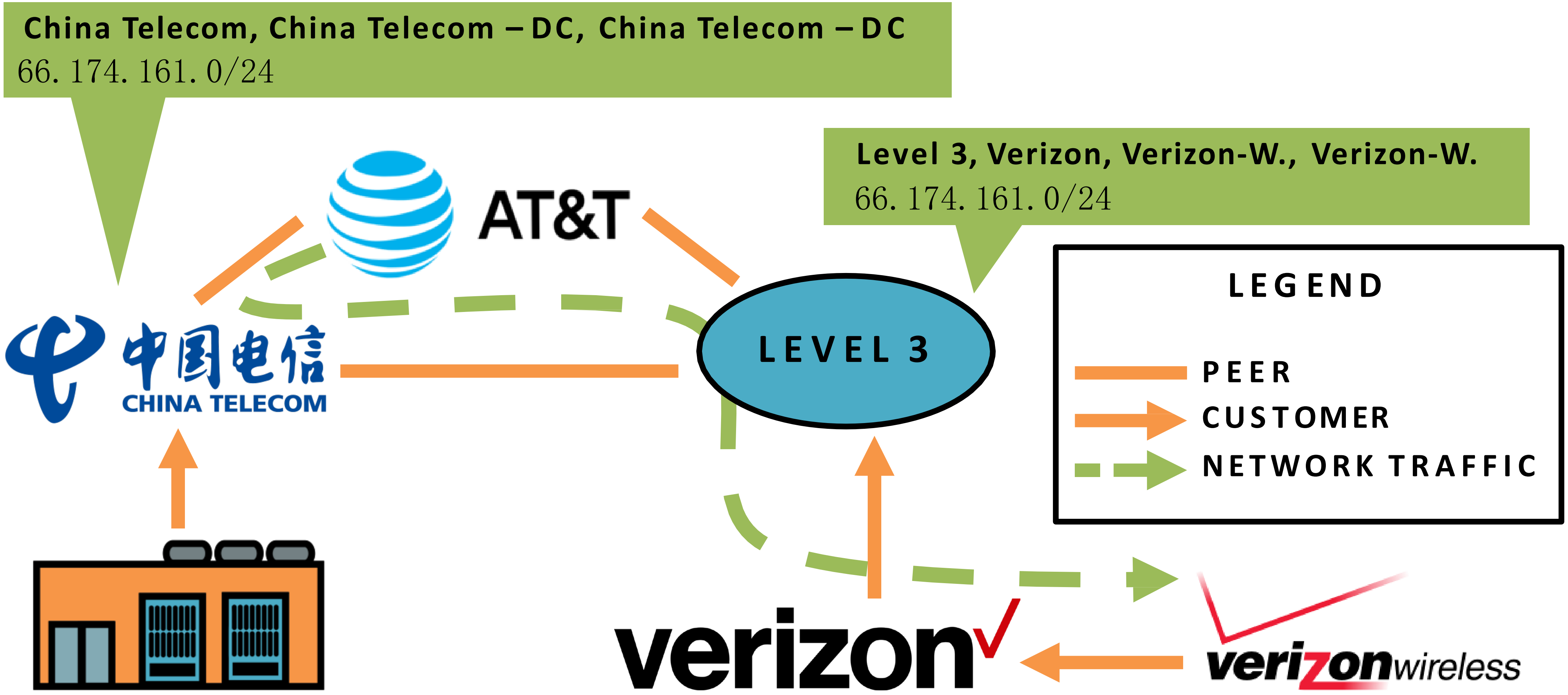
YouTube Timeframe:

- 100% at 10:30am
- 0% at 10:45am



DoS via route hijacking

DETOUR



DoS at Higher Levels

SSL/TLS handshake [SD’03]

RSA Encrypt

RSA Decrypt

O(RSA Encrypt) << O(RSA Decrypt)

DoS Mitigation

Client puzzles

Moderately hard problem:

- **Given challenge C find X such that**
  - $LSB_n(SHA\text{-}1(C \,||\, X)) = 0^n$
- **Assumption: takes expected 2^n time to solve**
  - For n=16 takes about 0.3sec on 1Ghz machine

Idea: slow down attacker

Main point: checking puzzle solution is easy.

DoS Mitigation

Client puzzles

During DoS attack:

- Everyone must submit puzzle solution with requests

When no attack:

- Do not require puzzle solution

DoS Mitigation

Client puzzles : Examples

TCP connection floods (RSA '99)

- Example challenge: C = TCP server-seq-num
- First data packet must contain puzzle solution
  - Otherwise TCP connection is closed

DoS Mitigation

Client puzzles : Examples

SSL handshake DoS: (SD’03)

- Challenge C based on TLS session ID
- Server: check puzzle solution before RSA decrypt

Same for application layer DoS and payment DoS

DoS Mitigation

Client puzzles : Benefits and limitations

Hardness of challenge: n

- Decided based on DoS attack volume

Limitations:

- Requires changes to both clients and servers
- Hurts low power legitimate clients during attack:
  - Clients on cell phones and tablets cannot connect

Puzzle Quiz

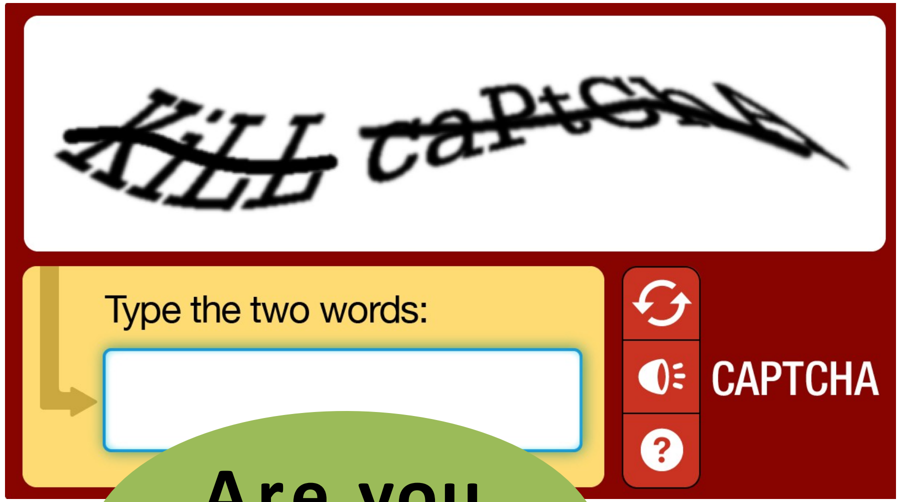
Which of the following statements are true?

- [ ] Client puzzles should be hard to construct. This is an indication of the level of difficulty to solve them.
- [x] Client puzzles should be stateless
- [x] Puzzle complexity should increase as the strength of the attack increases.

DoS Mitigation - CAPTCHAs

CAPTCHA

Completely Automated Public Turing test to tell Computers and Humans Apart



Are you human?

Idea: verify that connection is from a human

DoS Mitigation - CAPTCHAs

CAPTCHA

Completely Automated Public Turing test to tell Computers and Humans Apart

Applies to application layer DDoS [Killbots '05]

- During attack: generate CAPTCHAs and process request only if valid solution
- Present one CAPTCHA per source IP address

Idea: verify that connection is from a human

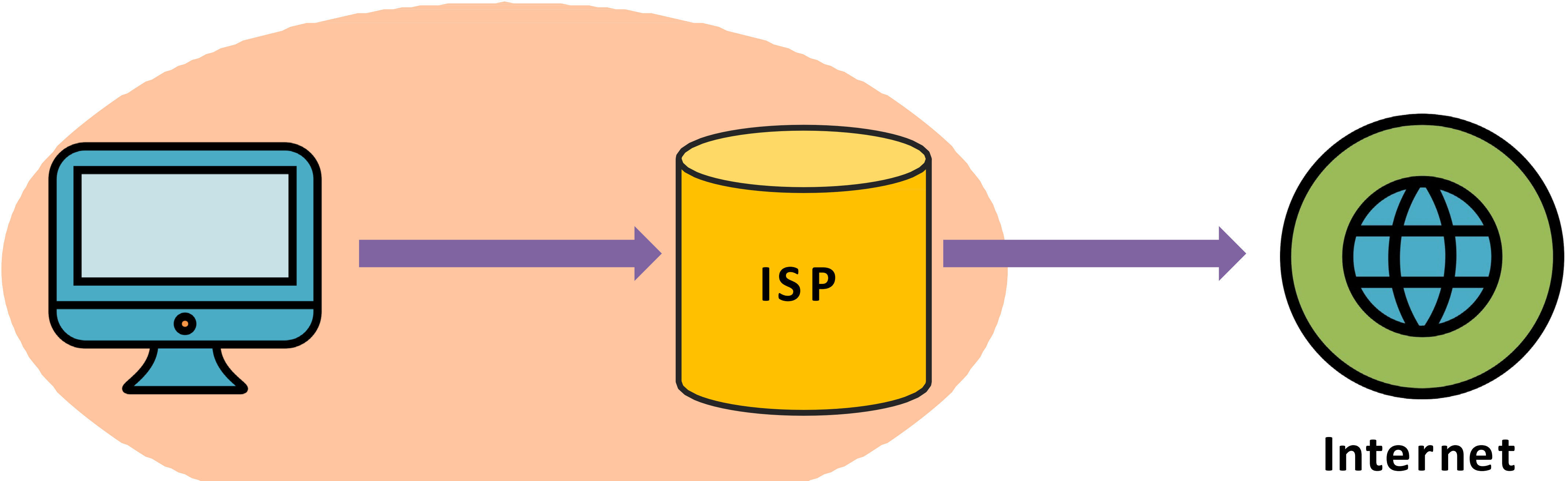
DoS Mitigation: Source Identification

<u>**Goal: identify packet source**</u>

Ultimate goal: block attack at the source

DoS Mitigation: Source Identification

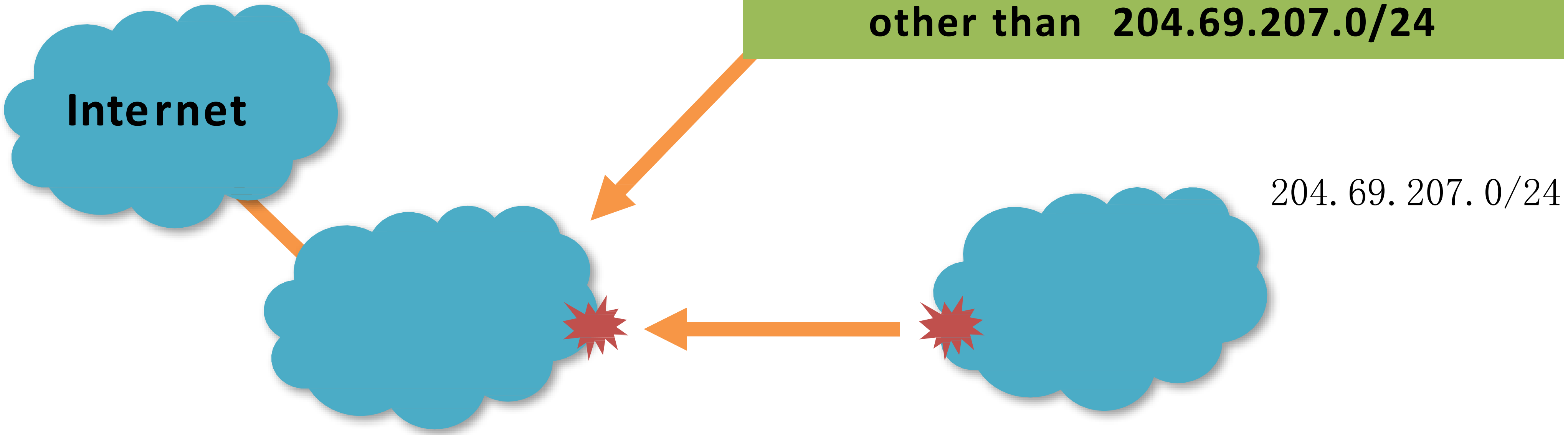
Ingress Filtering



DoS Mitigation: Source Identification

Ingress Filtering

Drop all packets with source address other than 204.69.207.0/24



Ingress filtering policy: ISP only forwards packets with legitimate source IP

DoS Mitigation: Source Identification

Ingress Filtering - Implementation problems

ALL ISPs must do this. Requires global trust.

- If 10% of ISPs do not implement → no defense
- No incentive for deployment

Recall: 309 Gbps attack used only 3 networks (3/2013)

2014:

- 25% of Auto. Systems are fully spoofable *(spoofer.cmand.org)*
- 13% of announced IP address space is spoofable

DoS Mitigation: Traceback

Traceback [Savage et al. ’00]

Goal:

- Given set of attack packets
- Determine path to source

How: change routers to record info in packets

Assumptions:

- Most routers remain uncompromised
- Attacker sends many packets
- Route from attacker to victim remains relatively stable

DoS Mitigation: Traceback

Simple Method:

Write path into network packet:

- Each router adds its own IP address to packet
- Victim reads path from packet

Problems:

- Requires space in packet
- Path can be long
- No extra fields in current IP format
  - Changes to packet format too much to expect

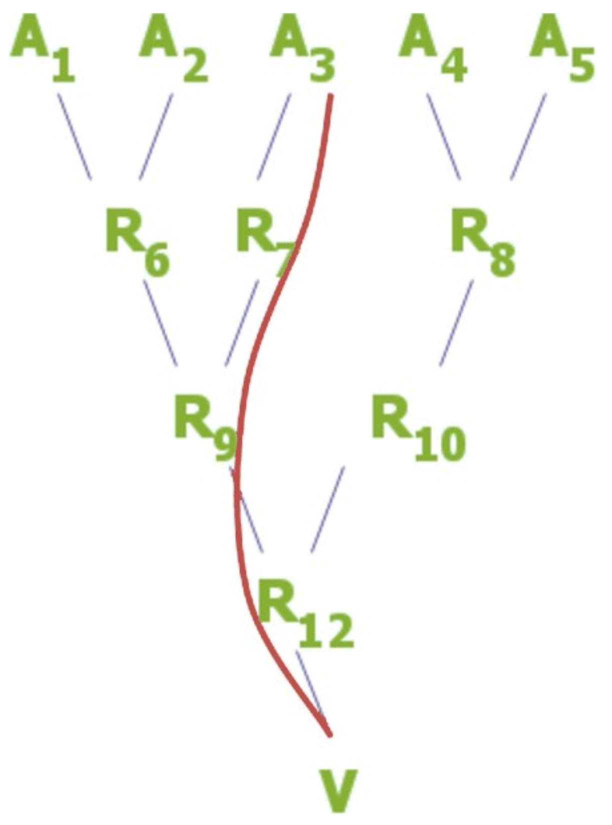
DoS Mitigation: Traceback

Better Idea

DDoS involves many packets on same path

Store one link in each packet

- Each router probabilistically stores own address
- Fixed space regardless of path length



Traceback Quiz

Which of the following are assumptions that can be made about Traceback?

- [ ] Attackers can generate limited types of packets
- [x] Attackers may work alone or in groups
- [ ] Attackers are not aware of the tracing mechanism

DoS Mitigation: Edge Sampling

Data fields written to packet:

- Edge: start and end IP addresses
- Distance: number of hops since edge stored

DoS Mitigation: Edge Sampling

Marking procedure for router R:

if coin turns up heads (with probability p) then

- **Write R into start address**
- **write 0 into distance field**

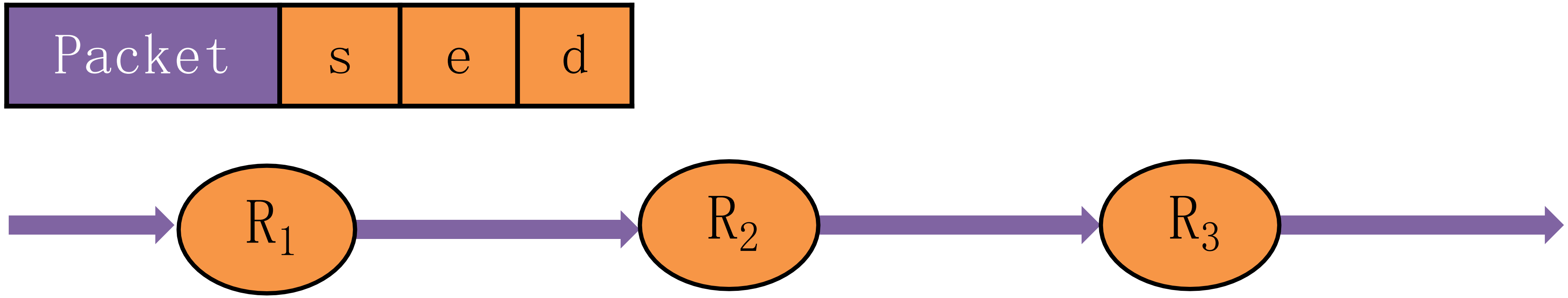
else

- **if** distance == 0 **write** R **into end field**
- **increment distance field**

DoS Mitigation: Edge Sampling

Packet received

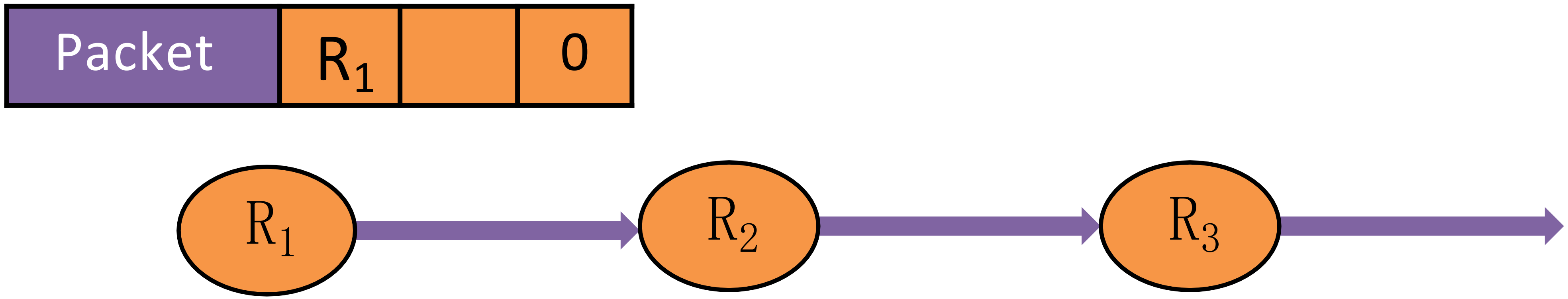
- R1 receives packet from source or another router
- Packet contains space for start, end, distance



DoS Mitigation: Edge Sampling

Begin writing edge

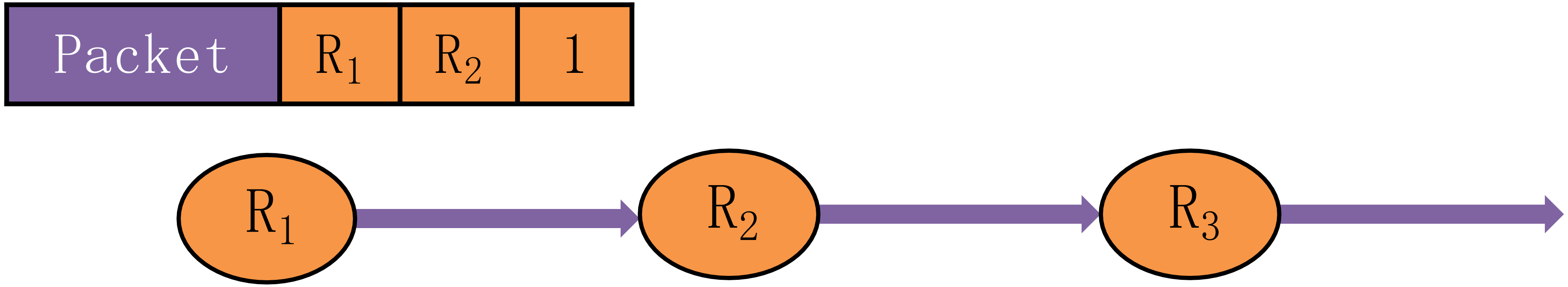
- R1 chooses to write start of edge
- R1 chooses to write start of edge Sets distance to 0



DoS Mitigation: Edge Sampling

Finish writing edge

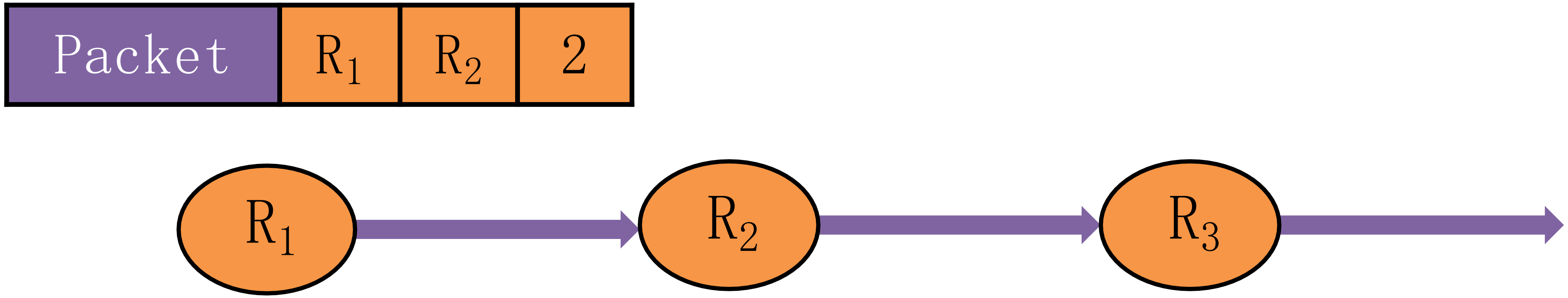
- R2 chooses not to overwrite edge
- Distance is 0
  - Write end of edge, increment distance to 1



DoS Mitigation: Edge Sampling

Increment distance

- R3 chooses not to overwrite edge
- Distance > 0
  - Increment distance to 2



DoS Mitigation: Edge Sampling

Path reconstruction

- Extract information from attack packets
- Build graph rooted at victim
  - Each (start,end,distance) tuple provides an edge
- # packets needed to reconstruct path

$$E(X) < \frac{\ln(d)}{p(1-p)^{d-1}}$$

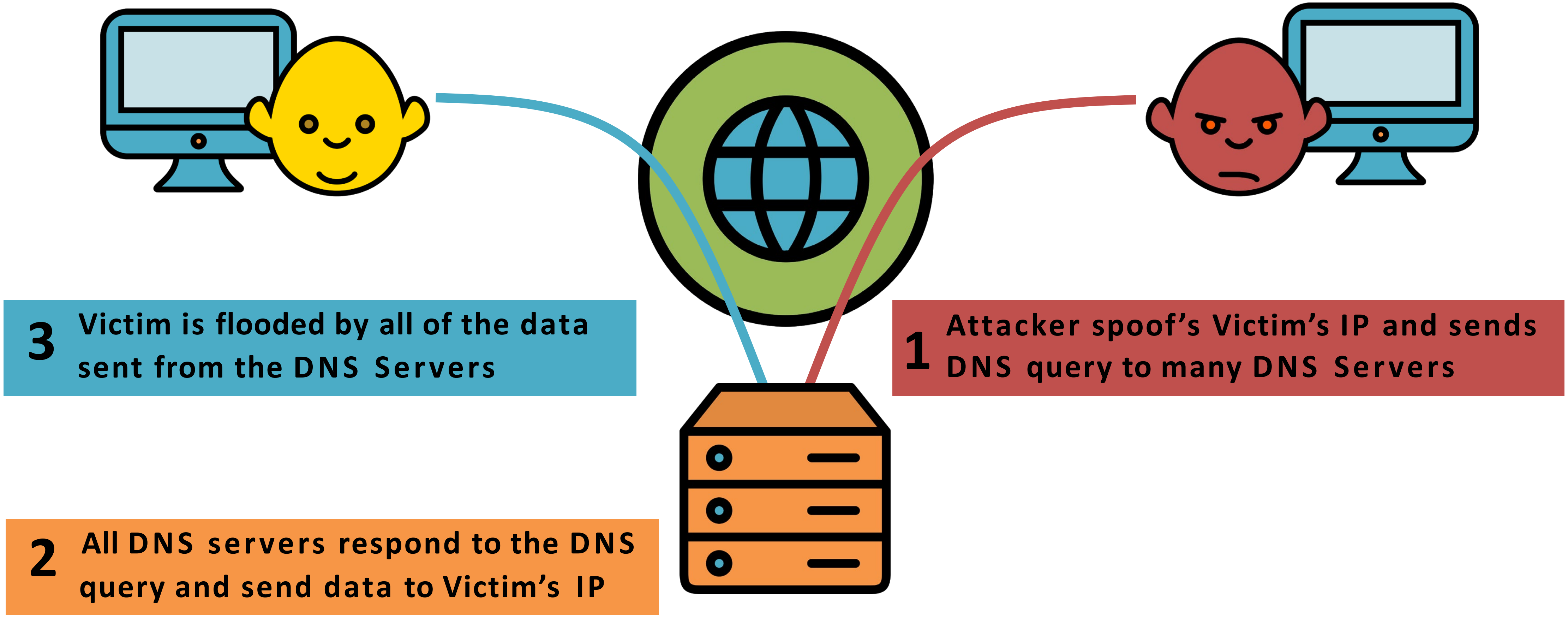
where p is marking probability, d is length of path

Edge Sampling Quiz

Select all the statements that are true for edge sampling:

- [x] Multiple attackers can be identified since edge identifies splits in reverse path
- [ ] It is difficult for victims to reconstruct a path to the attacker
- [x] Requires space in the IP packet header

Reflector Attack [Paxson '01]

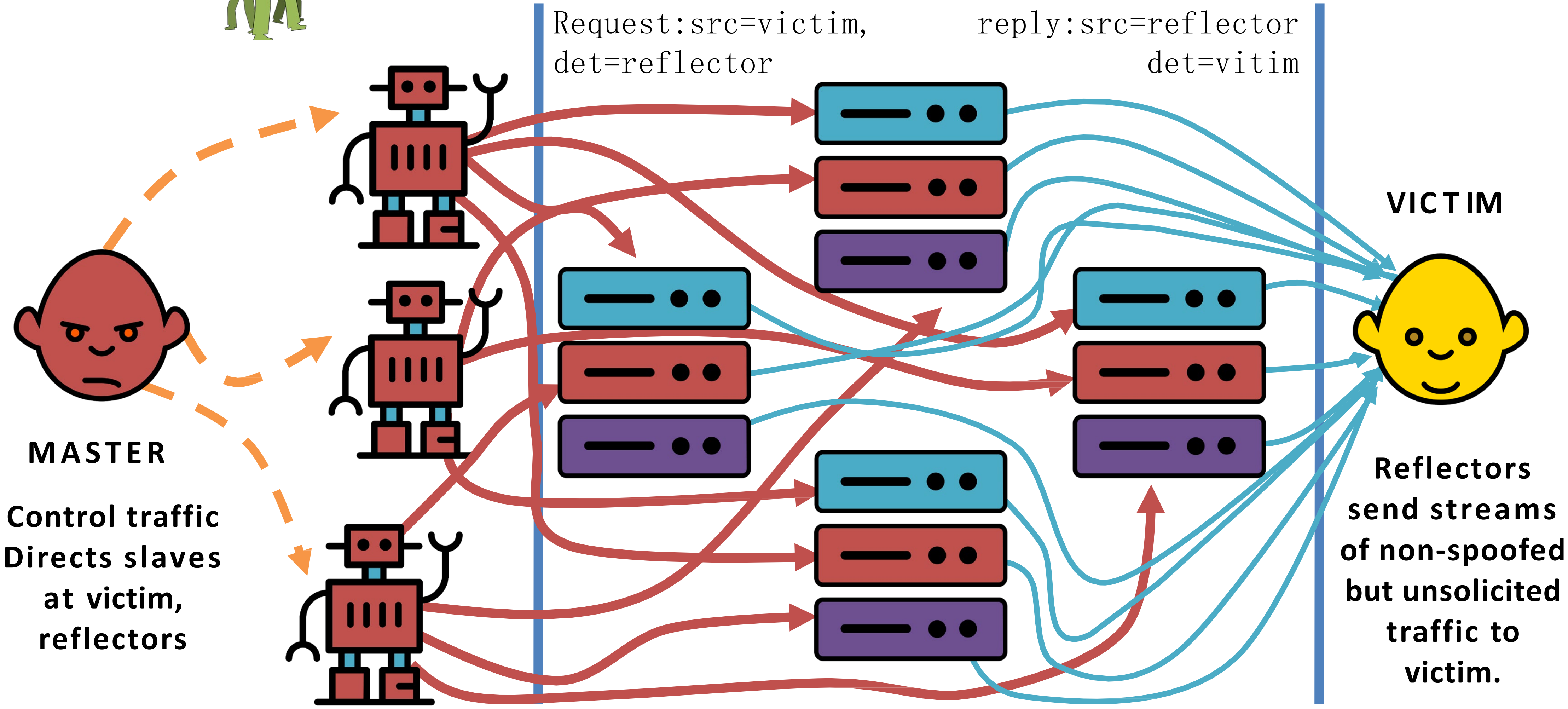


Reflector Attack [Paxson '01]

Examples:

- DNS Resolvers: UDP 53 with victim.com source
  - At victim: DNS response
- Web servers: TCP SYN 80 with victim.com source
  - At victim: TCP SYN ACK packet
- Gnutella servers

Reflector Attack [Paxson '01]



Reflector Attack Quiz

Self defense against reflector attacks should incorporate:

- [ ] Filtering - filter DNS traffic as close to the victim as possible.
- [x] Server redundancy - servers should be located in multiple networks and locations.
- [x] Traffic limiting - traffic from a name server should be limited to reasonable thresholds.

Capability Based Defense

Anderson, Roscoe, Wetherall	Preventing internet denial-of-service with capabilities. SIGCOMM ‘04.
Yaar, Perrig, and Song	Siff: A stateless internet flow filter to mitigate DDoS flooding attacks. IEEE S&P ’04.
Yang, Wetherall, Anderson	A DoS-limiting network architecture. SIGCOMM ’05

DoS Summary

Denial of Service attacks are real.

- Must be considered at design time.

Sad truth:

- Internet is ill-equipped to handle DDoS attacks
- Commercial solutions: CloudFlare, Prolexic

Many good proposals for Internet core redesign.