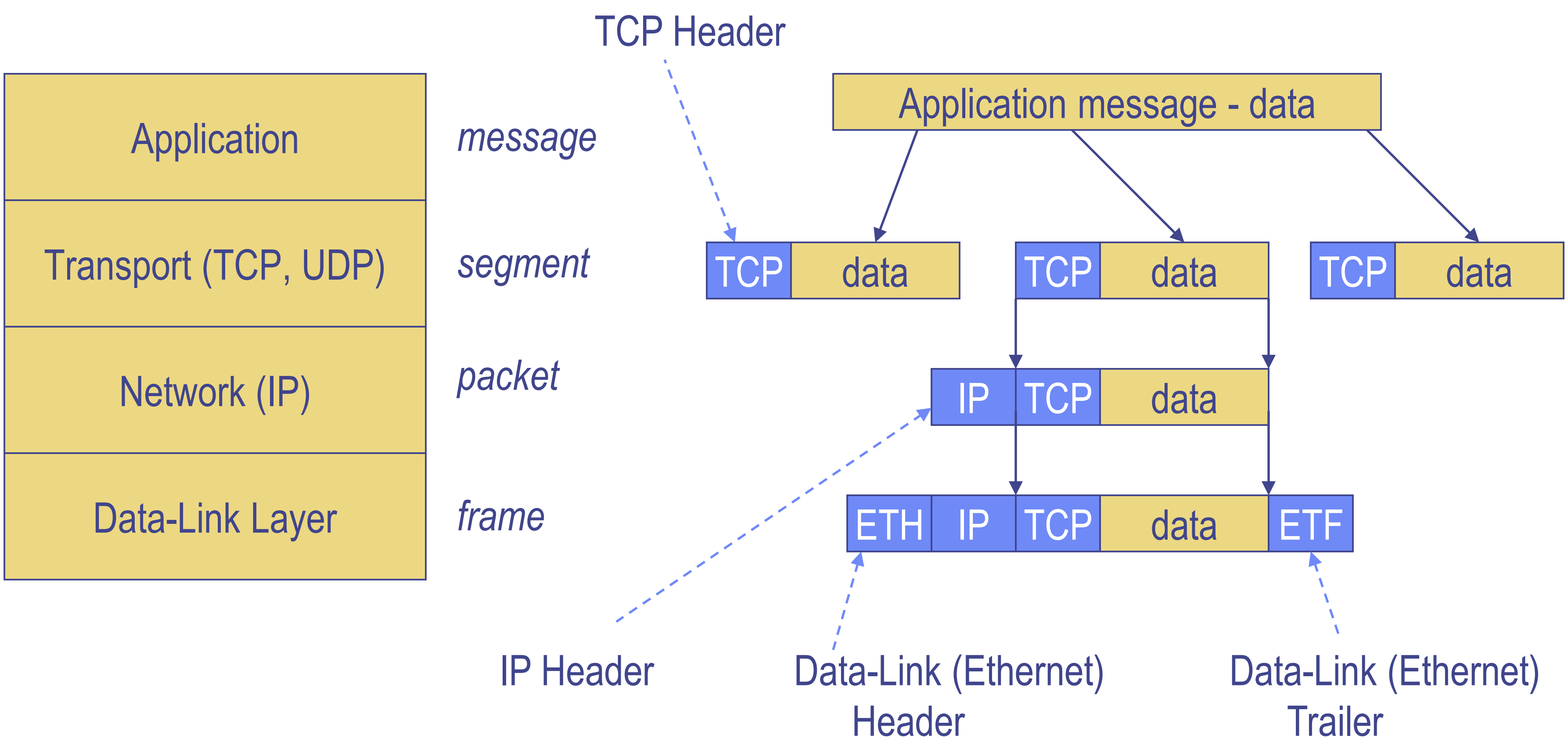
Vulnerabilities in Network Protocols

- Weaknesses in network protocols
  - Internet Protocol (IP)
  - Transmission Control Protocol (TCP)
  - User Datagram Protocol (UDP)
  - Internet Control Message Protocol (ICMP)

- Packet Sniffers

TCP/IP protocol stack



1. Weaknesses in IP

(Internet Protocol)

Internet Protocol

IP packet

- IP packets
  - Encapsulate TCP or UDP segments
  - Encapsulated into frames (data-link layer)
- Connectionless
  - Each packet is transported independently from other packets

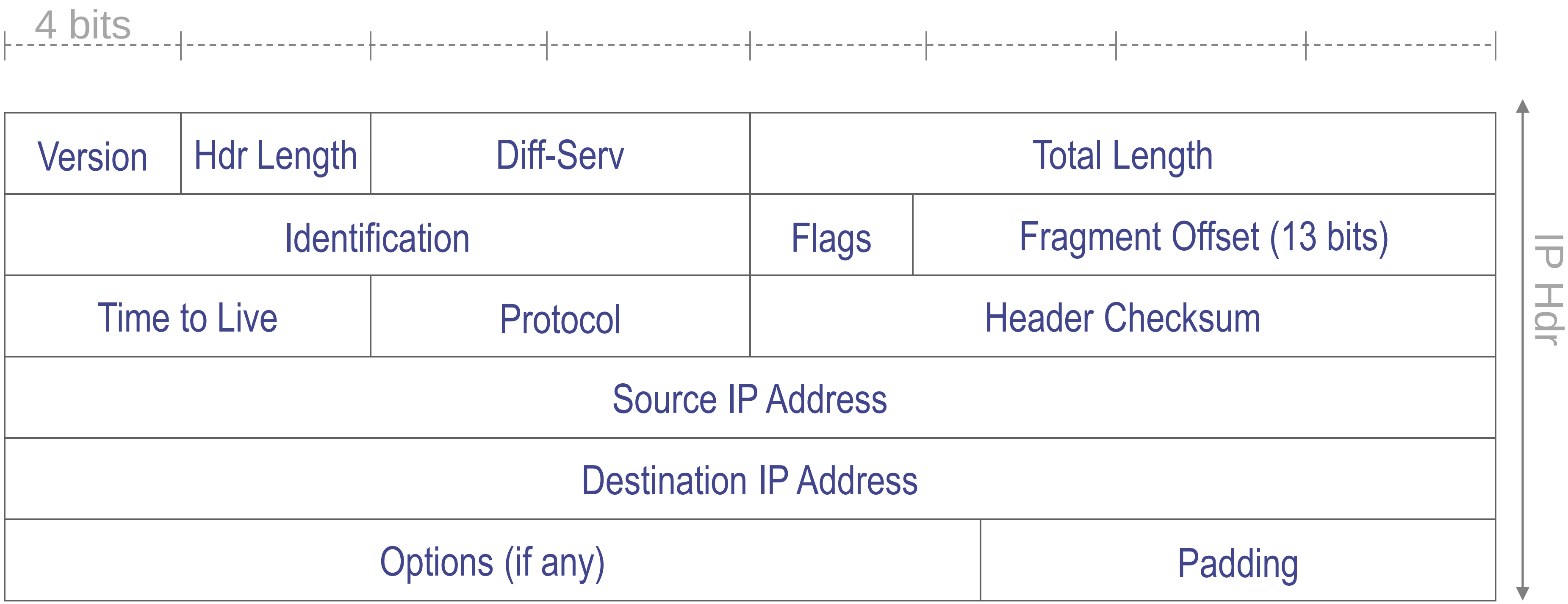
Internet Protocol

IP packet

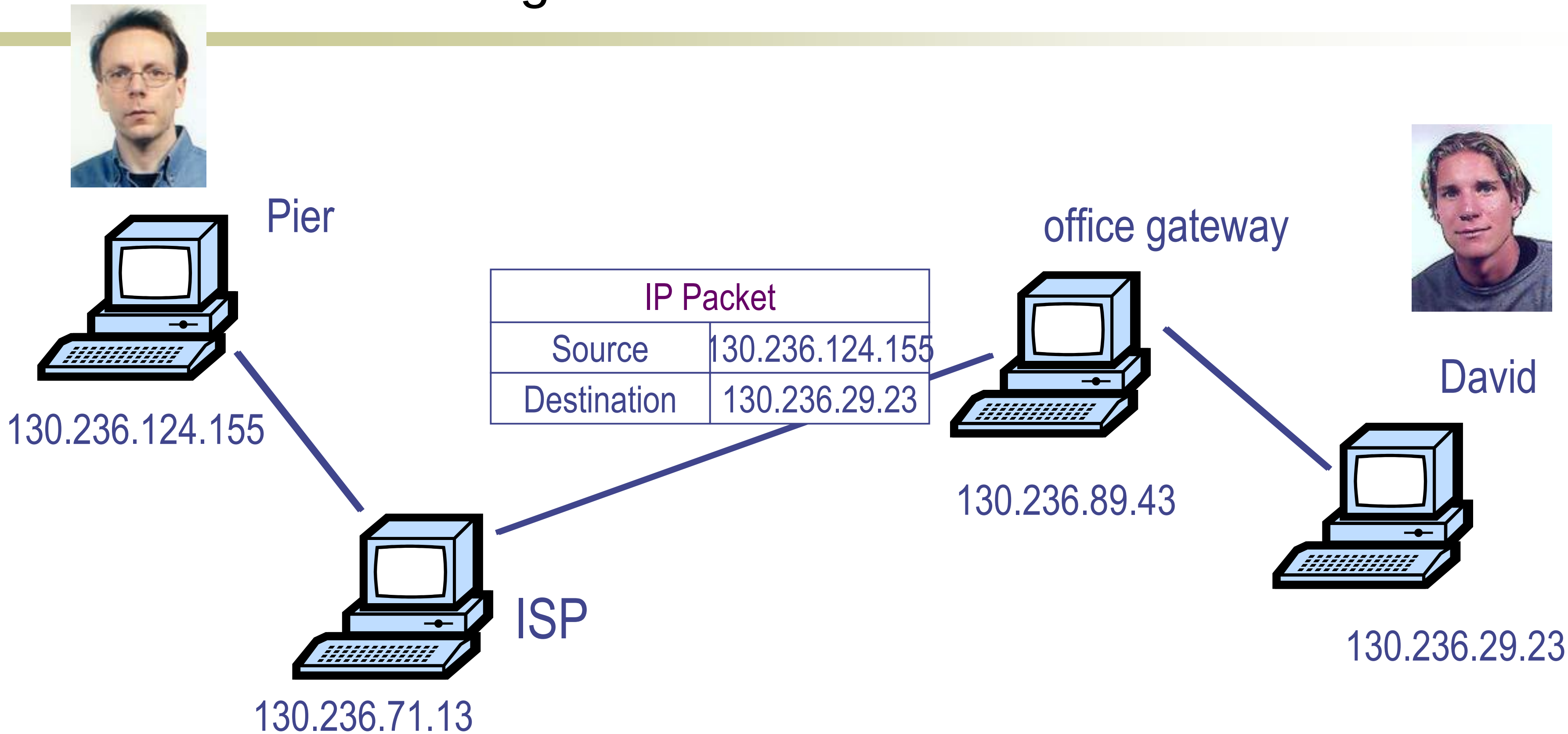
- Unreliable
  - Delivery on a best effort basis
  - No acknowledgments
  - Packets may be lost, reordered, corrupted, or duplicated
  - Checks for errors, but does not correct them

    If the checksum of an incoming packet is not correct, then the packet is simply dropped

IP packet (IP Version 4)



IP routing



- Internet routing uses numeric IP addresses
- Typical route uses several hops

IP protocol functions

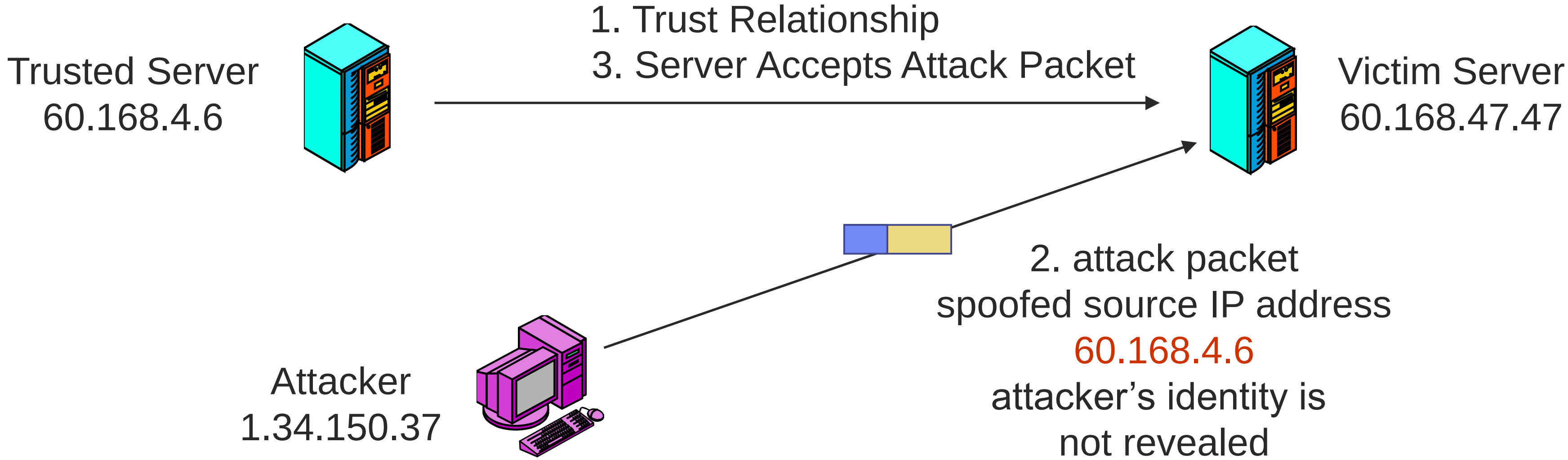
- Routing
  - IP host knows location of router (gateway)
  - IP gateway must know route to other networks
- Fragmentation and reassembly
  - If max-packet-size less than the user-data-size
- Error reporting
  - ICMP packet to source if packet is dropped
- TTL field: decremented after every hop
  - Packet dropped if TTL=0 (prevents infinite loops)

IP vulnerabilities

- Unencrypted transmission
  - Eavesdropping possible at any intermediate host during routing
- No source IP authentication
  - Sender can spoof src IP address (next slide)
- No integrity checking
  - Entire packet, header and payload, can be modified while in transit, enabling content forgeries, redirections, and man-in-the-middle attacks
- No bandwidth constraints
  - Large number of packets can be injected into network to launch a DoS attack
  - Broadcast addresses provide additional leverage

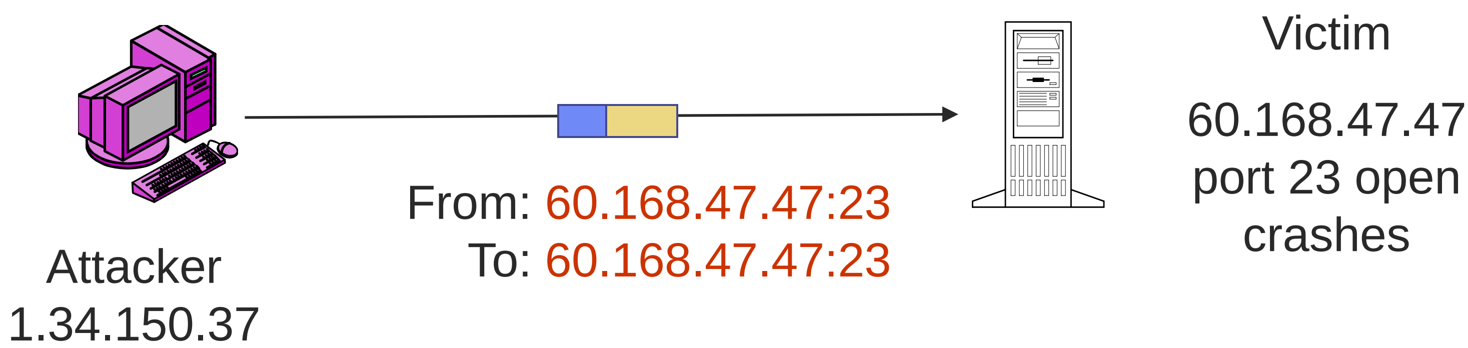
IP address spoofing attack

- Sending a message with a false IP address
  - Gives sender anonymity so that attacker cannot be identified
  - Can exploit trust between hosts if spoofed IP address is that of a host the victim host trusts
  - Don't rely on IP addresses for security
- Implication: DoS attack



History: LAND attack

- In 1997, many computers, switches, routers, and even printers, crashed when they received such a packet
  - Unexpected combination of parameters triggered a bug in many implementations



- Send a packet with
  - victim’s IP address in both source and destination address fields and
  - the same port number for the source and destination

Protocol field

- Identifies type of message encapsulated in the Data Field
  - 1=ICMP, 6=TCP, 17=UDP, etc.
- Firewalls need this information to know how to process the packet (lecture on Firewalls)

Time-to-Live field

- Each router decrements the TTL value by one to prevent infinite loops
- Router decrementing TTL field to zero discards the packet. Router also sends an error message to the sender
  - The packet containing this message reveals the sender's IP address to the attacker
- Traceroute uses TTL to map the route to a host
  - Tracert on Windows machines (Lec4)
- Firewalls can protect routers by dropping error messages

Options field

- Options values can be dangerous:

  It is possible to specify which routers to go through

  It is relatively common to drop all packets with options in firewalls both in incoming and outgoing packets

- With no options, the value of Header Length is 5

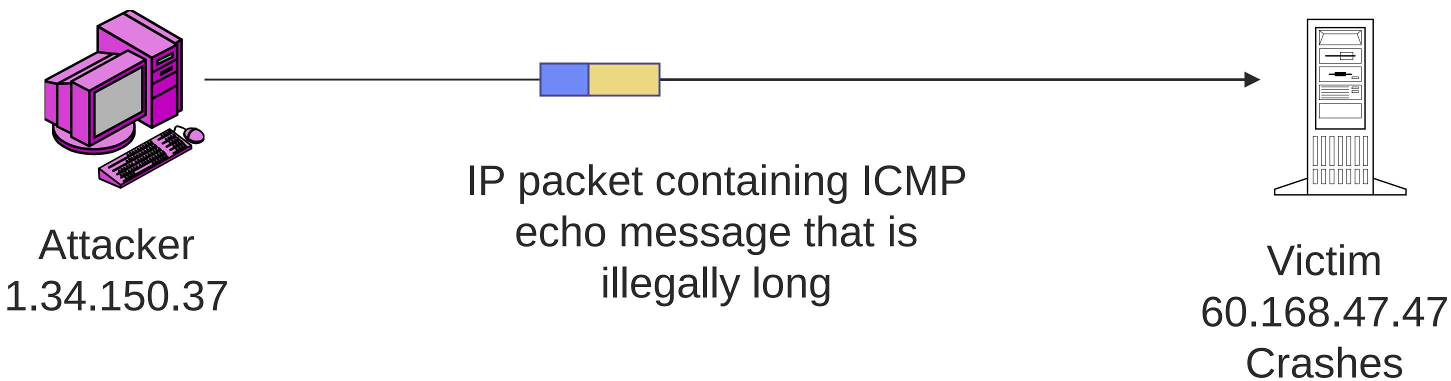
  4 bytes (32 bits) x 5 $\rightarrow$ 20 bytes

  If Header Length is more than 5, be suspicious

Total-Length field

- Field of size 2 bytes
- Gives max length of entire packet: 65,536 bytes (2^{16})
- History: Ping-of-Death attack (late 90's)

  Sends a ping (ICMP echo) packet with length greater than 65,536 bytes

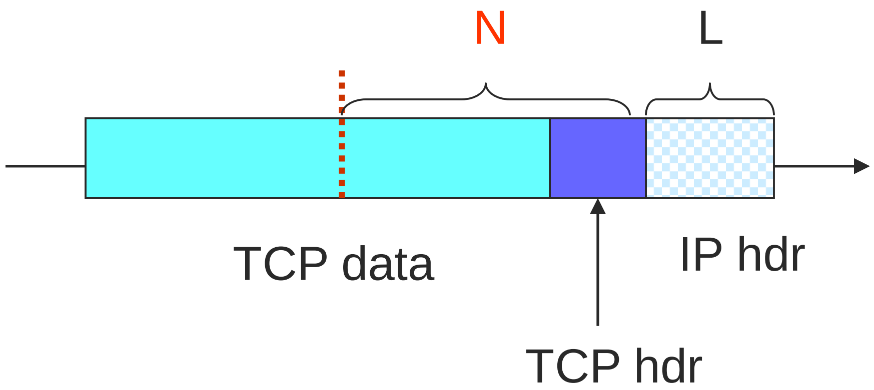


  - Early systems crashed
  - Many systems didn't know what to do with these packets!
  - Current systems drop such packets

IP fragmentation

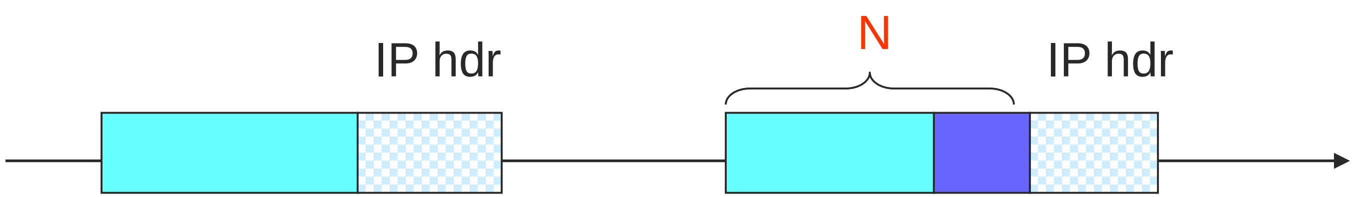
- Different networks may have different maximum packet length
- Routers must fragment IP packets
  - All fragments have the same Identification value
  - Fragment Offset values allows fragments to be ordered
    - The Offset field of the segment is set based on the offset of the segment in the original message
    - Offset field measured in unit of 8 bytes blocks
  - More Fragments bit is 0 in the last fragment

IP fragmentation



IP packet

- max packet length M
- identification number = id
- L = length of IP hdr
- M = N + L

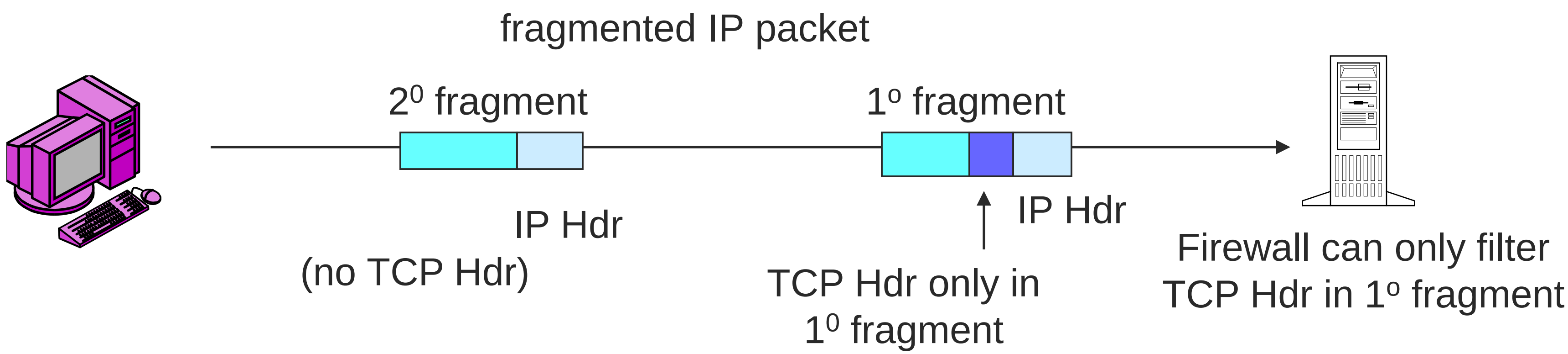


- Identification number = id
- Offset = N
- More Fragments = 0

- Identification number = id
- Offset = 0
- More Fragments = 1

IP fragmentation

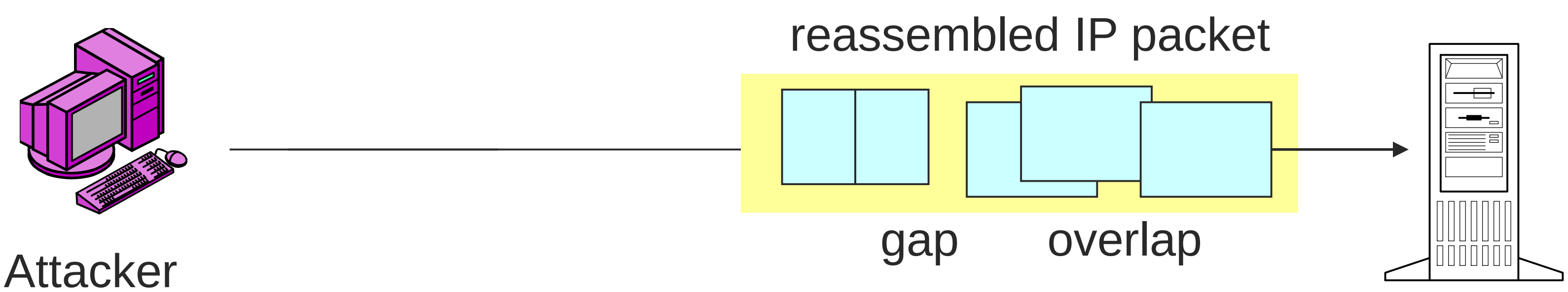
- Harms packet inspection



- TCP, UDP or ICMP header is present only in the first packet of the series
- The header is filtered in the first packet by firewalls
- Firewalls may drop the first packet because its header has dangerous content
  - Subsequent fragments in the series cannot be dropped (no headers)
  - Firewalls need to save information to drop all fragments!
- Fragmentation is rare today, normally all fragmented packets are dropped

History: Teardrop attack

- Fragmented IP packets that when reassembled do not make sense: gaps and overlaps
- Some operating systems crashed (late 90’s)
  - When Windows NT receives these invalid packets, it allocates memory. If enough of invalid packets are received Windows NT may hang with a STOP



History: Teardrop attack

- Crafted fragmented packet does not make sense when reassembled

- Identification number = id
- Offset < N
- More Fragments = 0

- Identification number = id
- Offset = 0
- More Fragments = 1
- Packet length = N + L

When re-assembled gives an overlap!

2. Weaknesses in TCP

(Transmission Control Protocol)

TCP

- Works at the Transport layer
- Popular application protocols built on top of TCP:
  - WWW, FTP and SSH
- Guarantees:
  - Reliable data transfer
  - Preserves delivery order of packets
  - Distinguishes data for distinct applications on the same host

TCP

- Connection-oriented, preserves order
  - Sender
    - Breaks data into packets - rely on IP to transmit them
    - Attaches sequence numbers to packets
  - Receiver
    - Acknowledges receipt - lost packets are re-sent
    - Reassembles packets in correct order
    - Checks data transmitted by comparing a checksum of the data with the checksum encoded in the packet

TCP segment

TCP | data

4 bits

<table>
<tr><td colspan="3">Source Port Number</td><td>Destination Port Number</td></tr>
<tr><td colspan="4">Sequence Number</td></tr>
<tr><td colspan="4">Acknowledgment Number</td></tr>
<tr><td>Hdr Length</td><td>Reserved</td><td>Flag Fields</td><td>Window Size</td></tr>
<tr><td colspan="3">TCP Checksum</td><td>Urgent Pointer</td></tr>
</table>

TCP Hdr

Ports

- A port is a number (0-65535) used to map data to a process

  Port numbers identify applications
- Ports are divided in three ranges:
  - Well-known ports (0-1023) used by major applications

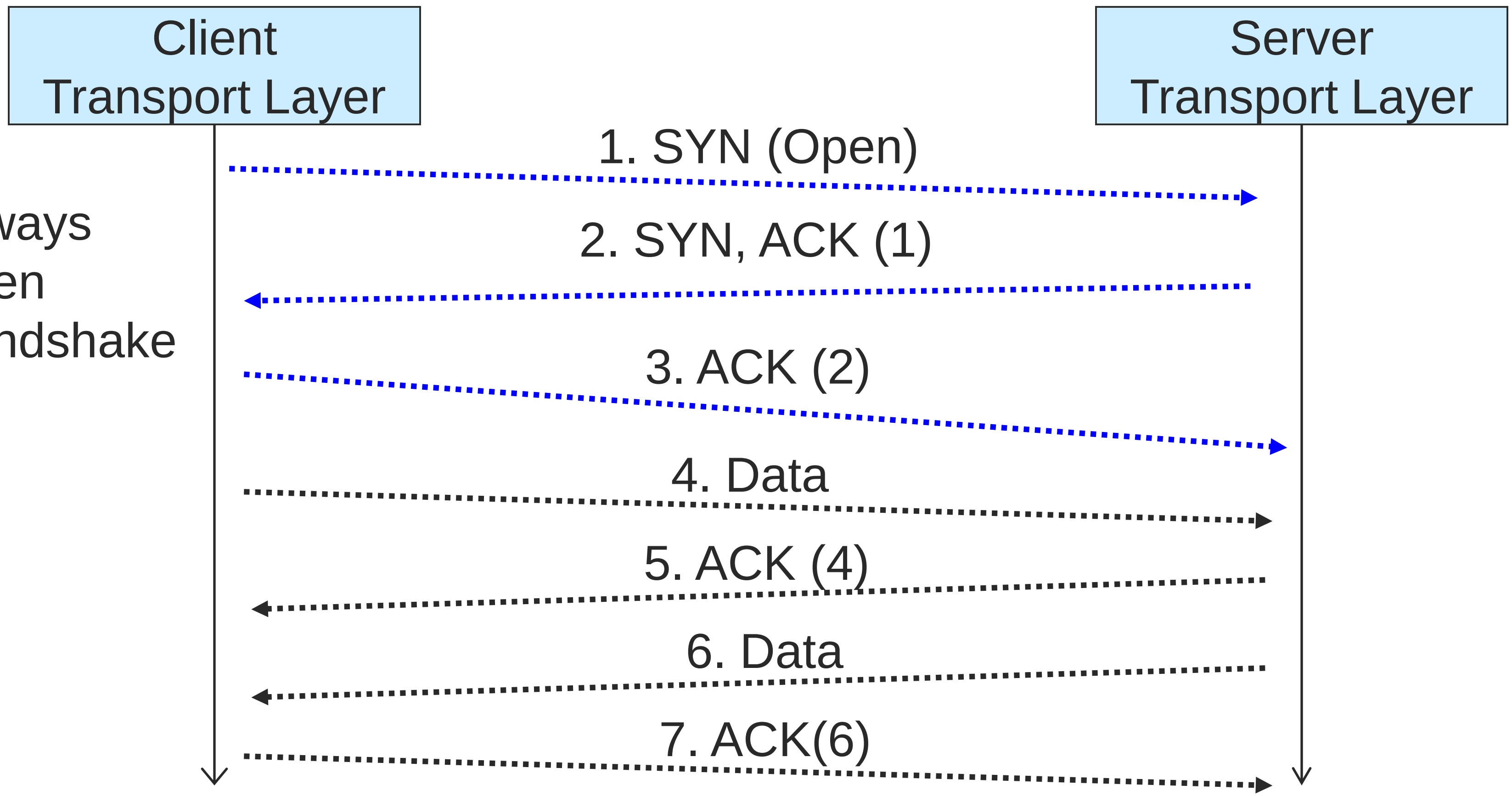
    example: HTTP=80, Telnet=23, FTP=21, SMTP=25
  - Registered ports (1024-49151) for any application
  - Ephemeral/dynamic/private ports (49152-65535) used by clients

Sockets

- A <u>socket</u> is a communication end-point
  - unique to every machine connected to a network
- A socket consists of an IP address and a port number
  - Designates a specific program on a specific machine
  - **128.171.17.13:80**

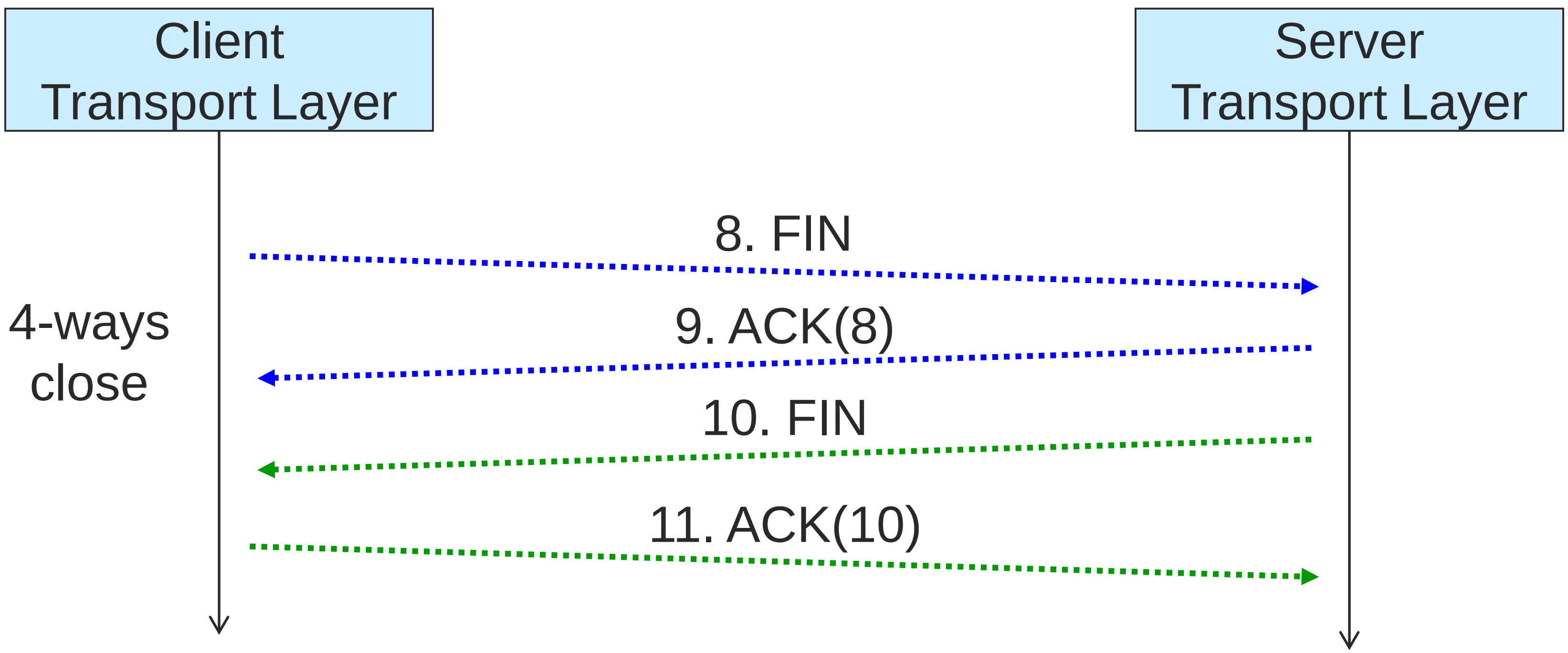
Opening a TCP session

3-ways
open
handshake

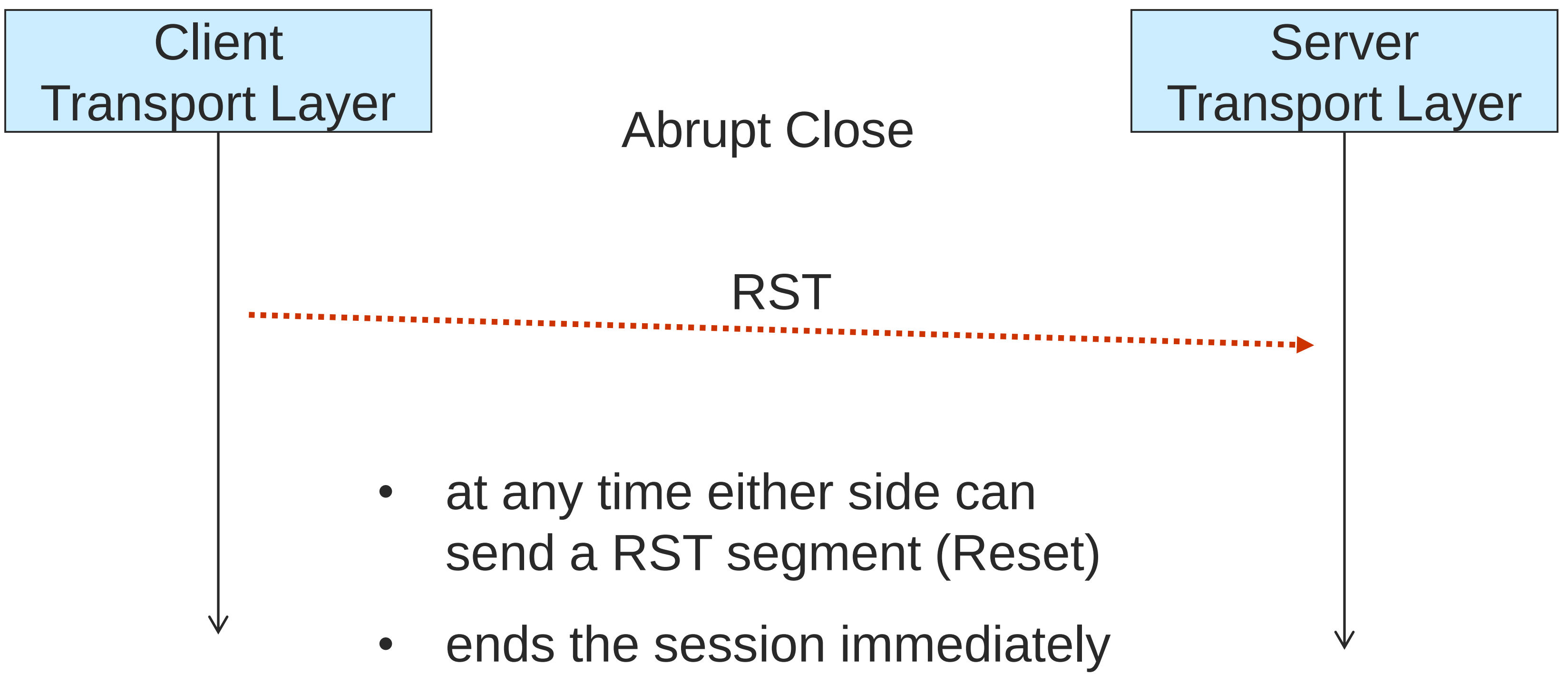


Closing a TCP session

- Normally, FIN is used in a 4-way close



Closing a TCP session



- at any time either side can send a RST segment (Reset)
- ends the session immediately

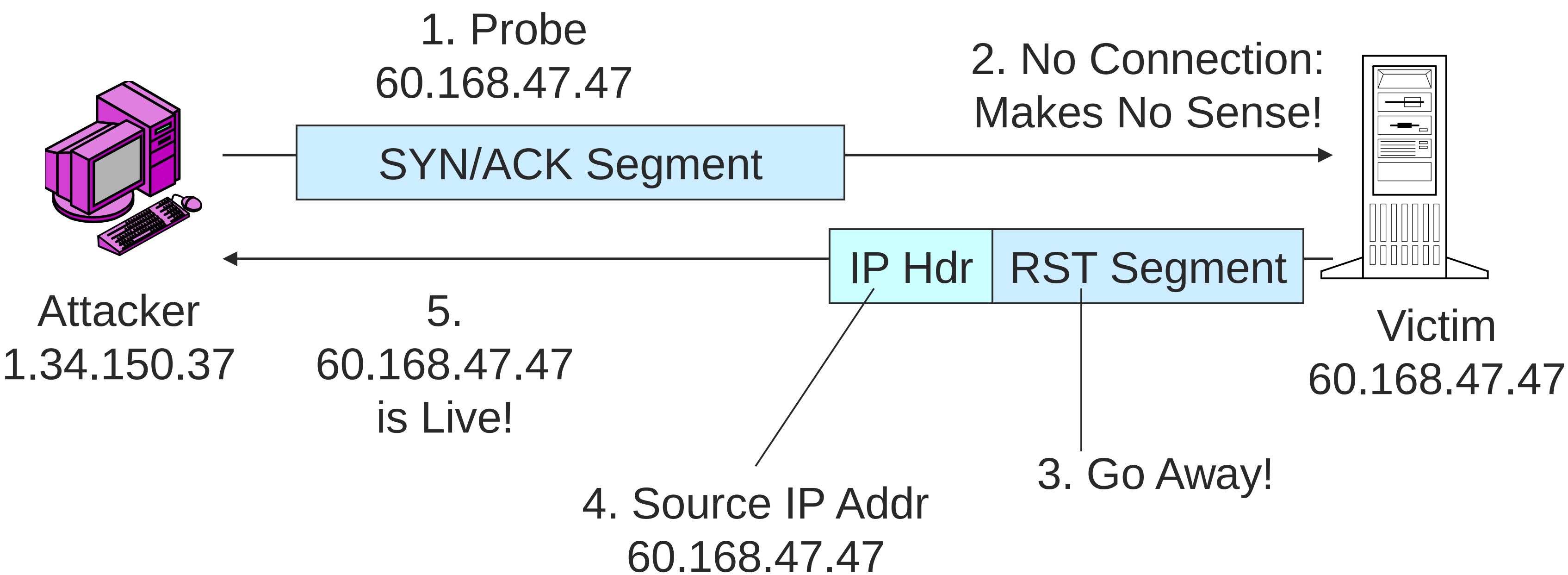
TCP vulnerabilities

- Network packets pass by untrusted hosts
  - Eavesdropping, packet sniffing
  - Especially easy when attacker controls a machine close to the victim
- TCP state can be guessed
  - Enables spoofing and session hijacking
- Denial of Service (DoS) vulnerabilities (next lecture)

TCP vulnerabilities

- RST can create a single-message close
  - Attackers can try to generate RSTs

SYN/ACK scanning attack



May work through a firewall since SYN/ACK is the reply of a connection established from the inside

TCP Connection Spoofing attack

- Attackers inject packets into an existing TCP connection
- TCP sequence numbers may prevent these kind of threats
  - TCP sequence numbers selected on random when TCP connection starts
  - Attackers watching network traffic know sequence numbers
  - Other attackers may be able to guess or send a large amount of RST messages to a host

3. Weaknesses in UDP

(User Datagram Protocol)

UDP

- Works at the Transport layer
- Connectionless protocol
  - Send UDP segments to the application at the specified port of the IP address
  - Significantly fast
- UDP is <u>not</u> a reliable protocol
  - No acknowledgment
  - No congestion control
  - No message continuation
- Applications: media streaming, broadcast

UDP segment

UDP data

4 bits

Source Port Number	Destination Port Number
UDP Length	UDP Checksum

UDP Hdr

UDP vulnerabilities

- <u>Port Spoofing</u>
  - Incorrect application uses a well-known port
  - Example: port 80 (HTTP) which is allowed through firewalls

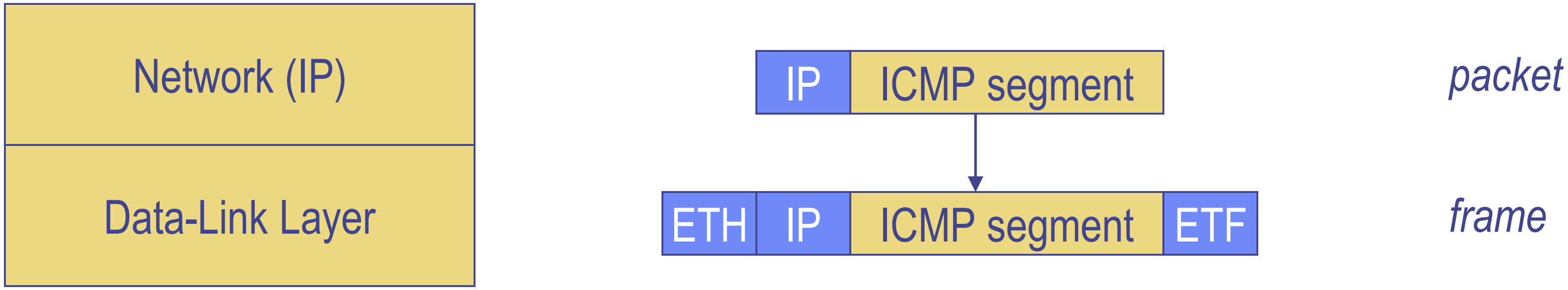
- <u>UDP segment insertion</u>
  - Insert UDP segments into an ongoing dialog stream
  - Hard to detect because no sequence numbers in UDP
  - This requires protection at the application level
    - Firewalls may not know application protocol
    - Application protocol may not even detect inserted datagrams

4. Weaknesses in ICMP

(Internet Control Message Protocol)

Internet Control Message Protocol

- Works at the Network layer
- Used for network testing and debugging
    - Tools based on ICMP: Ping and Traceroute
- ICMP segments encapsulated into IP packets



ICMP segment

ICMP segment

4 bits

Type	Code	Depends on Type and Code
Depends on Type and Code		

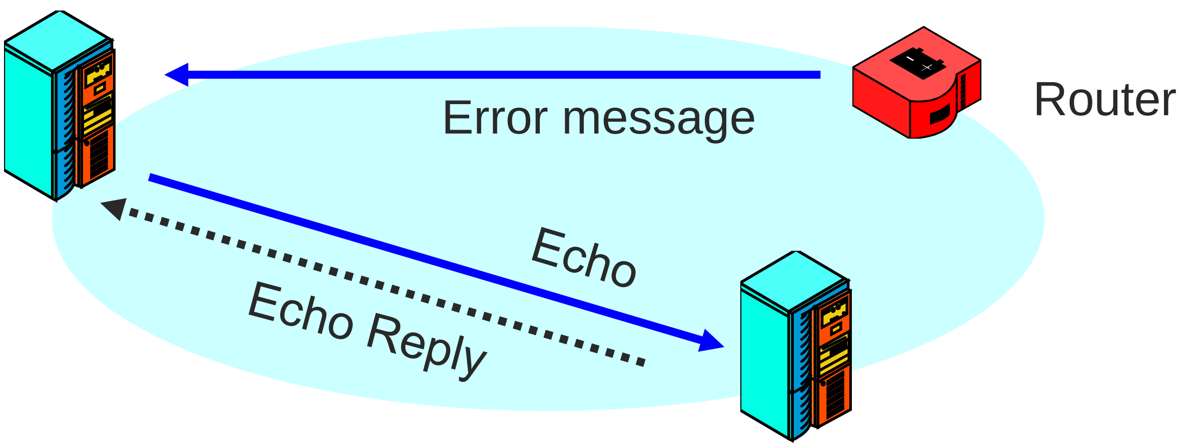
- Type field: category of supervisory message
- Code field: subcategory of type (set to zero if there is no code)
- To see an example of ICMP segment:

  1. run Wireshark
  2. open a DOS window
  3. execute **`c:\> ping liu.se`**

ICMP error messages

- ICMP messages can be used to:
  - see if a host is reachable
  - change how a host operates

    like to slow down its transmission rate
  - tell a router to send all traffic for a particular network to another network



ICMP error messages

- Inform sender of errors but there is no error correction
- Host unreachable (Type 3, multiple codes)
  - Many codes for specific reasons for host being unreachable
  - Host unreachable packet's source IP address confirms to attacker that a router is alive and becomes a potential victim
- Same with port unreachable messages
  - Networks can be mapped this way
- Tracert command can be used both for debugging and for mapping a network
  - Returned from routers when TTL=0

Conclusions

- Protocols should be simple
  - Easy to verify by firewalls
  - Easy to write robust implementations
- Core protocols not designed for security
  - Eavesdropping, packets injection, etc.
  - Applications must take care of offering the necessary level of protection needed
- More secure variants exist:
  - IP $\rightarrow$ IPsec
  - SSL/TLS

5. Packet Sniffers

Packet Sniffers

- Packet sniffers read information traversing a network
  - Packet sniffers intercept network packets
  - Can be used as legitimate tools to analyze a network
    - Monitor network usage
    - Filter network traffic
    - Analyze network problems
  - Can also be used maliciously
    - Steal information (passwords, conversations, etc.)
    - Analyze network information to prepare an attack
- Packet sniffers can be either software or hardware based
  - Sniffers are dependent on network setup

Packet Sniffers

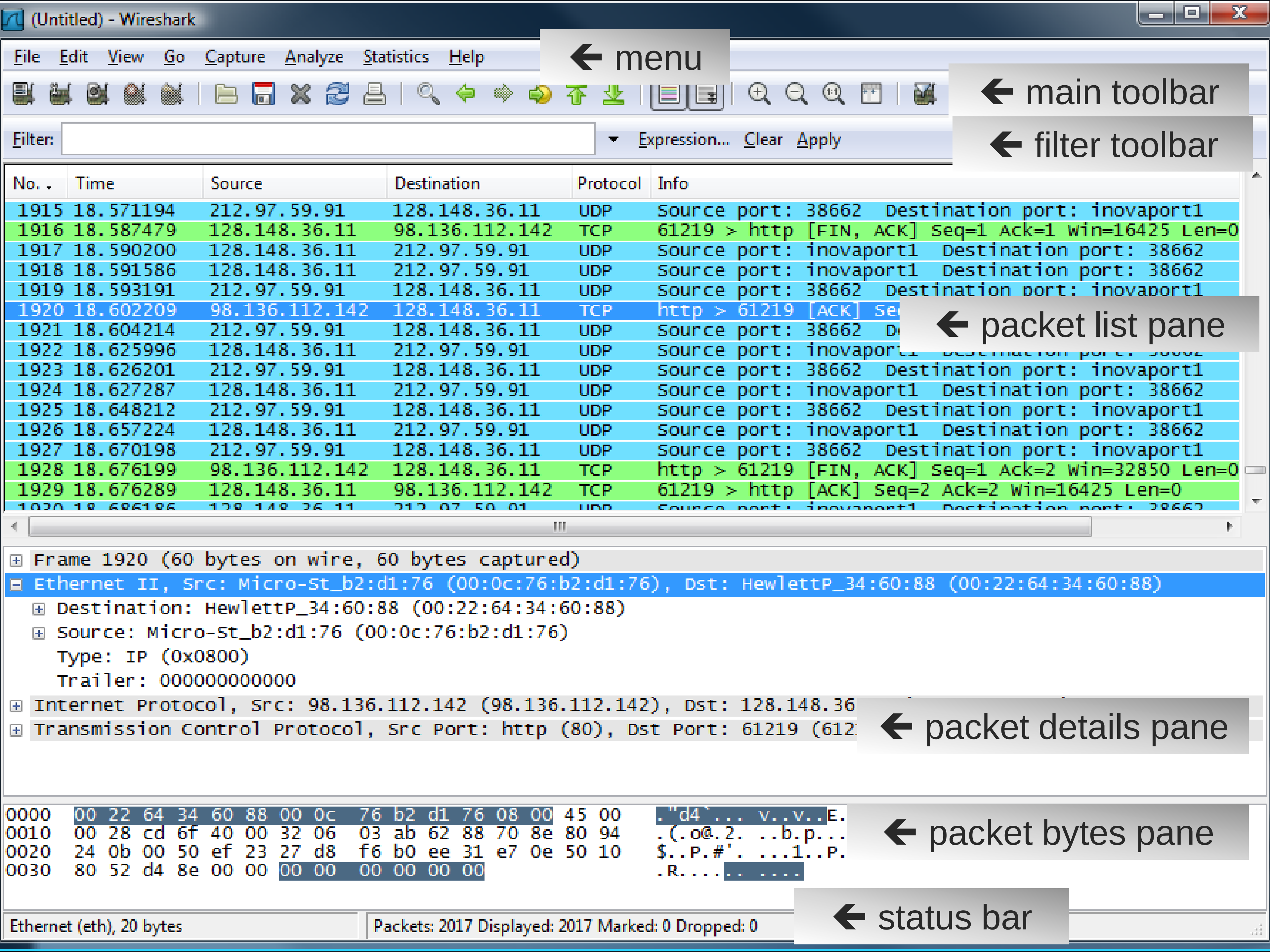
- Possible to sniff data (frames) on the same network segment
- Data travelling on the same network segment is received by every device on the segment
- Upon reception, before accepting the frame, the host in the segment will compare the frame’s destination MAC address with its own MAC address
- If the network interface (Ethernet card) of a host is operating in promiscuous mode, the host will retain every frame

Stopping Packet Sniffing

- The best way is to encrypt packets securely
  - Sniffers can capture the packets, but they are meaningless
  - SSH is also a much more secure method of connection
    - Private/Public key pairs makes sniffing virtually useless
  - On switched networks, almost all attacks will be via ARP spoofing
    - Add machines to a permanent store in the cache
    - This store cannot be modified via a broadcast reply
    - Thus, a sniffer cannot redirect an address to itself
- The best security is to not let sniffers
  - Sniffers need to be on your subnet in a switched hub
  - All sniffers need to somehow access root at some point to start themselves up

Wireshark

- Packet sniffer and protocol analyzer
  - Captures and analyzes frames
  - Supports plugins
- Usually required to run with administrator privileges
- Setting the network interface in promiscuous mode captures traffic across the entire LAN segment and not just frames addressed to the machine
- Freely available on www.wireshark.org


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Help
menu
main toolbar
Filter:
Expression... Clear Apply
filter toolbar
No. Time Source Destination Protocol Info
1915 18.571194 212.97.59.91 128.148.36.11 UDP Source port: 38662 Destination port: inovaport1
1916 18.587479 128.148.36.11 98.136.112.142 TCP 61219 > http [FIN, ACK] Seq=1 Ack=1 Win=16425 Len=0
1917 18.590200 128.148.36.11 212.97.59.91 UDP Source port: inovaport1 Destination port: 38662
1918 18.591586 128.148.36.11 212.97.59.91 UDP Source port: inovaport1 Destination port: 38662
1919 18.593191 212.97.59.91 128.148.36.11 UDP Source port: 38662 Destination port: inovaport1
1920 18.602209 98.136.112.142 128.148.36.11 TCP http > 61219 [ACK] Se
1921 18.604214 212.97.59.91 128.148.36.11 UDP Source port: 38662 D
1922 18.625996 128.148.36.11 212.97.59.91 UDP Source port: inovapor
1923 18.626201 212.97.59.91 128.148.36.11 UDP Source port: 38662 Destination port: inovaport1
1924 18.627287 128.148.36.11 212.97.59.91 UDP Source port: inovaport1 Destination port: 38662
1925 18.648212 212.97.59.91 128.148.36.11 UDP Source port: 38662 Destination port: inovaport1
1926 18.657224 128.148.36.11 212.97.59.91 UDP Source port: inovaport1 Destination port: 38662
1927 18.670198 212.97.59.91 128.148.36.11 UDP Source port: 38662 Destination port: inovaport1
1928 18.676199 98.136.112.142 128.148.36.11 TCP http > 61219 [FIN, ACK] Seq=1 Ack=2 Win=32850 Len=0
1929 18.676289 128.148.36.11 98.136.112.142 TCP 61219 > http [ACK] Seq=2 Ack=2 Win=16425 Len=0
packet list pane
Frame 1920 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: Micro-St_b2:d1:76 (00:0c:76:b2:d1:76), Dst: HewlettP_34:60:88 (00:22:64:34:60:88)
Destination: HewlettP_34:60:88 (00:22:64:34:60:88)
Source: Micro-St_b2:d1:76 (00:0c:76:b2:d1:76)
Type: IP (0x0800)
Trailer: 000000000000
Internet Protocol, Src: 98.136.112.142 (98.136.112.142), Dst: 128.148.36
Transmission Control Protocol, Src Port: http (80), Dst Port: 61219 (612
packet details pane
0000 00 22 64 34 60 88 00 0c 76 b2 d1 76 08 00 45 00 ."d4`... v..v..E.
0010 00 28 cd 6f 40 00 32 06 03 ab 62 88 70 8e 80 94 .(.o@.2. ..b.p...
0020 24 0b 00 50 ef 23 27 d8 f6 b0 ee 31 e7 0e 50 10 $..P.#'. ...1..P.
0030 80 52 d4 8e 00 00 00 00 00 00 00 00 .R...... ....
packet bytes pane
Ethernet (eth), 20 bytes
Packets: 2017 Displayed: 2017 Marked: 0 Dropped: 0
status bar