Network Security

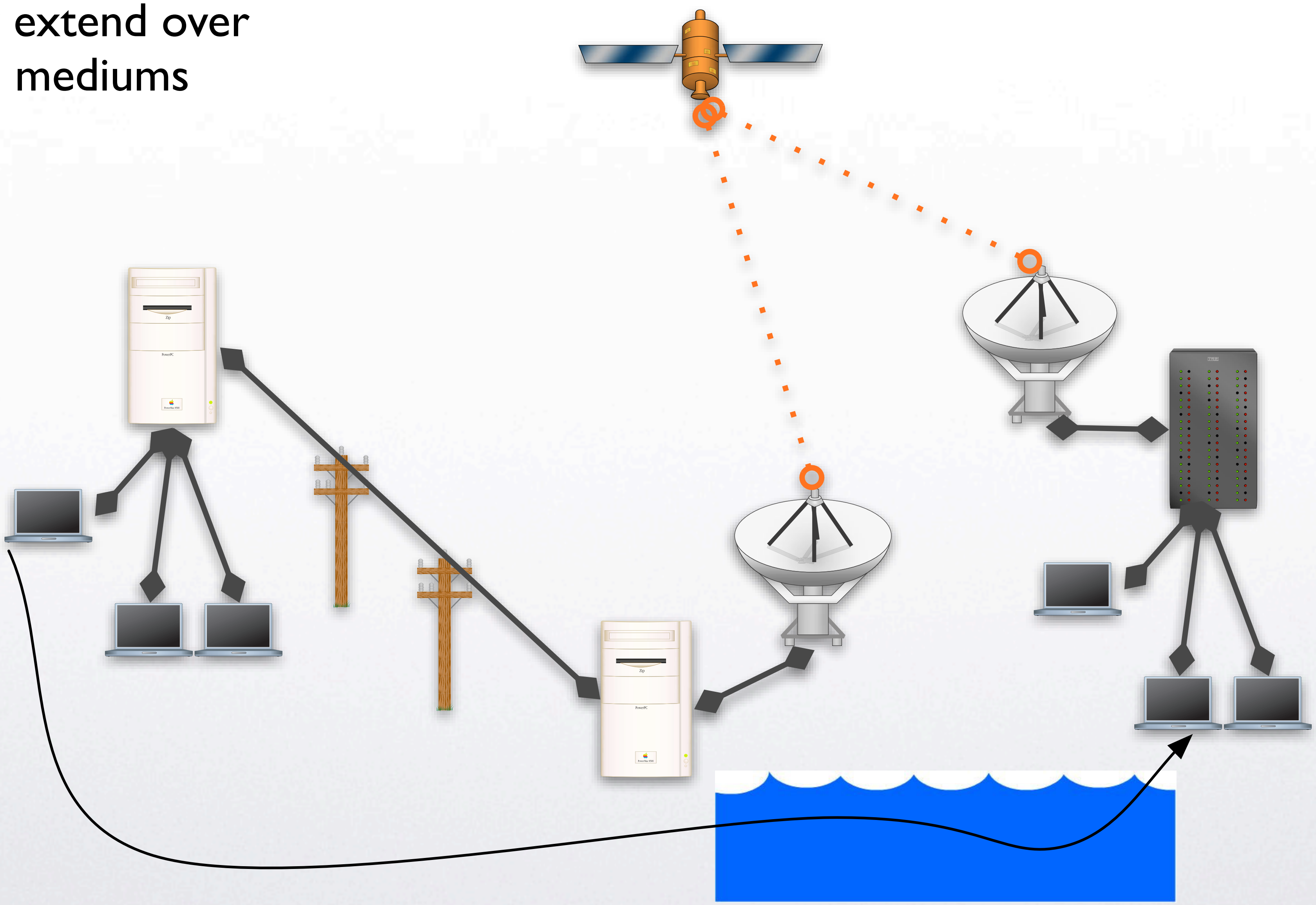
Aggelos Kiayias

Networks...

- Computers connected to each other.
- Each machine has a unique address.
- Each message from the sender to receiver may stop at many intermediate hops till it reaches its destination. ...routing...

Networks

A network may extend over various physical mediums



Communication Media

- Wire (copper wire: *cheap, slow*) 10 Mbps, ~100m. Carries electrical signal.
- Coax Cable (wire+insulation jacket) 100Mbps ~500m.
- Optical fiber (thin strand of glass). Carries pulses of light. 1000Mbps. ~4km.
- Radio signals. *Wireless*. good for short connections.
- microwave, infrared, satellite etc...

OSI Model

- Open Systems Interconnection Model

7	Application	User-level data
6	Presentation	data format (ascii etc.)
5	Session	sequencing
4	Transport	Flow control (acks, retransmissions errors)
3	Network	Routing (where to send)
2	Data Link	Local delivery
1	Physical	bit level representation

Example : e-mail

7	Application	e-mail composition
6	Presentation	text based transliteration, compression
5	Session	-
4	Transport	error-correcting codes, logical connection
3	Network	chop in packets - put addresses
2	Data Link	chop in frames - add MAC addresses
1	Physical	chop in bits - transmit

TCP/IP

- Transmission Control Protocol/ Internet Protocol.
- Four layers:
  - Application.
  - Host-to-Host Transport.
  - Internet.
  - Physical

TCP/IP

Application Layer	Prepare messages from user	Addressing/ Interaction
Transport Layer (e.g., TCP)	Packets are made	Sequencing, Reliability Error Correction
Network Layer (IP)	Into Datagrams	Routing
Data Link Layer	Connection between adjacent hosts - Bits	
Physical	Bit representation	

Data Link Layer Frames

- Source and destination Physical Addresses
- Encoding of bits
- Physical layer aspects (e.g., modulation).

IP Datagrams

- Contain *time to live* information (# of hops).
- Source and Destination IP addresses.
- Information about the encapsulated protocol.

TCP Packets

- Source / Destination ports.
- Acknowledgment number for connecting packets of a session.
- Sequence numbers.
- Integrity (checksums).

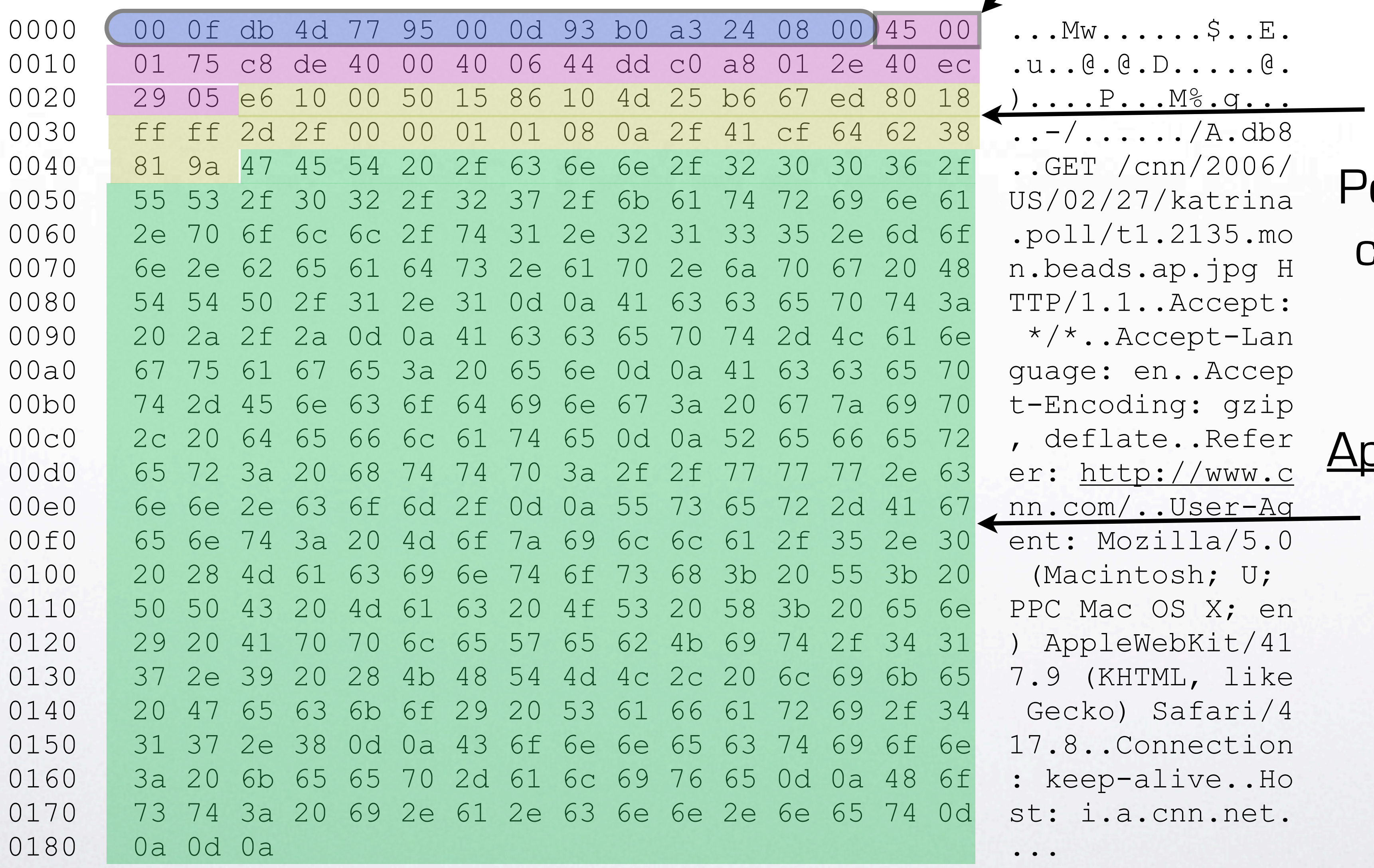
Application Data

- Depending on the application layer protocol used.

Example

Physical Layer: eth2
the 2 MAC addresses
+ IP indication

Network layer: IP
IP addresses, TTL,
checksum, fragmentation



```
0000  00 0f db 4d 77 95 00 0d 93 b0 a3 24 08 00 45 00   ...Mw......$..E.
0010  01 75 c8 de 40 00 40 06 44 dd c0 a8 01 2e 40 ec   .u..@.@.D.....@.
0020  29 05 e6 10 00 50 15 86 10 4d 25 b6 67 ed 80 18   )....P...M%.g...
0030  ff ff 2d 2f 00 00 01 01 08 0a 2f 41 cf 64 62 38   ..-/....../A.db8
0040  81 9a 47 45 54 20 2f 63 6e 6e 2f 32 30 30 36 2f   ..GET /cnn/2006/
0050  55 53 2f 30 32 2f 32 37 2f 6b 61 74 72 69 6e 61   US/02/27/katrina
0060  2e 70 6f 6c 6c 2f 74 31 2e 32 31 33 35 2e 6d 6f   .poll/t1.2135.mo
0070  6e 2e 62 65 61 64 73 2e 61 70 2e 6a 70 67 20 48   n.beads.ap.jpg H
0080  54 54 50 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a   TTP/1.1..Accept:
0090  20 2a 2f 2a 0d 0a 41 63 63 65 70 74 2d 4c 61 6e    */*..Accept-Lan
00a0  67 75 61 67 65 3a 20 65 6e 0d 0a 41 63 63 65 70   guage: en..Accep
00b0  74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 70   t-Encoding: gzip
00c0  2c 20 64 65 66 6c 61 74 65 0d 0a 52 65 66 65 72   , deflate..Refer
00d0  65 72 3a 20 68 74 74 70 3a 2f 2f 77 77 77 2e 63   er: http://www.c
00e0  6e 6e 2e 63 6f 6d 2f 0d 0a 55 73 65 72 2d 41 67   nn.com/..User-Ag
00f0  65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30   ent: Mozilla/5.0
0100  20 28 4d 61 63 69 6e 74 6f 73 68 3b 20 55 3b 20    (Macintosh; U;
0110  50 50 43 20 4d 61 63 20 4f 53 20 58 3b 20 65 6e   PPC Mac OS X; en
0120  29 20 41 70 70 6c 65 57 65 62 4b 69 74 2f 34 31   ) AppleWebKit/41
0130  37 2e 39 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65   7.9 (KHTML, like
0140  20 47 65 63 6b 6f 29 20 53 61 66 61 72 69 2f 34    Gecko) Safari/4
0150  31 37 2e 38 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e   17.8..Connection
0160  3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d 0a 48 6f   : keep-alive..Ho
0170  73 74 3a 20 69 2e 61 2e 63 6e 6e 2e 6e 65 74 0d   st: i.a.cnn.net.
0180  0a 0d 0a                                          ...
```

Transport Layer: TCP
Ports, Seq Ack numbers,
checksum, timestamps

Application Layer: HTTP
Request: GET
Request URI
Referrer
User-agent info
Connection info

Internet Protocols

- Data link Layer: ethernet, wi-fi etc.
- Network Layer: ICMP (Internet control message), IP etc.
- Transport Layer: UDP (user datagram protocol), TCP etc.
- Application Layer: Finger, FTP (file transfer), HTTP (hypertext transfer), IMAP (internet message access), IRC (internet relay chat), POP (post office), SMTP (simple mail transfer), TELNET (terminal emulation), X-window, etc.

UDP Protocol

- user datagram protocol.
- lightweight alternative to TCP.
- Faster, ligher - adds 8 bytes for control.
- stateless sending and no ordering.
- used for application layer protocols as SNMP (simple network monitoring), Syslog (system audit log), Time etc.

How do data find their way?

- *Application.* HTTP request to www.website.com (DNS resolution)
- *Transport.* which in turn will result to some packets directed to a certain **port #**.
- *Internet.* which in turn will result to some frames directed to a **IP address.**
- *Physical.* which in turn will result to some actual bits being sent to **MAC address**

Addressing

- Two mappings are necessary:
    - From host name to IP address.
    - From IP address to MAC address.
- Host name will be mapped to an IP address through a protocol called DNS
- MAC address will be obtained from an address resolution table.

Transmitting a packet

- A packet needs to be directed to a certain IP address.
- To figure out where to send it next, a routing table is consulted. Example:

```
Routing table:
Destination        Gateway            Flags    Refs      Use  Netif Expire
default            137.99.11.1        UGSc       16     1863    en0
...
```

Transmitting a packet, II

- Once the (intermediate hop) IP address is determined the packet must be split into frames and directed to the right MAC address.
- Internet to Ethernet address translation:

```
Address                  HWtype  HWaddress           Flags Mask            Iface
137.99.11.1              ether   00:0B:46:9A:1B:3F   C                     eth0
```

Receiving a packet

- Keep it or forward it.
- Based on destination address (and perhaps other parameters).
- Forward it using the routing table as before.
- Routing and Address Resolution tables are dynamically updated.

IP Addresses and DNS

- 32-bit (IPv6 will offer 128-bits).
- IP address correspond to names according to the Domain Name Service (DNS).
- Given a certain name at the application layer a query will be transmitted to a *Name Server* to resolve it for the corresponding IP address.

Trace route

```
 1  192.168.63.11 (192.168.63.11)  3.896 ms  2.122 ms  1.511 ms
 2  195.134.67.1 (195.134.67.1)  1.794 ms  2.839 ms  1.784 ms
 3  grnetRouter.L1.uoa.athens-3.access-link.grnet.gr (194.177.209.97)  1.984 ms  2.262 ms  3.360 ms
 4  eie2-to-koletti1.backbone.grnet.gr (195.251.27.46)  2.196 ms  4.345 ms  3.539 ms
 5  core1.ams.net.google.com (195.69.144.247)  72.084 ms  72.003 ms  72.743 ms
 6  209.85.248.88 (209.85.248.88)  74.182 ms *  74.916 ms
 7  64.233.175.246 (64.233.175.246)  75.377 ms  75.800 ms  74.950 ms
 8  209.85.255.143 (209.85.255.143)  77.161 ms 72.14.239.197 (72.14.239.197)  79.314 ms 209.85.255.166
(209.85.255.166)  90.792 ms
 9  72.14.232.37 (72.14.232.37)  81.473 ms 72.14.232.41 (72.14.232.41)  84.950 ms 72.14.232.37
(72.14.232.37)  84.960 ms
10  ez-in-f104.1e100.net (66.102.13.104)  79.149 ms  78.015 ms  76.316 ms
```

How traceroute works

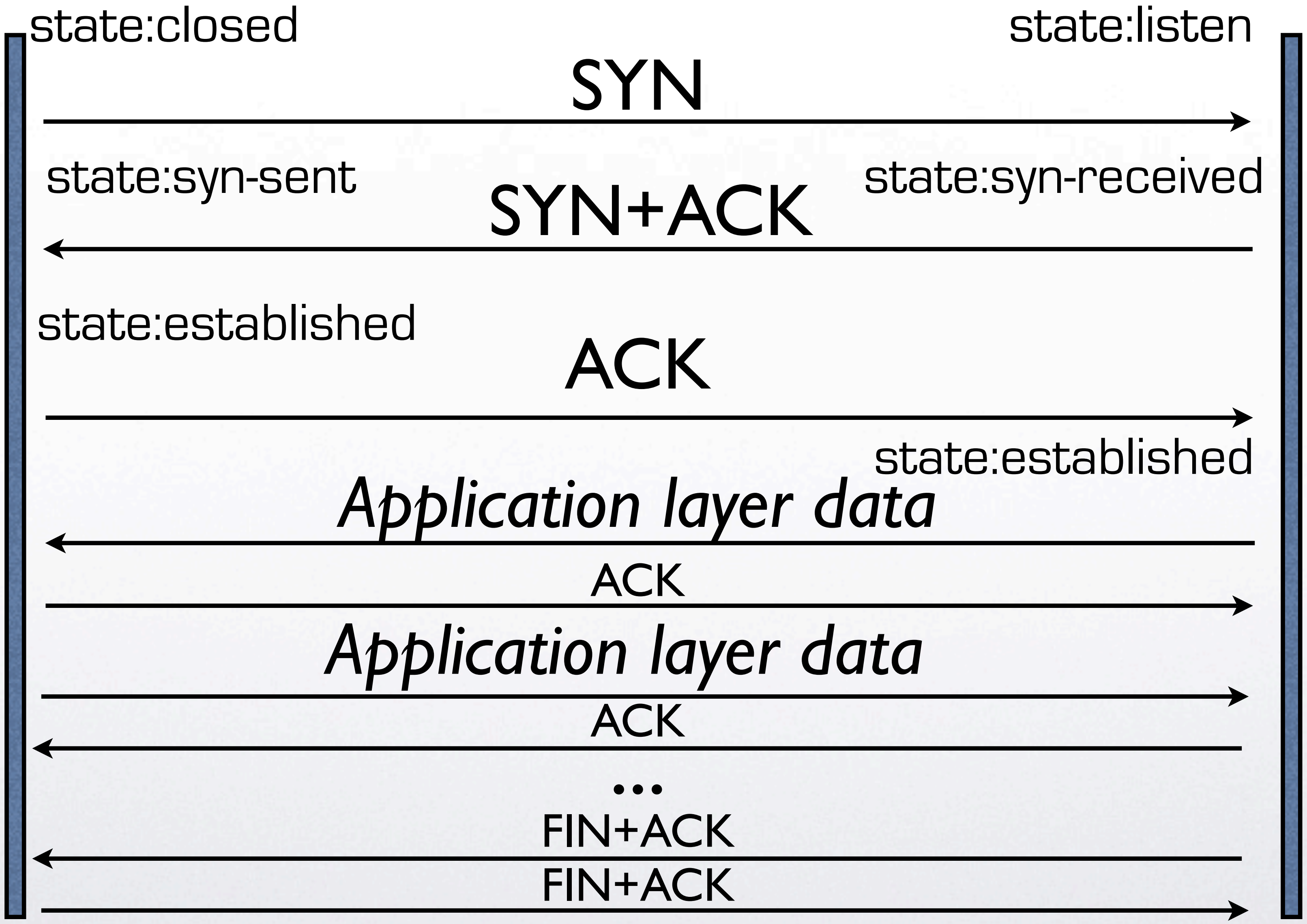
- Using a special IP header filed called TTL : time to live.
- TTL = number of hops a packet is allowed to make. Each router decreases by one.
- When TTL reaches 0 then a router discards the packet and notifies originator.
- For traceroute: send repeatedly packets and calibrate TTL as 1, 2, 3, 4, 5, ...
- not all routers necessarily respond (* * *)

Client Server Model

- Application protocols, FTP, HTTP, Telnet etc.
- Server listens to port for client requests.
- Client initiates protocol

Talking over TCP/IP

Client



state:closed

state:listen

SYN

state:syn-sent

state:syn-received

SYN+ACK

state:established

ACK

state:established

Application layer data

ACK

Application layer data

ACK

...

FIN+ACK

FIN+ACK

Server

Packet Sniffing

- Every computer in the Internet sends and receives packets.
- Anyone with the appropriate privileges in a certain host can “sniff” the packets that are being forwarded by the host (and not only the packets that are directed to the host).
- In this case the host is said to be in promiscuous mode.

Packet Sniffing, II

- If you have privileges for promiscuous mode then you can capture all traffic within the sub-network the host belongs to.

- You cannot capture traffic outside your sub-network (e.g., traffic that is not directed towards your gateway).

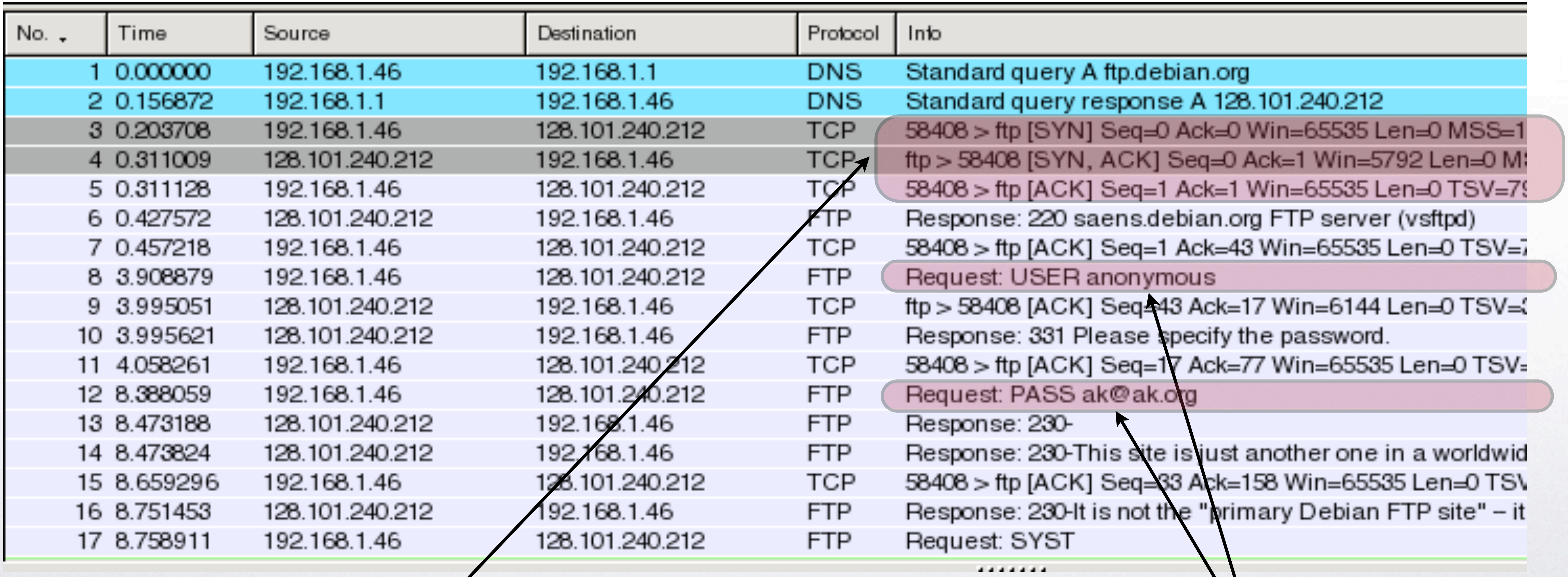
- When you use your computer do always ponder what is your sub-network (consider: sitting at a cafe connected to a wireless access point)

Wireshark

- Wireshark is a powerful “network protocol analyzer”
- Not only it sniffs data when put on promiscuous mode but also “knows” the protocols and structures the packets in the appropriate format.

Wireshark, II

screen dump of capture window after an FTP connection

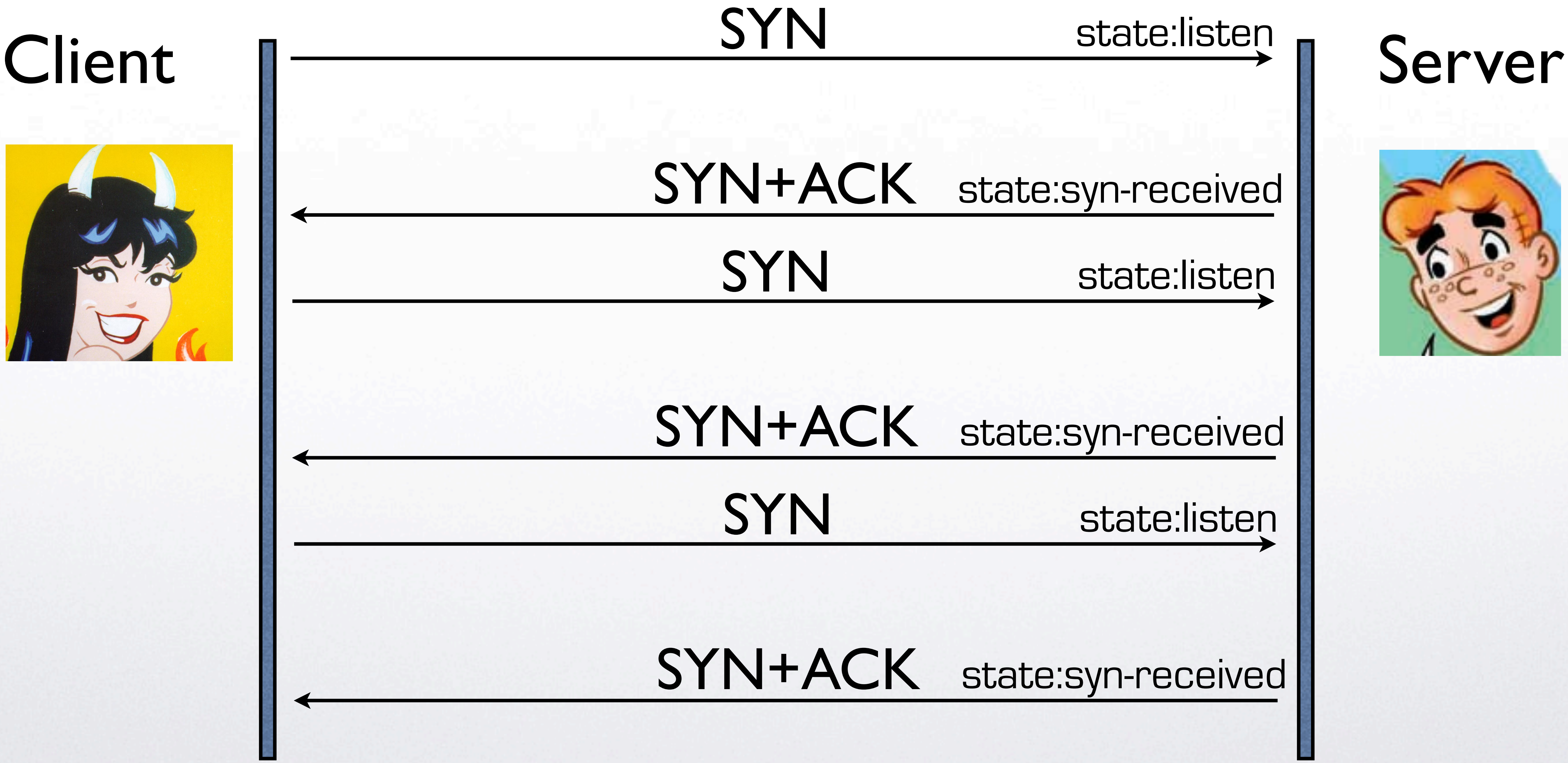


No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.46	192.168.1.1	DNS	Standard query A ftp.debian.org
2	0.156872	192.168.1.1	192.168.1.46	DNS	Standard query response A 128.101.240.212
3	0.203708	192.168.1.46	128.101.240.212	TCP	58408 > ftp [SYN] Seq=0 Ack=0 Win=65535 Len=0 MSS=1
4	0.311009	128.101.240.212	192.168.1.46	TCP	ftp > 58408 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 M
5	0.311128	192.168.1.46	128.101.240.212	TCP	58408 > ftp [ACK] Seq=1 Ack=1 Win=65535 Len=0 TSV=7
6	0.427572	128.101.240.212	192.168.1.46	FTP	Response: 220 saens.debian.org FTP server (vsftpd)
7	0.457218	192.168.1.46	128.101.240.212	TCP	58408 > ftp [ACK] Seq=1 Ack=43 Win=65535 Len=0 TSV=
8	3.908879	192.168.1.46	128.101.240.212	FTP	Request: USER anonymous
9	3.995051	128.101.240.212	192.168.1.46	TCP	ftp > 58408 [ACK] Seq=43 Ack=17 Win=6144 Len=0 TSV=
10	3.995621	128.101.240.212	192.168.1.46	FTP	Response: 331 Please specify the password.
11	4.058261	192.168.1.46	128.101.240.212	TCP	58408 > ftp [ACK] Seq=17 Ack=77 Win=65535 Len=0 TSV=
12	8.388059	192.168.1.46	128.101.240.212	FTP	Request: PASS ak@ak.org
13	8.473188	128.101.240.212	192.168.1.46	FTP	Response: 230-
14	8.473824	128.101.240.212	192.168.1.46	FTP	Response: 230-This site is just another one in a worldwid
15	8.659296	192.168.1.46	128.101.240.212	TCP	58408 > ftp [ACK] Seq=33 Ack=158 Win=65535 Len=0 TSV
16	8.751453	128.101.240.212	192.168.1.46	FTP	Response: 230-It is not the "primary Debian FTP site" – it
17	8.758911	192.168.1.46	128.101.240.212	FTP	Request: SYST

the three-way handshake

Observe the password and username

A malicious client



Client

Server

SYN state:listen

SYN+ACK state:syn-received

SYN state:listen

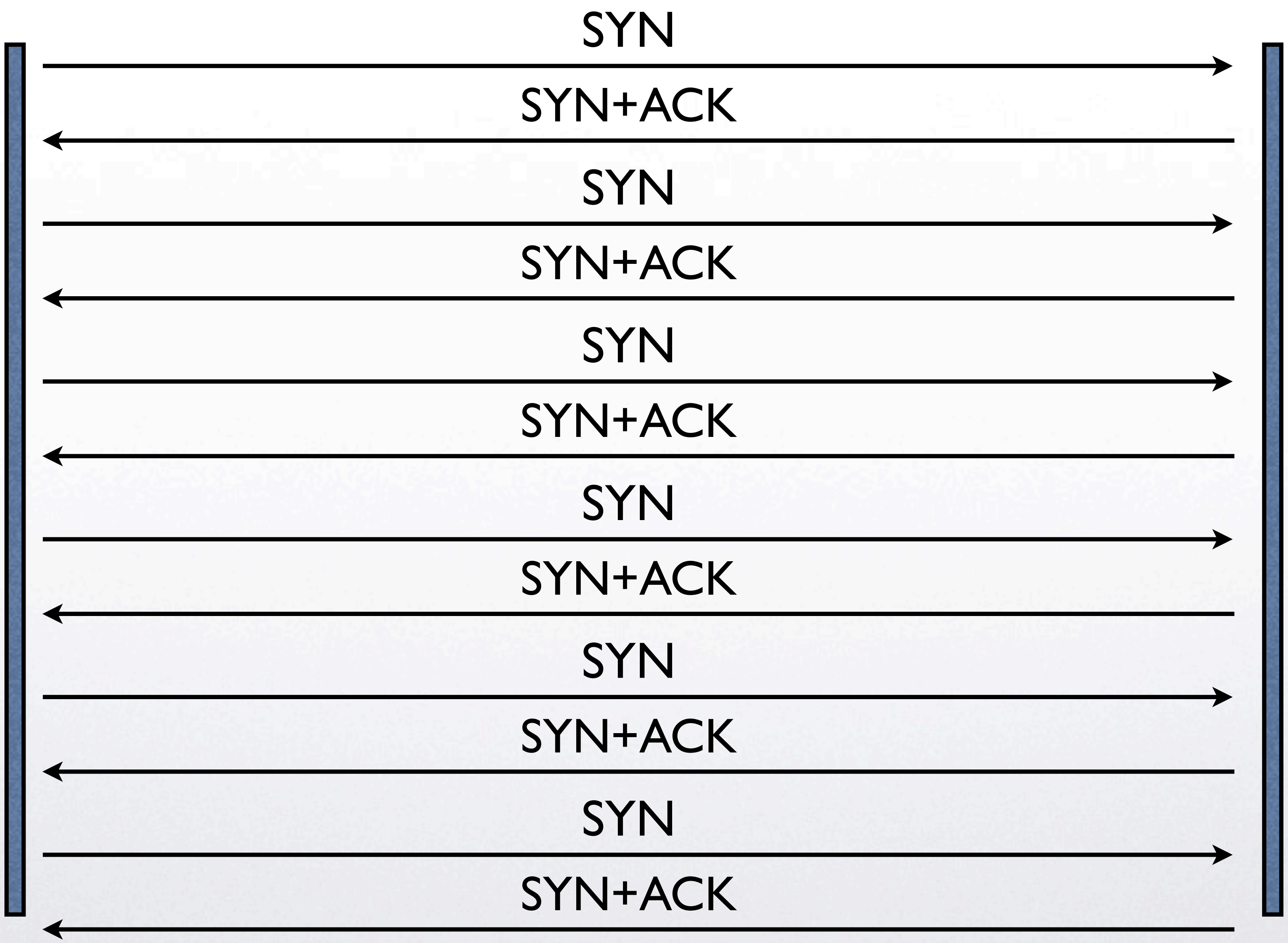
SYN+ACK state:syn-received

SYN state:listen

SYN+ACK state:syn-received

A malicious client, II

Client



SYN

SYN+ACK

SYN

SYN+ACK

SYN

SYN+ACK

SYN

SYN+ACK

SYN

SYN+ACK

SYN

SYN+ACK

Server

IP Spoofing

SYN Flooding

- Client bombards Server with SYN packets that are never ACK'ed.
- IP spoofing can be used to make packets look their coming from other places.
- Physical limitation: *bandwidth*.
- If bandwidth on client is substantial compared to server there is serious potential for a *Denial of Service (DoS) Attack*

DoS Attacks

- Deplete / misconfigure / misallocate the resources on a target server host so that it cannot serve its clients.
    - resources:
        - bandwidth.
        - memory.
        - cpu
        - ...

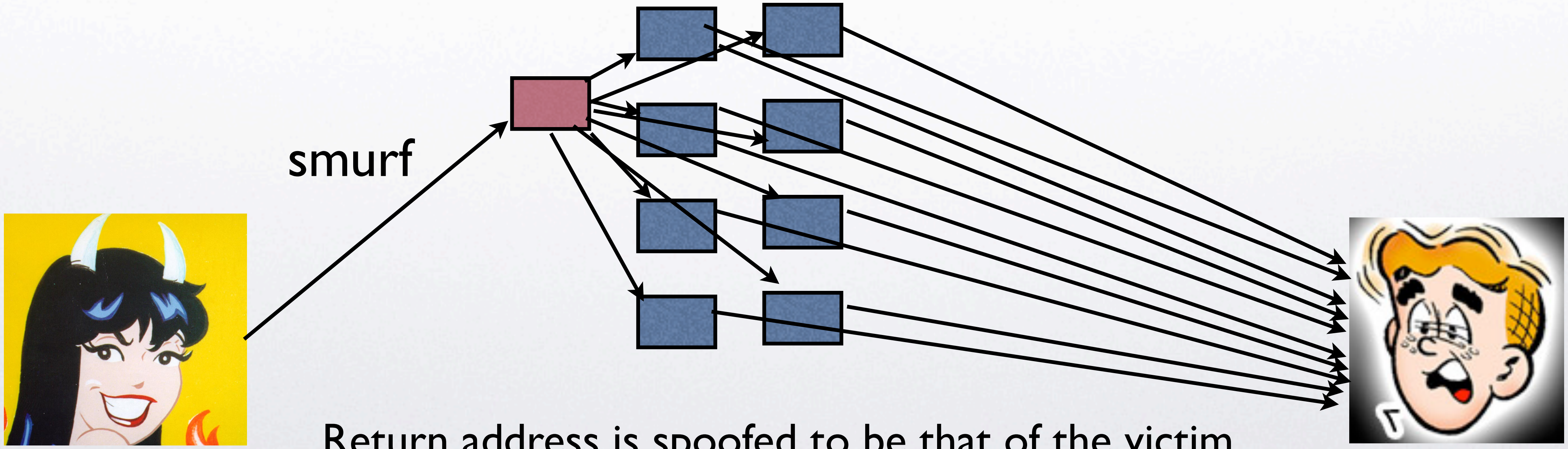
DoS By Flooding

- SYN Flooding - we just saw it.
- Ping Floods: this is a flooding of ICMP Echo Request packets.

```
aggelos@grub:~$ ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.686 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.611 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.617 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.611/0.638/0.686/0.034 ms
```

Smurf Attack

- An enhanced Ping flood attack that utilizes IP Broadcast:
  - it is possible to specify the destination IP address as a broadcast to all hosts in a subnetwork.



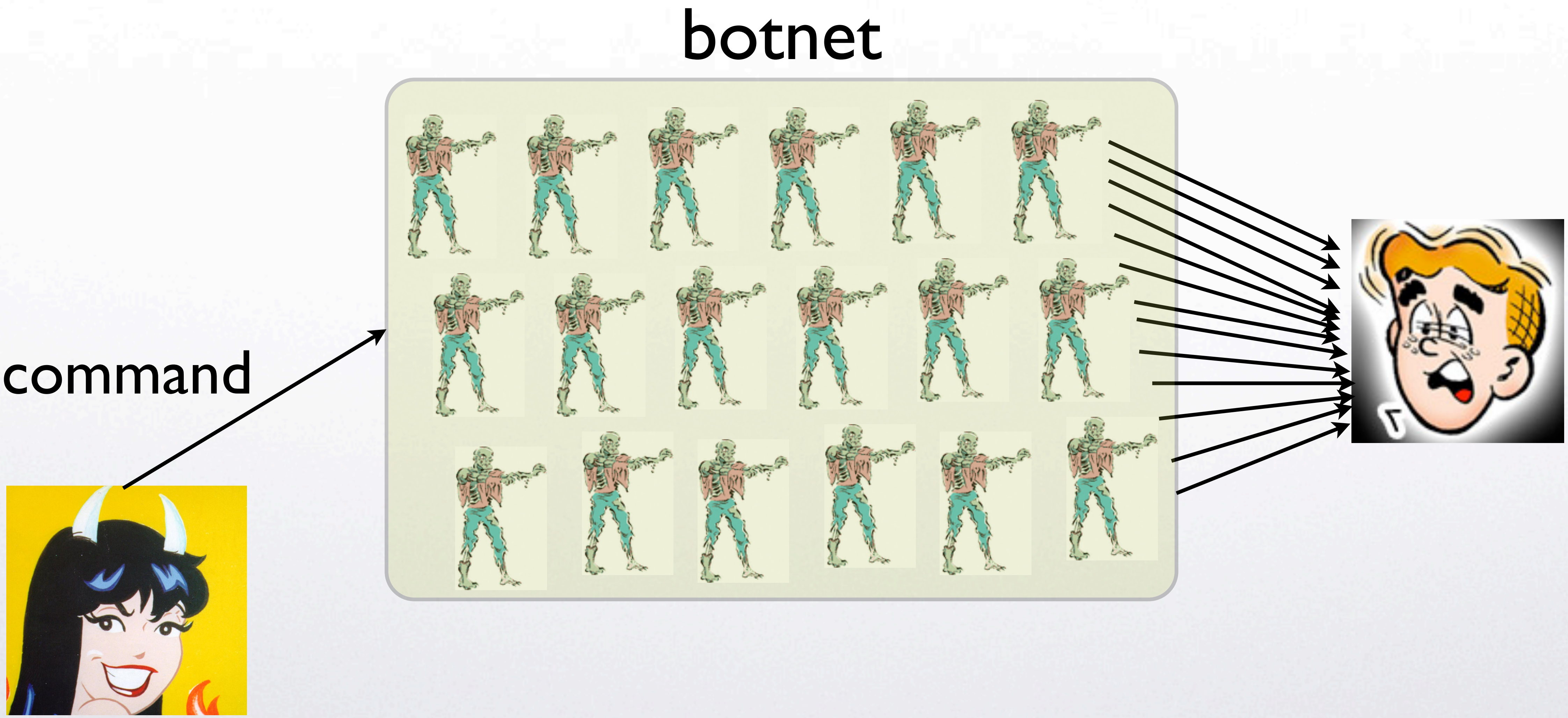
Return address is spoofed to be that of the victim.

DoS by Crashing

- Any action of a client that may crash a server results in a DoS.
  - Buffer overflows.
  - Once upon a time: ***Ping of Death*** delivering a ping packet of size greater than 64Kb used to crash systems in good old times1997-98.

Distributed DoS

- The real deal!



botnet

command

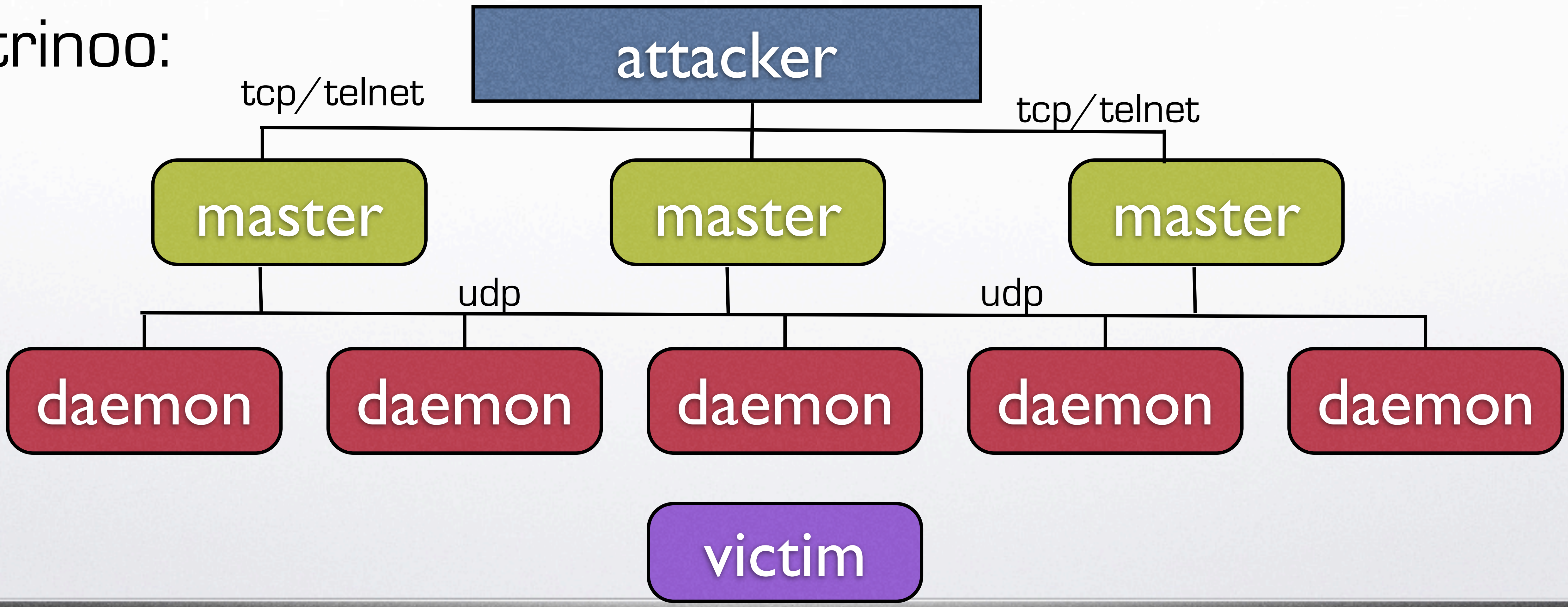
Zombification

- Host compromised by virus, worm, trojan.
- Runs rootkit remote administration tool.
- When commanded it launches DoS attack against victim.
- Attack seems to be coming from everywhere!

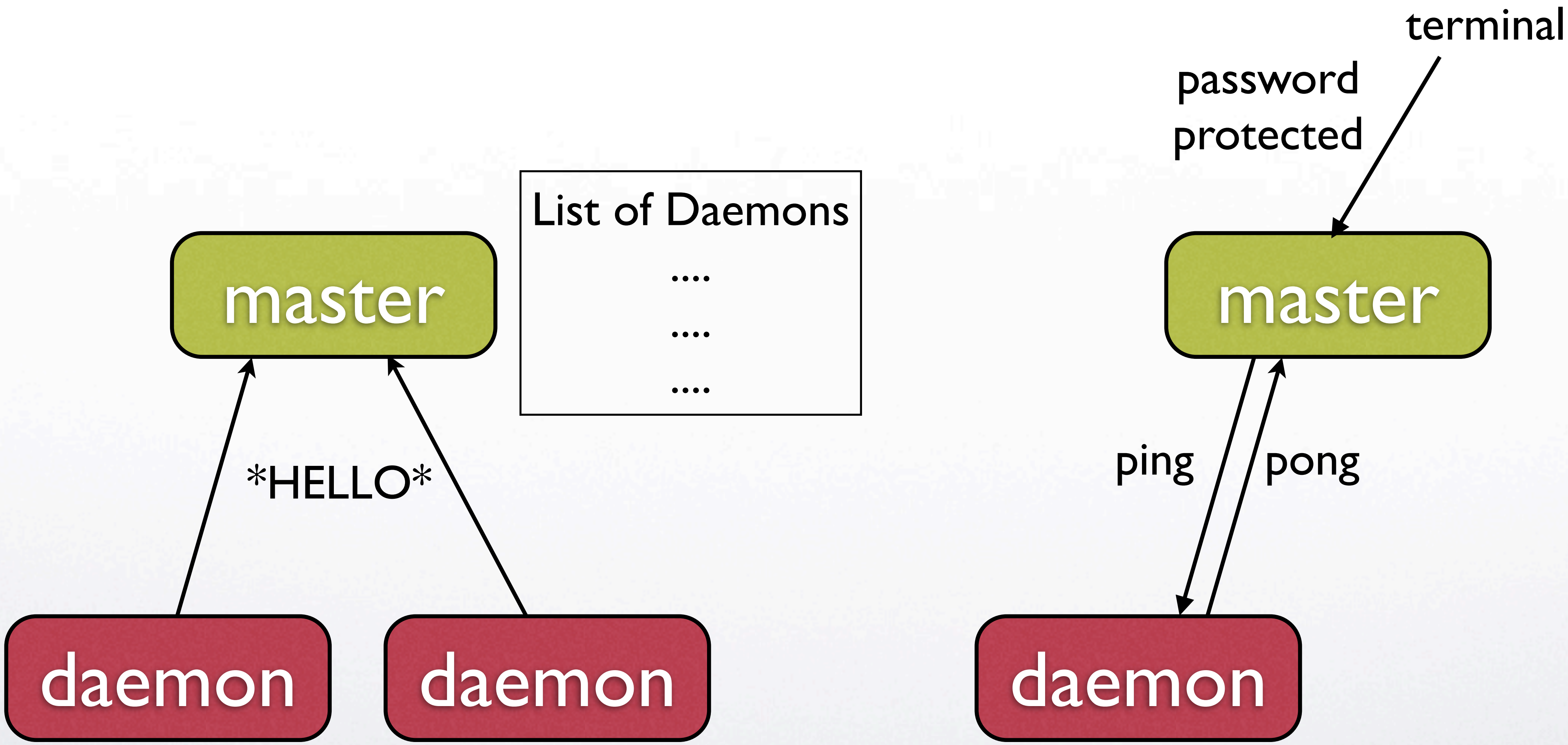
Tools of the trade.

- trinoo (trin00)/ wintrinoo, TFN2K (tribal flood network 2K), Stacheldraht, etc.

trinoo:



trinoo networks



http://staff.washington.edu/dittrich/misc/trinoo.analysis

Logging in to a trinoo master

```
attacker$ telnet 10.0.0.1 27665
Trying 10.0.0.1
Connected to 10.0.0.1
Escape character is '^]'.
kwijibo
Connection closed by foreign host.
 . . .

attacker$ telnet 10.0.0.1 27665
Trying 10.0.0.1
Connected to 10.0.0.1
Escape character is '^]'.
betaalmostdone
trinoo v1.07d2+f3+c..[rpm8d/cb4Sx/]


trinoo>
```

special port

correct password

trinoo commands

```
dos IP DoS the IP address specified.  A command ("aaa l44adsl
       IP") is sent to each Bcast host (i.e., trinoo daemons)
       telling them to DoS the specified IP address.

mdos <ip1:ip2:ip3>
       Multiple DoS.  Sends a multiple DoS command
       ("xyz l44adsl 123:ip1:ip2:ip3") to each Bcast host.

bcast     List all active Bcast hosts.

mdie pass Disable all Bcast hosts, if the correct password is
       specified. A command is sent ("d1e l44adsl") to each
       Bcast host telling them to shut down.  A separate
       password is required for this command.
```

DDoS Tools

- TFN2k:
  - like trinoo but with more attack capabilities. spoofed IP addresses.
- Stacheldraht
  - like trinoo and TFN2k but more capabilities, encrypted communication.

Command & Control

- Hard code address of master process into zombie code (obvious disadvantage)
- Internet Relay Chat (IRC). An IRC server hosts a number of channels.
  - a bot can incorporate a stripped IRC client to login to a certain channel.
  - Once logged on, wait for commands.

Botnet statistics

- 1000 Bots with 128KBit/s upstream > 100 MBits/s *[for such kinds of bandwidth a company may pay >>$10K per month]*
- Several large botnets of > 50,000 bots have been observed in the wild. Recently even larger sizes.
- November 2004 - January 2005, specific study *(see below)* observed 226 DDoS-attacks against 99 unique targets.
- Some botnet operators are very unskilled (destroying their botnets by simple mistakes, lacking basic programming skills etc.)

http://www.honeynet.org/papers/bots/

In the news...

- January 24, 2006:
  - J. J. Ancheta, 20 yr. pleads guilty to charges of conspiracy, damaging computers used by the US government and fraud.
  - faces max. sentence of 25 years.
  - controlled >100,000 bots, used for various illegal activities.
  - As part of the plea he forfeits the proceeds of his illicit activities that include $60,000 cash, a BMW, computer equipment.

http://news.zdnet.com/2100-1009_22-6030270.html

In the news...

03-18-05

NEWARK, N.J. - A Michigan man was arrested today on a federal charge that he hired a New Jersey juvenile to conduct highly destructive computer attacks on competitors of his online sportswear business, including a web-based New Jersey company.

...

According to the complaint, the computer attacks were conducted by the New Jersey juvenile from the juvenile's home computer. The juvenile secretly infected thousands of computers with copies of a computer program known as a "bot" (short for "robot"). As described in the complaint, a "bot" can have legitimate functions, but can also be used to gain unauthorized access to and control over computers that they infect, and can thus cause the infected computers to attack other computers. "Bots" used for such illicit purposes are frequently disguised as MP3 music files or photographs that unsuspecting computer users download from public Internet sites. Having downloaded an infected file, a computer user is usually unaware of the presence of a "bot" on his or her computer.

http://www.usdoj.gov/usao/nj/publicaffairs/NJ_Press/files/arab0318_r.htm

11-10-2004

...

You have 2 choices," Card Services International was told via e-mail earlier this year. "You can ignore this email and try to keep your site up, which will cost you tens of thousands of dollars ... or you can send us $10K by Western Union to make sure your site experiences no problem. If you choose not to pay for our help, then you will probably not be in business much longer, as you will be under attack each weekend for the next 20 weeks."

It wasn't a bluff. The Kentucky-based credit card processing firm suffered about a week's worth of outages before blocking the attack, according to president Jay Broder. The firm didn't pay, and the FBI is investigating the incident.

http://www.msnbc.msn.com/id/6436834/

Botnet Uses Beyond DDoS

- The convenience of separation between ‘hacking a computer’ and ‘committing a crime.’
- Sending Spam.
- Click-Fraud.
- Identity Theft.
- Stealing files: e.g., game Diablo-2 items were stolen and sold to EBay..

we will return to those later

Other DoS attacks

- The Fork Bomb: any program that constantly forks by creating child processes that do the same.
- Any time a process is called the O/S allocates memory for the process’s requirements and enters the process specifics in a data structure.

Example:

```
int main(void)
{
  while(1) {
    fork();
  }
  return 0;
}
```

this little program may crash your linux-box

You can write fork-bombs for all major languages.

Defending against DOS, I

- Attacks like Smurf, Fork Bombs etc. result from the ability of an entity to allocate more resources than necessary in normal operation.
- Restricting such capability will thwart the related attack, e.g.,
  - restrict the use of IP broadcasting.
  - restrict the number of processes a user may create.

Defending against DOS, II

- Being proactive also helps:
    - A network router may detect outbound traffic that serves a DoS attack (e.g., outbound traffic with randomly spoofed source IP address. Such traffic can be dropped before leaving the network).

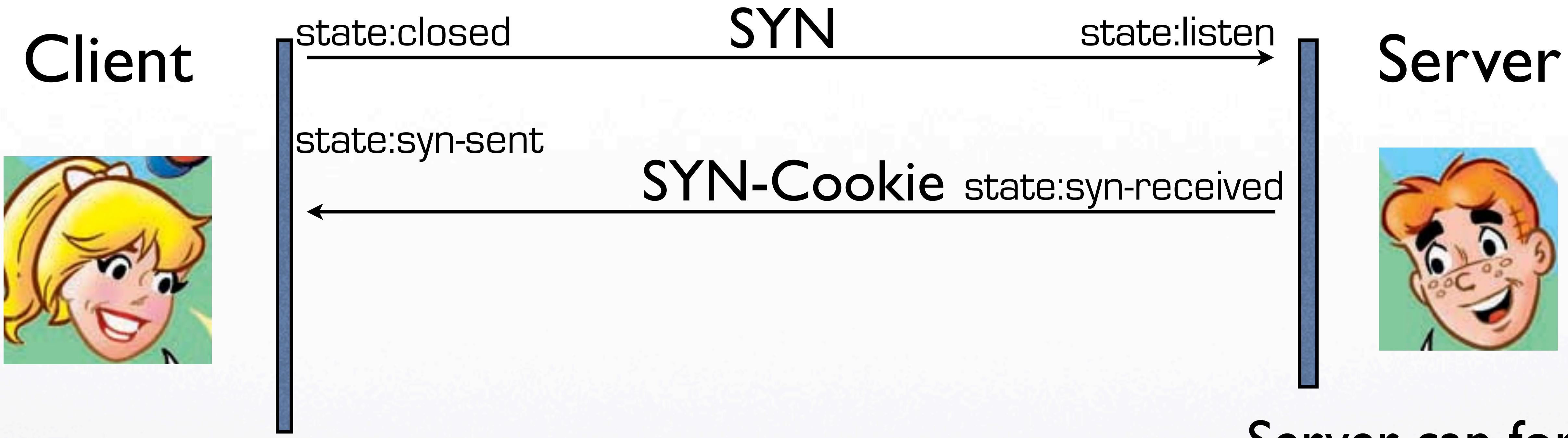
Defending against DDoS

- Harder since an attack may look as quite legitimate traffic: (e.g., HTTP)
- Challenge: distinguish between good traffic and DDoS traffic.
- We will examine some general approaches for DDoS defense next.

Simple Measures

- Try to use various parameters to make a system understand it is being attacked:
  - counting number of half-open connections (syn-rcvd).
  - counting number of connections refused.
- Once some red-flags are observed the system may shorten the time that it keeps half-open connections active.

SYN Cookies



Client

state:closed

SYN

state:listen

Server

state:syn-sent

SYN-Cookie

state:syn-received

A SYN-Cookie contains info derived by the source-address port, destination address and port etc.

Server can forget everything about the client till the ACK message that will include the SYN-Cookie

implemented in BSD

Proofs of Work

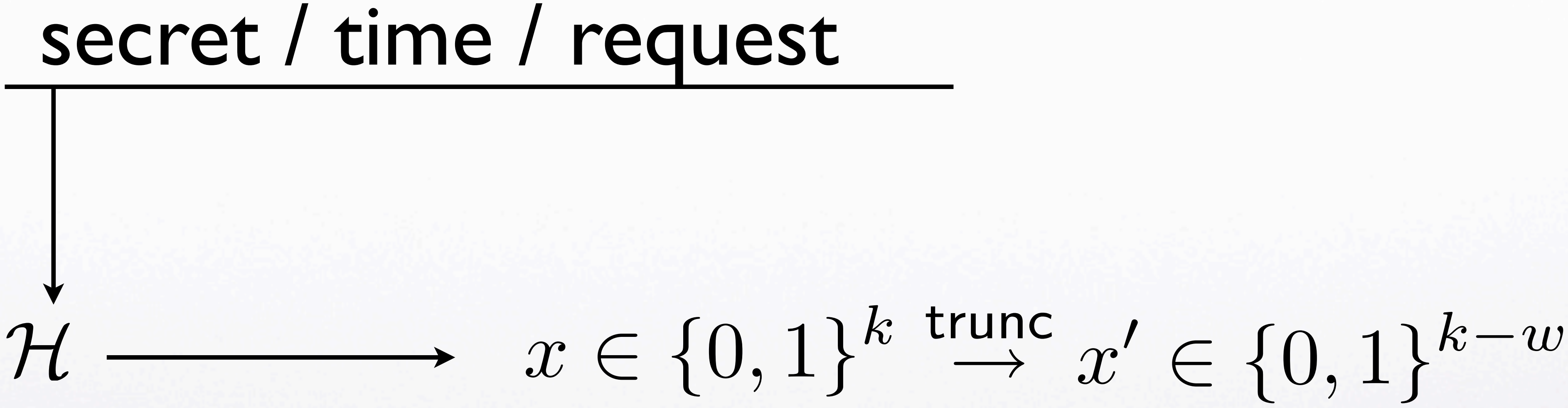
- Client is requested to solve a puzzle in order to establish a connection.
  - Pros: works in any client-server system. can be tuned to slow down malicious systems.
  - Cons: needs changes in the basic infrastructure (e.g., changes in both client and server). Will not eliminate bandwidth emaciation.

Example.

From Juels/Brainard (RSA)

Hash function

$$\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^k$$



secret / time / request

$$\mathcal{H} \longrightarrow x \in \{0,1\}^k \overset{\text{trunc}}{\rightarrow} x' \in \{0,1\}^{k-w}$$

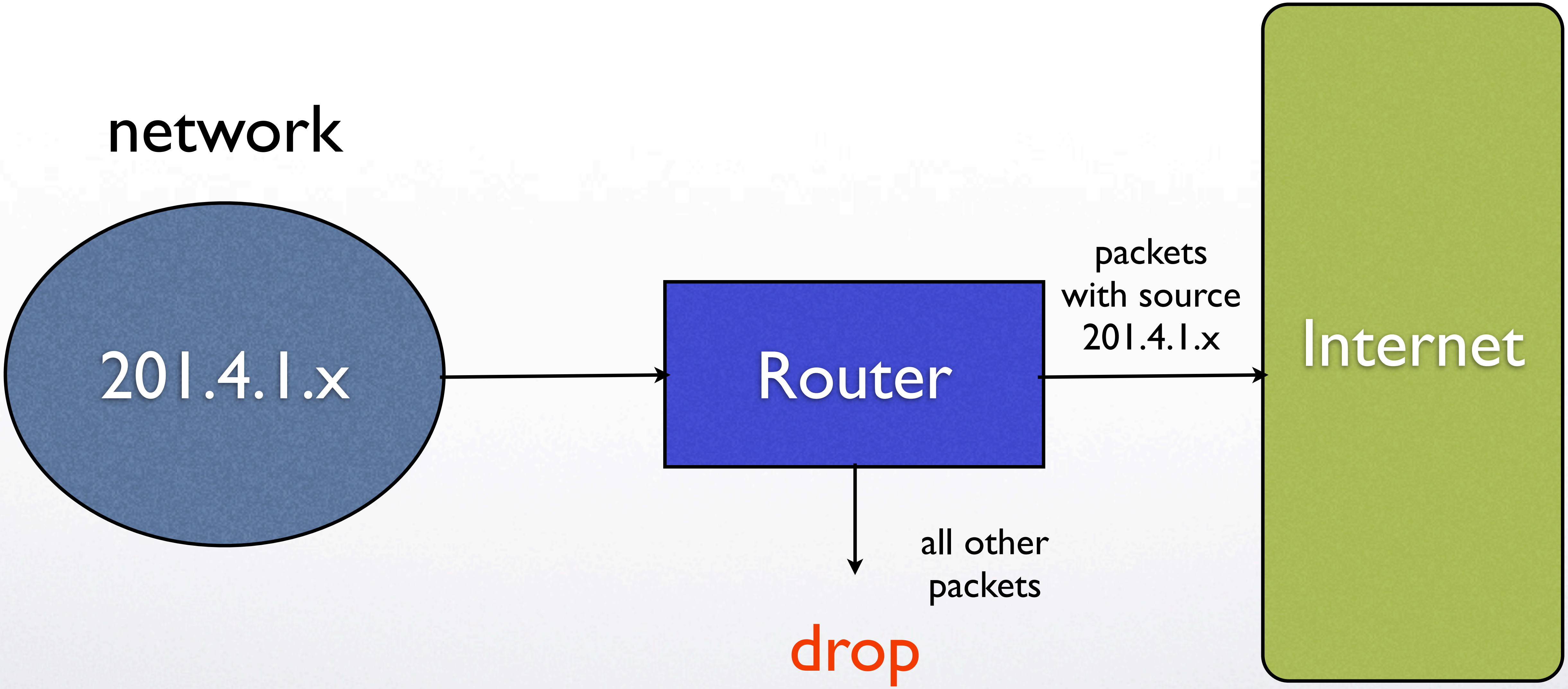
Server to client: sends $x', y = \mathcal{H}(x)$ stores (x, y)

Client needs to respond with $x'' : \mathcal{H}(x' || x'') = y$

IP Throttling

- Any IP address that issues a big upstream will have its incoming packets being dropped at a certain rate.
  - Advantage: can be configured in your router. No need to modify client/server.
  - Disadvantage: what are the right settings?

Ingress Filtering



network

201.4.1.x

Router

packets with source 201.4.1.x

Internet

all other packets

drop

http://www.faqs.org/rfcs/rfc2827.html

IP Traceback

- A “forensic” technique:
  - try to identify the ‘real’ source of a packet.
  - Have routers mark packets with their IP address. => problem: not enough space in a packet.

Edge Marking.

- With some low probability *q* a router marks a packet. Marking:
  - If packet unmarked: enter your IP address as the start IP address.
- If distance = 0: enter your IP address as the end IP address.
- Otherwise increase the distance.

Encoding IP addresses

- Use only the 16 bits of the IP identification field used for fragmentation.
- How to pack 64+5 bit information into 16 bits?

$$\mathcal{H}_1, \mathcal{H}_2 : \{0,1\}^{32} \rightarrow \{0,1\}^{11}$$

- Use

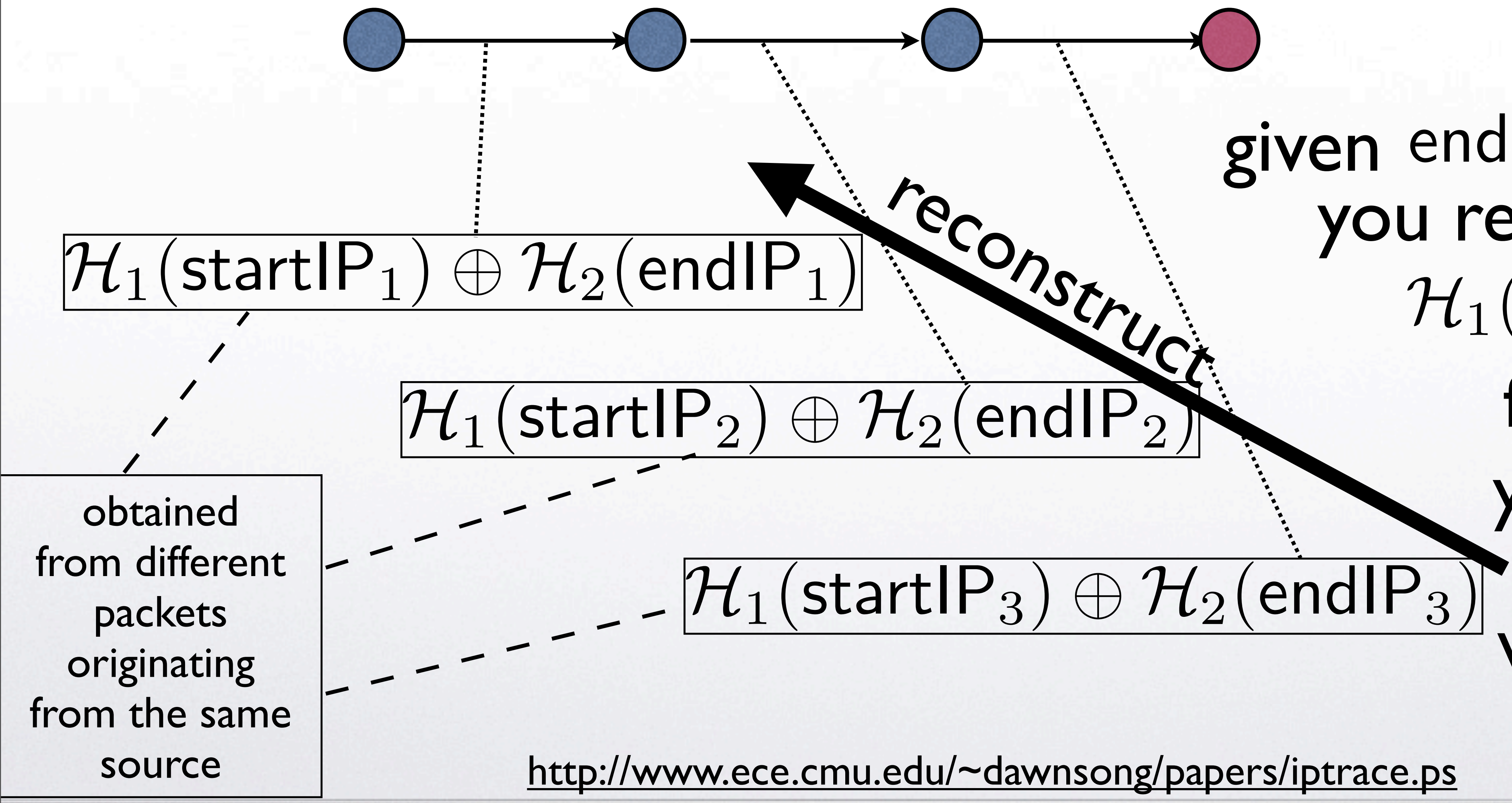
first router

$\mathcal{H}_1(\text{startIP})$

second router

$\mathcal{H}_1(\text{startIP}) \oplus \mathcal{H}_2(\text{endIP})$

Reconstruct Path



$\mathcal{H}_1(\text{startIP}_1) \oplus \mathcal{H}_2(\text{endIP}_1)$

$\mathcal{H}_1(\text{startIP}_2) \oplus \mathcal{H}_2(\text{endIP}_2)$

$\mathcal{H}_1(\text{startIP}_3) \oplus \mathcal{H}_2(\text{endIP}_3)$

obtained from different packets originating from the same source

given endIP_3 you recover $\mathcal{H}_1(\text{startIP}_3)$

from which you recover startIP_3 which equals endIP_2

etc.

http://www.ece.cmu.edu/~dawnsong/papers/iptrace.ps