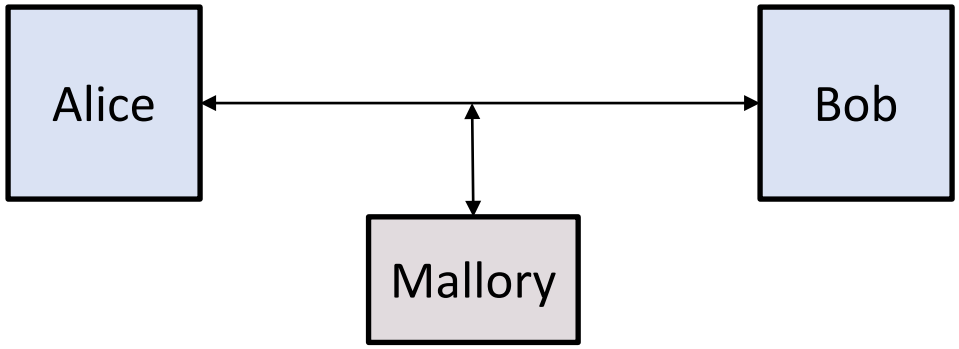
Lecture 6: Network Security

6.1 Background: Network layers
6.2 Name resolution and attacks
6.3 Denial of Service (Dos) attacks
6.4 Useful network security tools
6.5 Protection: Securing the communication channel using cryptography
6.6 Protection: Firewall
6.7 Protection: Network security management

See: [PF6.1], [PF6.2],[PF6.4],[PF6.6], [PF6.9]

A Secure Channel: *Mission Accomplished*?

- Using cryptographic techniques + PKI, we can establish a **secure channel** over an insecure underlying communication network



- *Mission accomplished?*
- There are **many other issues**:
  - Other information, in particular **networking information**, need to be protected
  - Other security **requirements**: e.g. availability (against DDoS), anonymity (Tor)
  - Network access control (e.g. firewall), monitoring, and other management **issues**
  - Web security above networking: in the next lecture

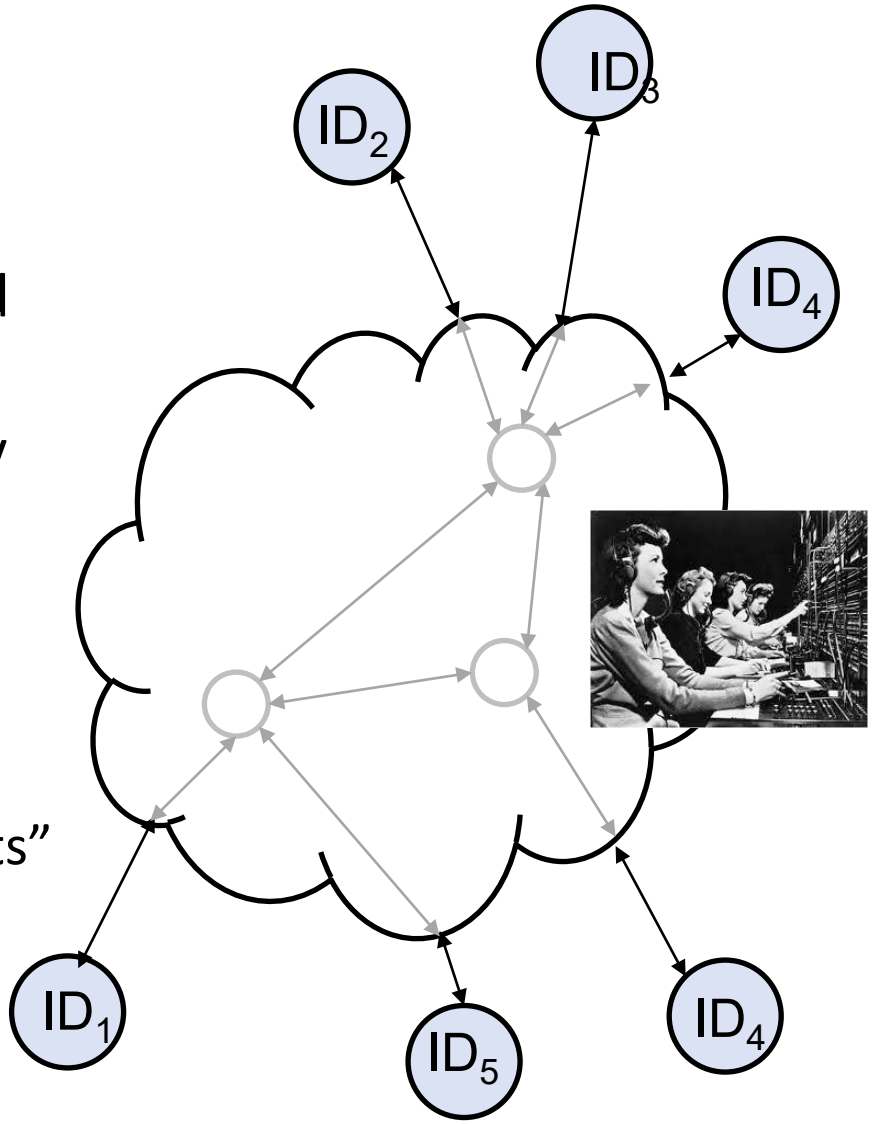
6.1 Background: Network Layers

Important notions relevant to this module:

- Network layers
- Naming
- Ports

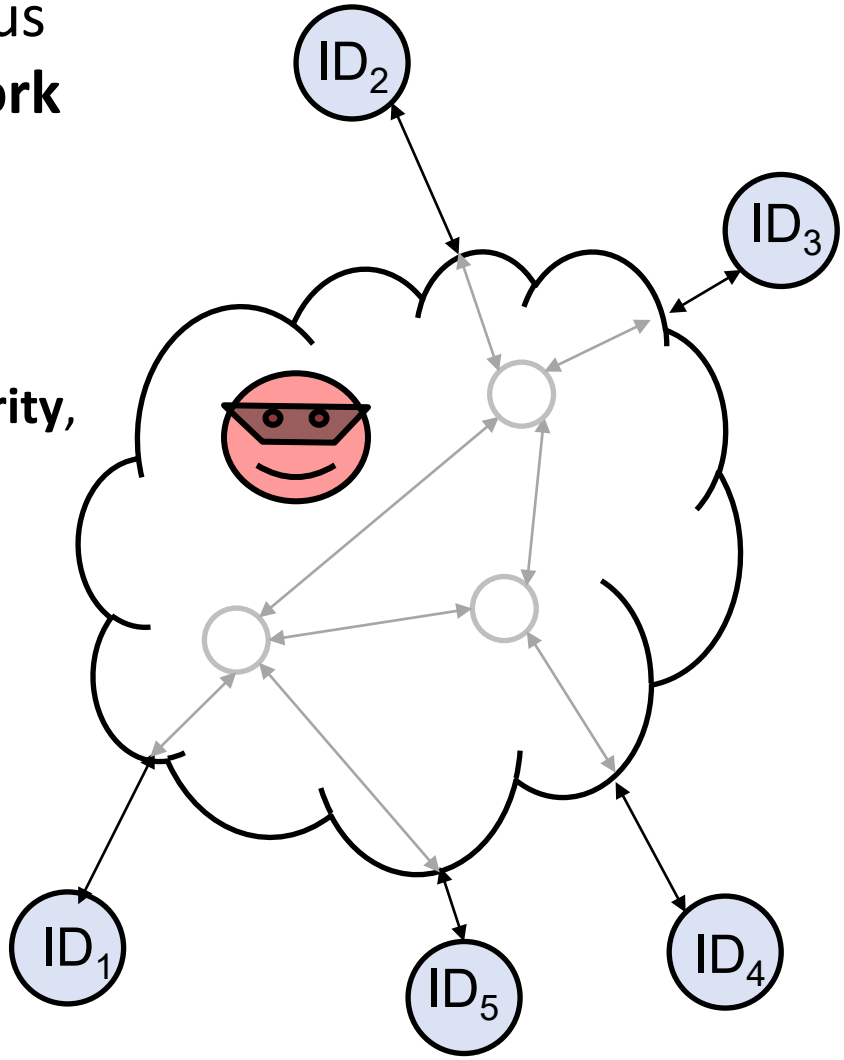
Computer Network

- Computer network allows for **communication** between entities
- To share networking resources and enhance robustness, instead of having dedicated line between any two nodes, ***packet switching*** is deployed:
  - Messages route via multiple switches and routers
  - Messages are broken into “packets”



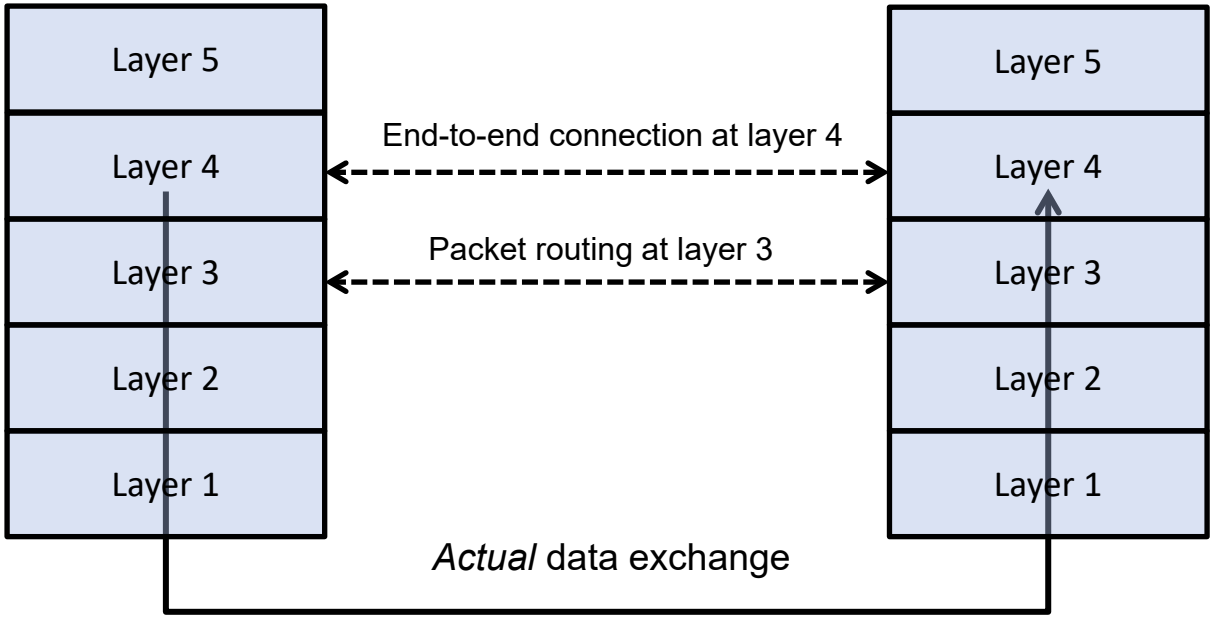
Network Security

- While networking technologies focus on how to deliver messages, **network security** focuses on the following **objectives** *under the presences of attackers*:
  1. Preserving the **confidentiality**, **integrity**, and **authenticity** of messages from communicating entities
  2. Maintaining the **availability** of the network and network services
  3. **Controlling** and **monitoring** network traffic flow



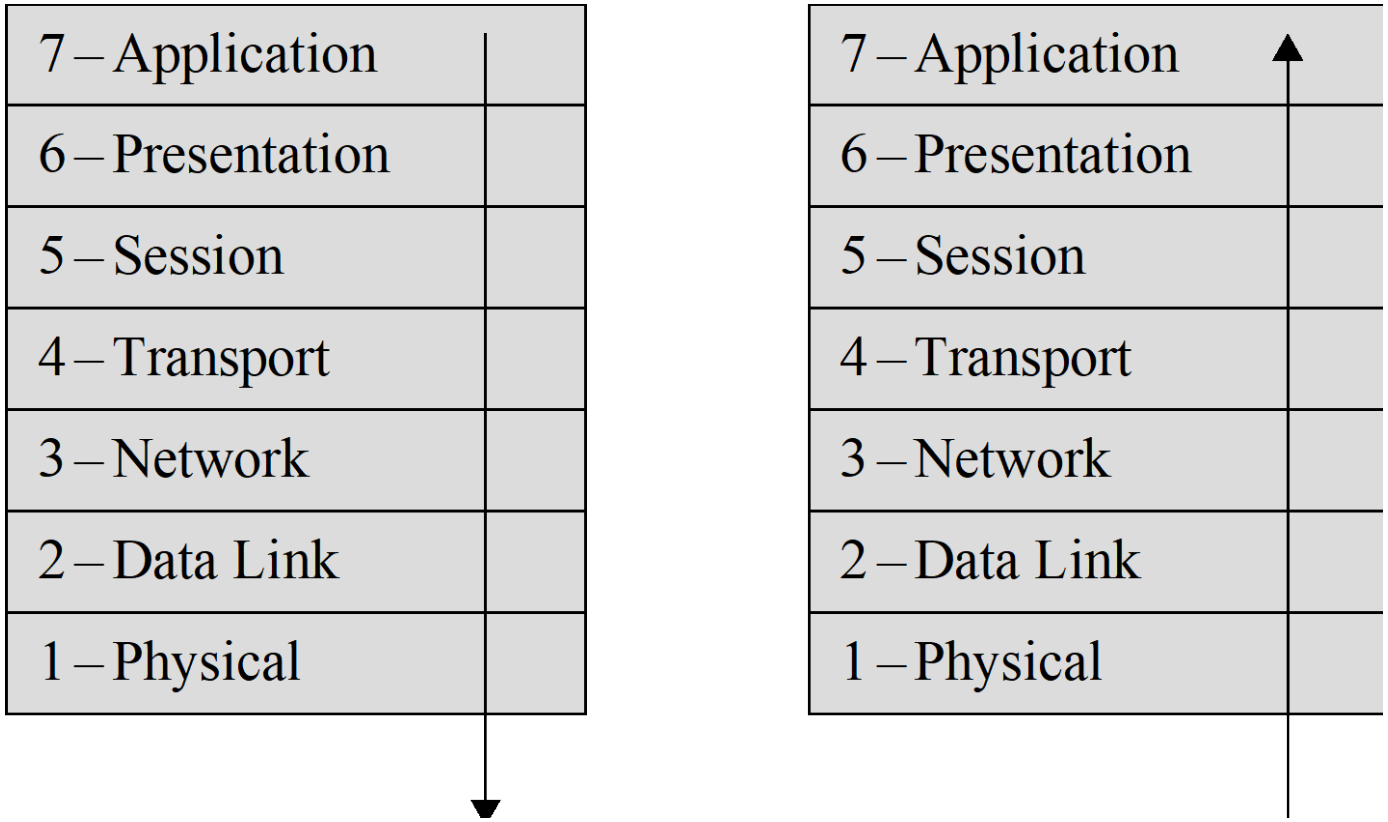
Network Layering

- A networking protocol consists of several ***network layers***
- This partitions a complex communication system into several **abstraction layers**:
  - The **peer entities** at the **same** layer *N* "conceptually" communicate with each other by executing a protocol *at that layer*
  - Layer *N*-1 provides services to entities in layer *N*: the **layer-*N*** protocol is built **on top of** a virtual connection at layer ***N*-1** below

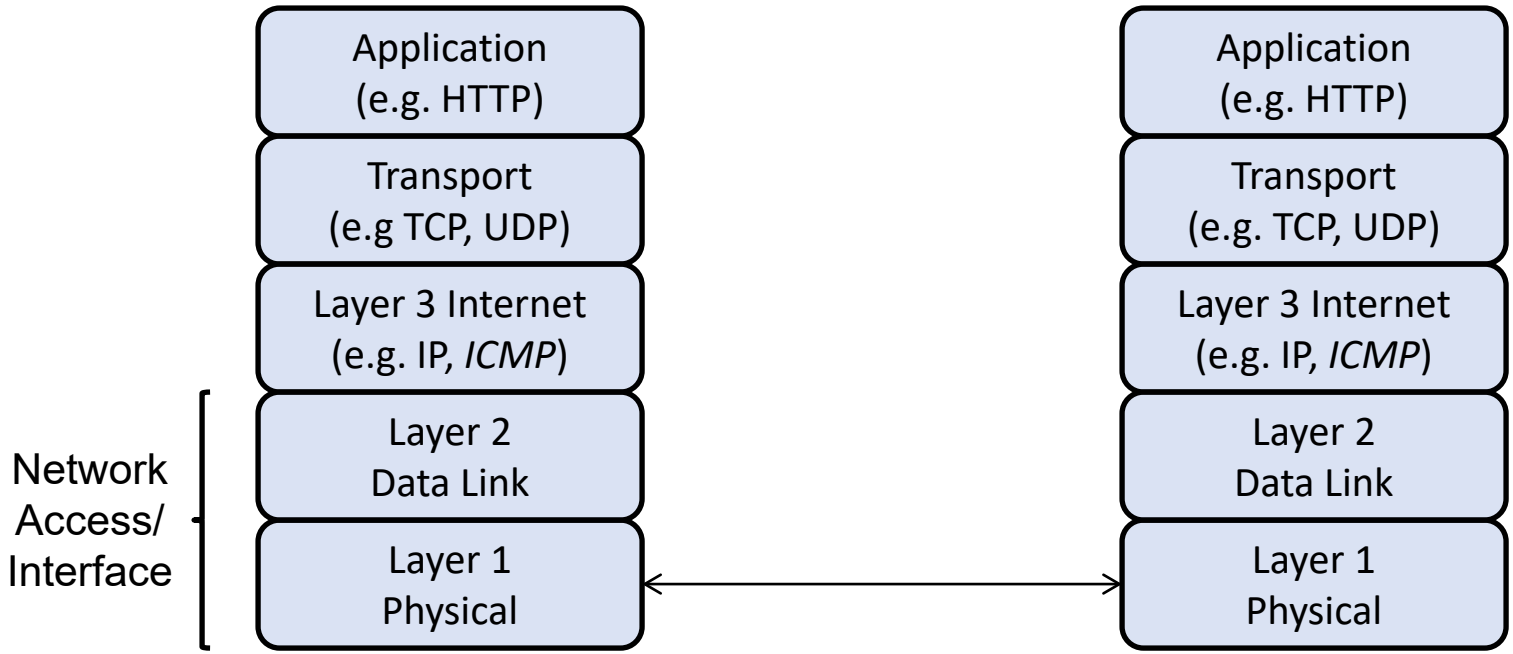


The OSI Seven-Layer Network Model

The ***Open Systems Interconnection (OSI) model***:
a conceptual/reference model that standardizes the *communication functions* of a telecommunication/computing system



The Internet (TCP/IP) Reference Model



Note: While the Open Systems Interconnection model (OSI model) gives 7 layers, in practice and in this module, we are interested in the above TCP/IP layering.

Network Layering

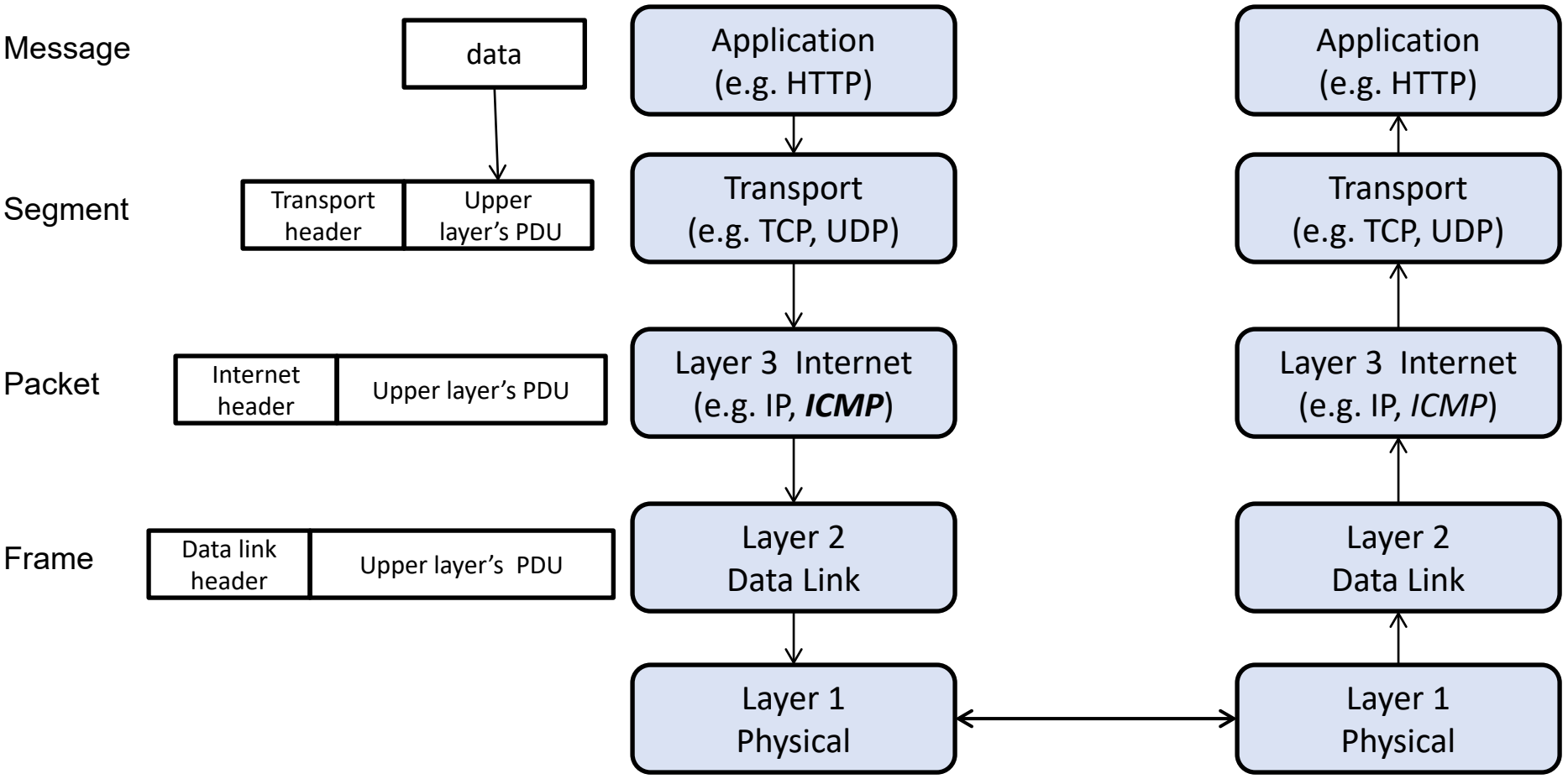
- Example of a protocol at **layer 5** (authentication protocol):

  (1) $A \rightarrow B$: "hello"

  (2) $A \leftarrow B$: certificate of B

- The **end-to-end connection** at layer 4 sends the message "hello" from *A* to *B* in Step (1), and sends *B*'s certificate in Step (2)
- The **routing protocol** at layer 3 routes the message packets from *A* to *B* over the Internet
- At **layer *N***:
  - A message to be sent is called: layer-*N* ***protocol data unit*** (PDU)
  - **Encapsulation** of upper (i.e. *N*+1) layer's PDU

Network Layers and Message Encapsulation



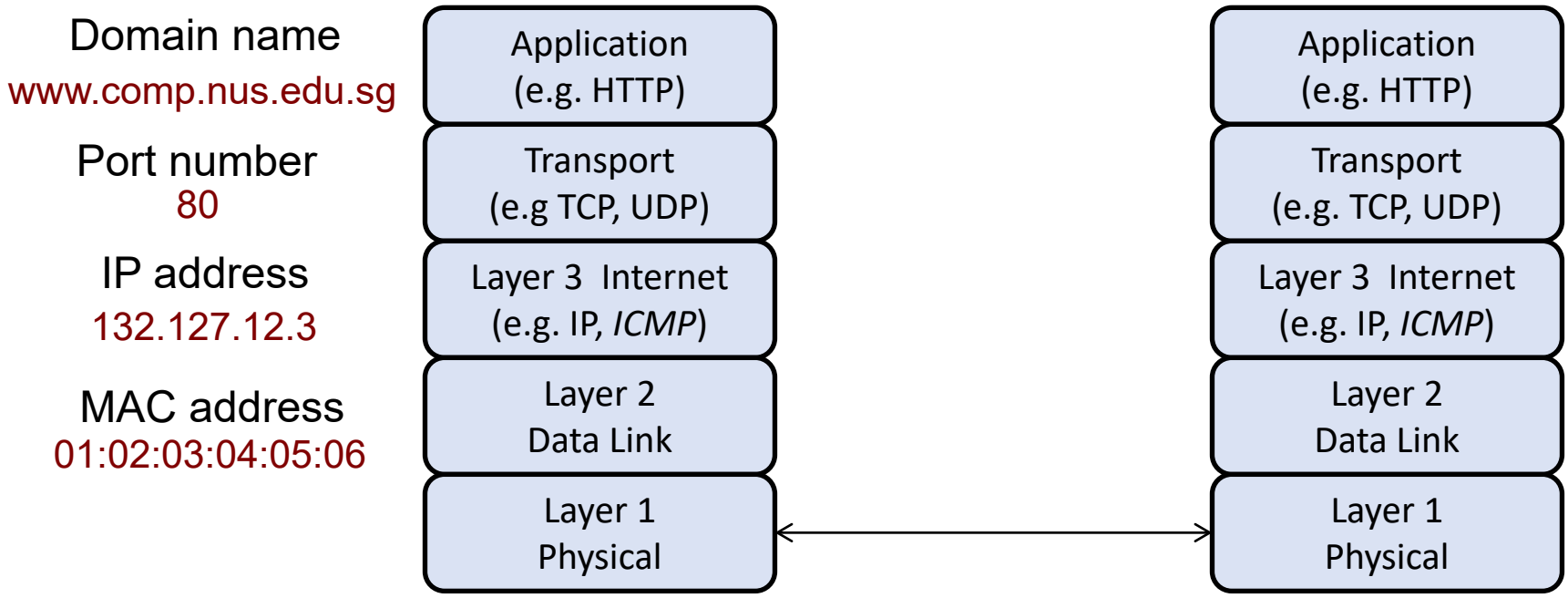
General PDU format:

Header: meta data **Payload**: the message/information intended to be sent

Note: An **ICMP** packet gets encapsulated by IP.
Yet, ICMP is commonly considered as part of Layer 3, since it assists IP.

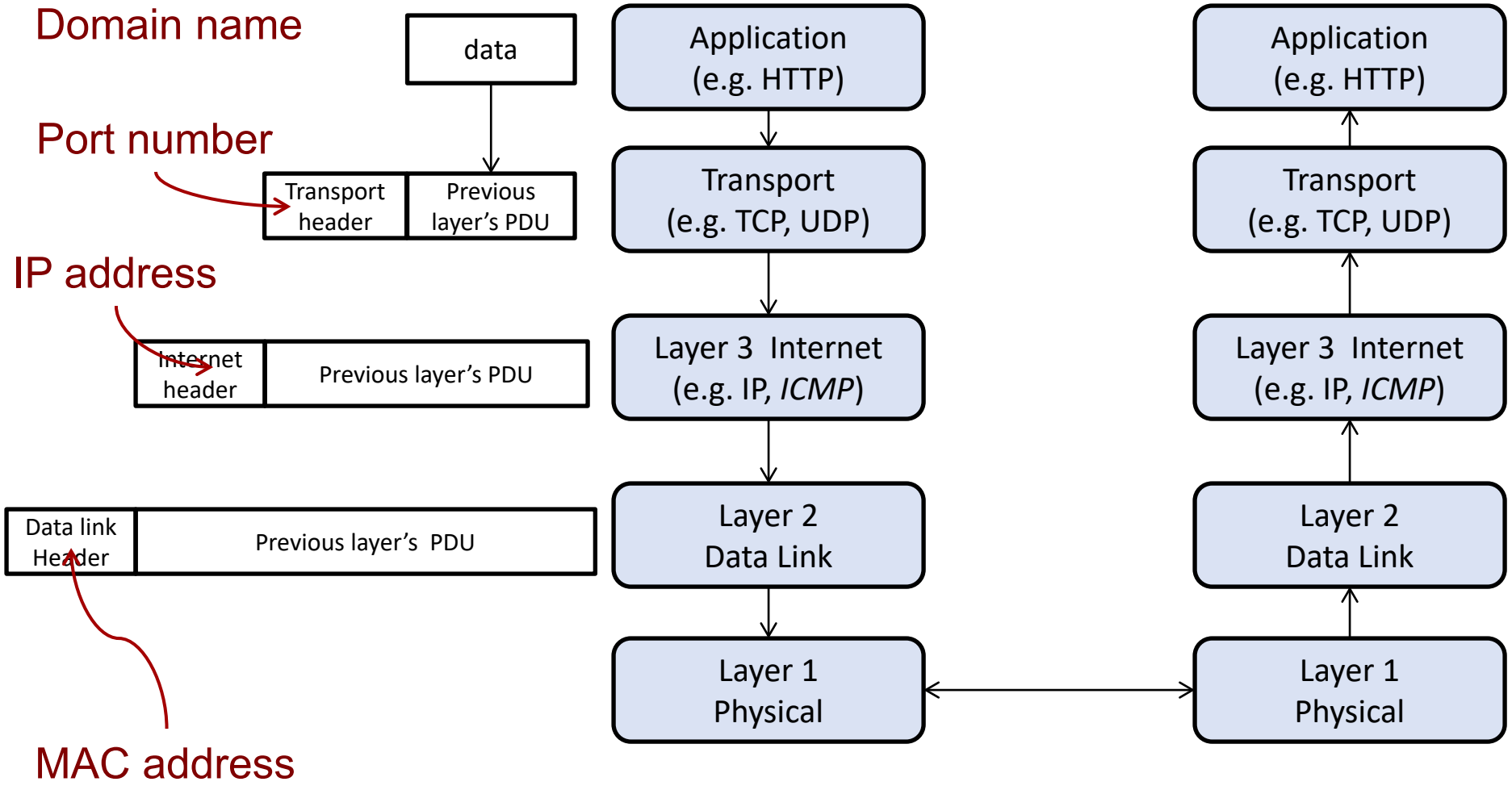
Internet Layers: Different Addressing Schemes

- Different ***addressing schemes*** at different layers



Note: MAC (medium access control) is *not* to be confused with crypto's MAC

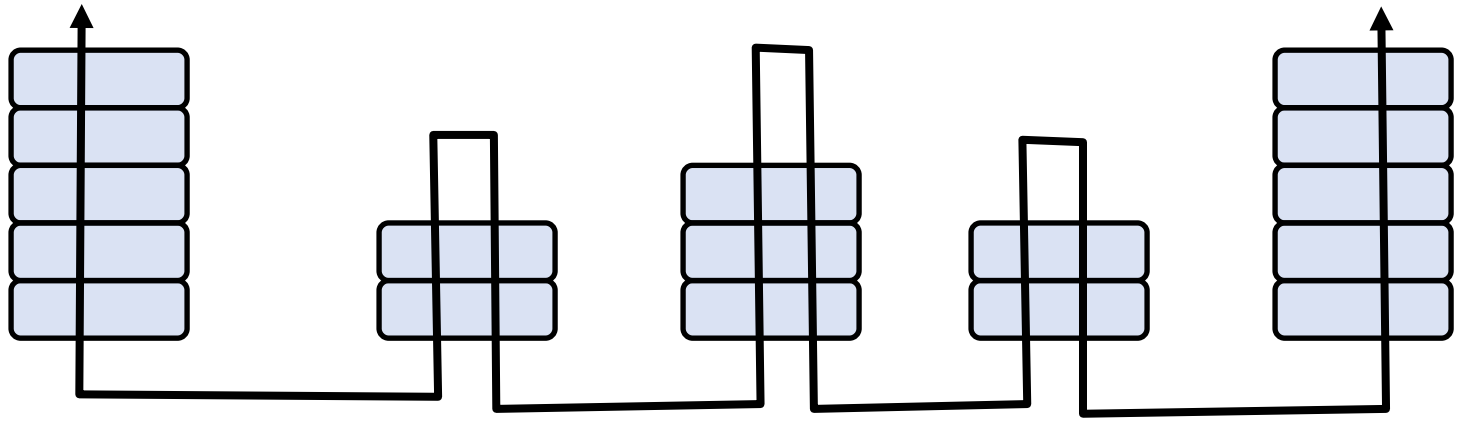
Addressing at Various Layers



Multiple Hops from Sender to Destination

- Note that data may go through **multiple *hops***
- Note that intermediate nodes might change header information (typically up to layer 3)

- Can you guess each device type in the diagram below?
- Some networking devices: *router*, *switch*, *hub*, *repeater*

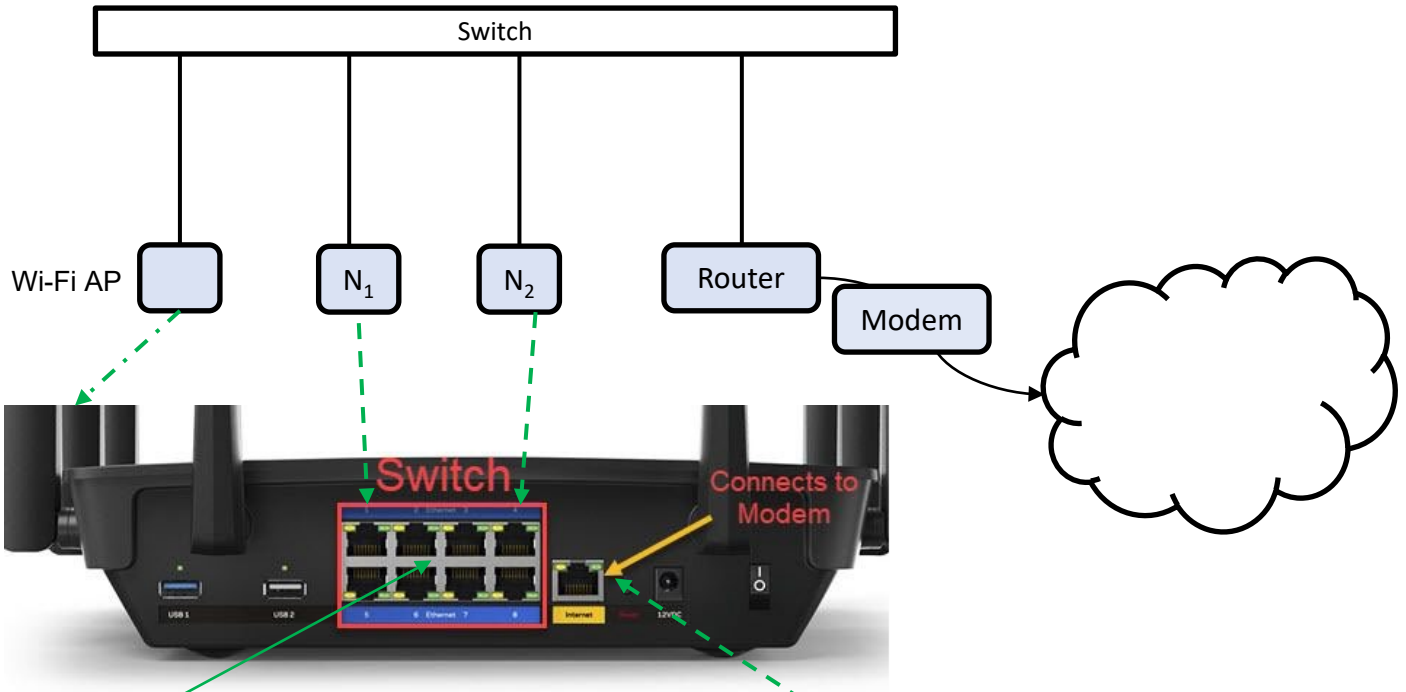


See: https://en.wikipedia.org/wiki/Computer_network#Network_nodes

Some Notes on Home Wireless Router

Our usual "home wireless router" is not *just* a router:

- It's a **switch**: so that you can wired-connect your machines to the network
- It's a **Wi-Fi access point**: so that you can wireless-access the Internet
- It's a **router**: so that you can connect your LAN to the Internet (via ISP)



Challenges in Network Security

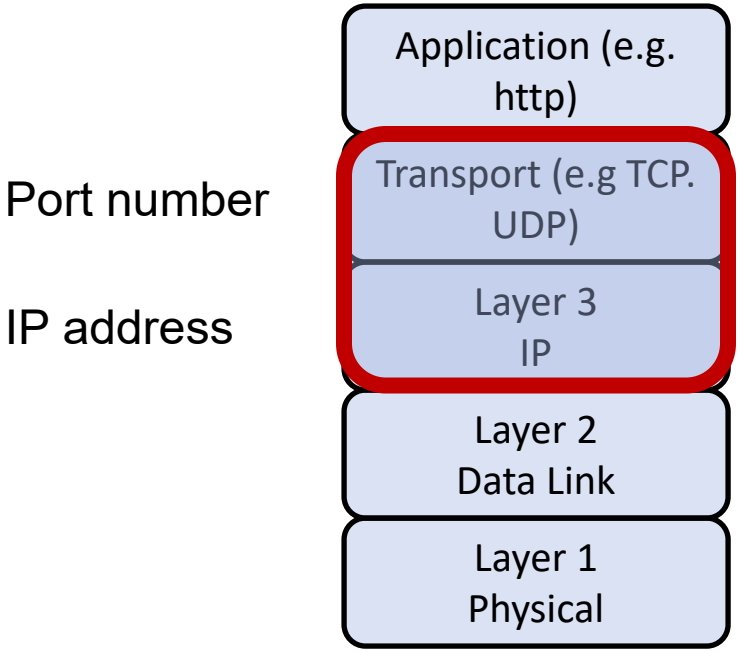
- [Complexity] Nodes are owned by **different semi-trusted parties**:
  - An intermediate node **has access** to both **header and payload** of routed packets: it can read (affecting confidentiality), modify (affecting authenticity), or drop (affecting availability) the packets
  - An attacker at a **certain layer in a host** can access all information at and below the layer: without any protection mechanisms, an attacker can easily send a packet with **spoofed** "source" IP address
- [Legacy & security tradeoff] Initial design of many networking protocols did *not* consider **intentional attacks** with the following possible threats (from Lecture 1):
  - **Interception**: unauthorized viewing
  - **Modification**: unauthorized change
  - **Fabrication**: unauthorized creation
  - **Interruption**: preventing authorized access

Challenges in Network Security

- [Security Requirements] Many additional different **security requirements**:
  - **Availability** can't be handled by crypto alone
  - Other requirements: **routing integrity**, **accountability**, **anonymity**, etc.
- [Management] There is a need to **isolate** and **control** information flow and network access

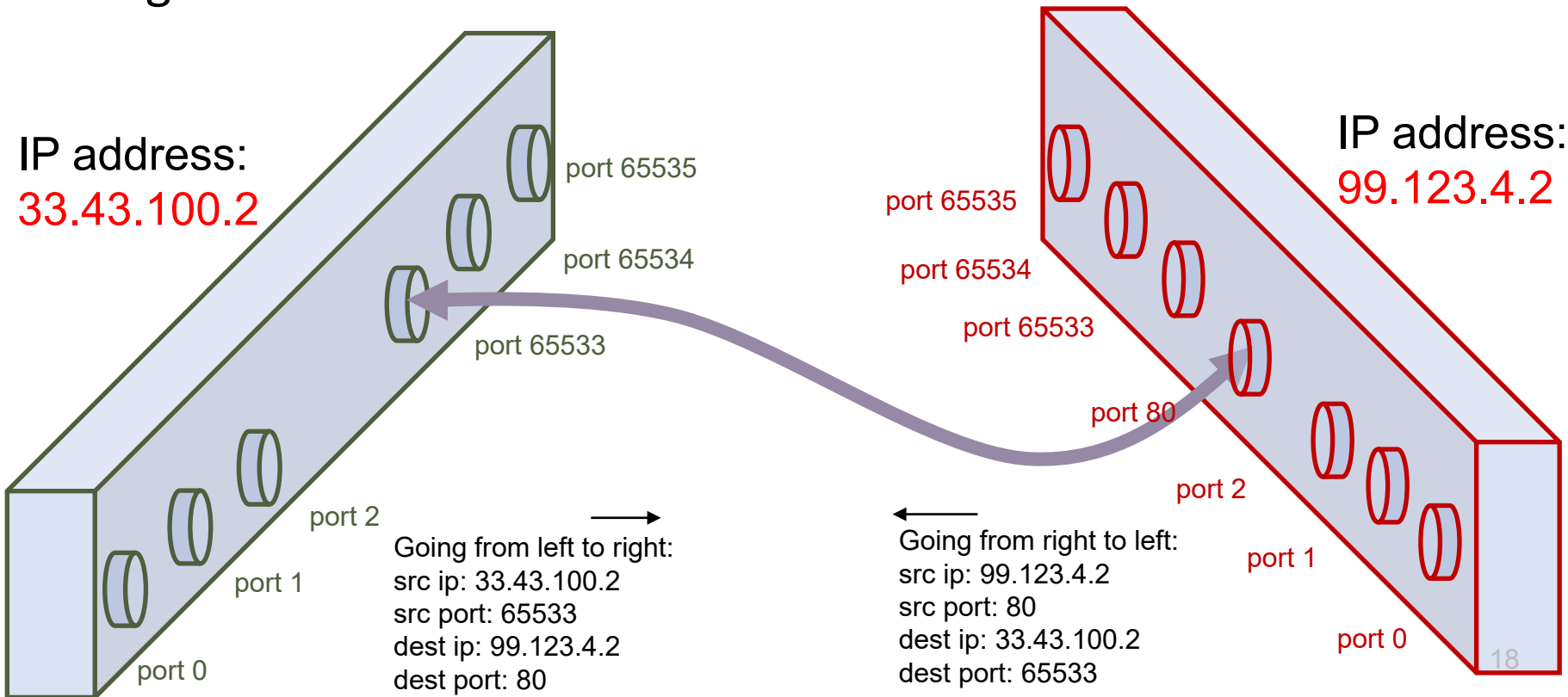
Additional Remarks on TCP, UDP, IP

(You can skip this part if you have already learnt networking)



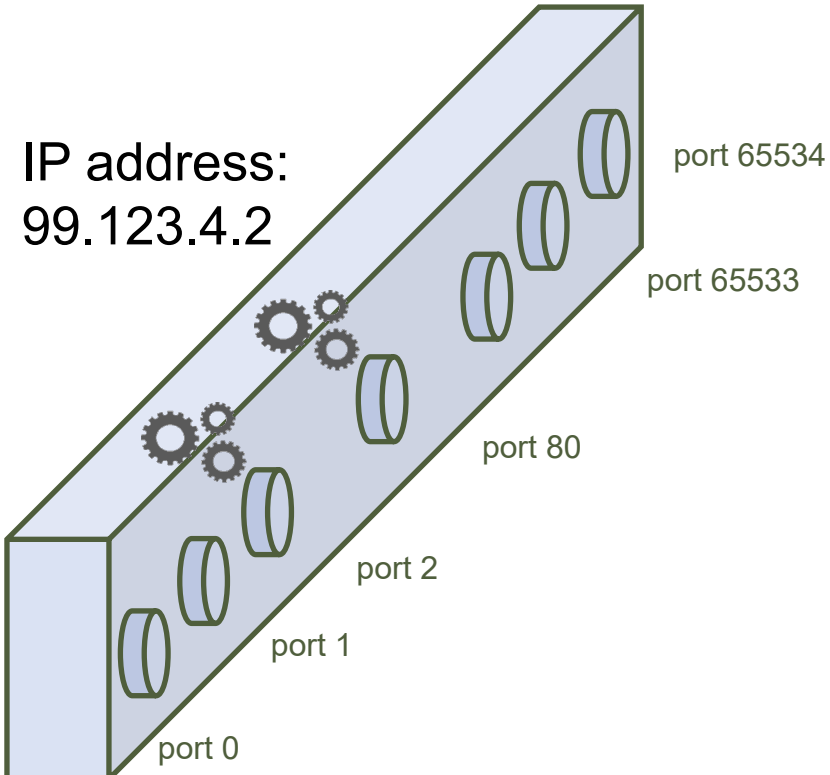
Transport and IP Layers

- A **network service** provided by a server:
  cam be accessed using both the server's ***IP address*** and ***port***
- Each node in the network has 65536 ports:
  well-known ports (system ports) are numbered from 0 to 1023
- A **connection** connects two ports of two communicating nodes:
  e.g. between 33.43.100.2:**65533** and 99.123.4.2:**80**



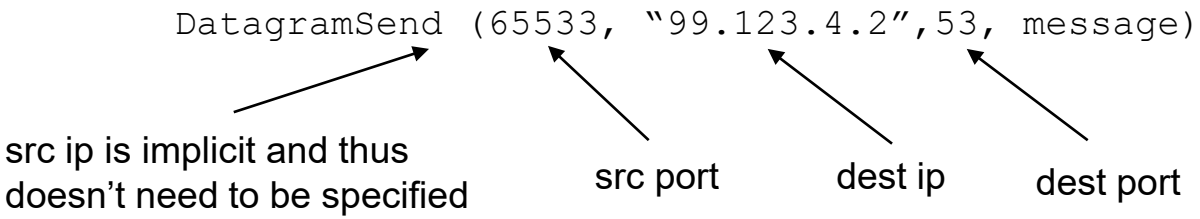
What do We Mean by "Listening to a Port", "Closed Port"?

- We can imagine that behind a certain port (e.g. 80) of a host, there is **an application process** waiting to process incoming data:
  - In such cases, we say that the process in the host is "**listening**" to the port, and the port is a "**listening port**"
- If it's not a listening port, then it is a "**closed port**":
  - Data sent to a closed port will be dropped



UDP

- If an application wants to send a **UDP datagram**, the library call is typically of the following form:



- The library call would construct the corresponding *IP packet* to be sent using data-link layer protocol:

- There is limit on the size of message: at most ~65,000 bytes
- The library call does not return a result indicating whether the destination has indeed received the packet:
  - There is a possibility that the packet is lost!
  - This is the property of UDP protocol (i.e. how it works): ***connectionless-oriented*** *and* ***unreliable communication***

TCP

- In contrast, TCP/IP is ***connection-oriented***
- An application would typically make the library calls of the following form and in the specified order:

```
-   P = open_connect (65533, "99.123.4.2", 80)
-        send (P, out_message)
-        read (P, in_message)
-   close_connection(P)
```

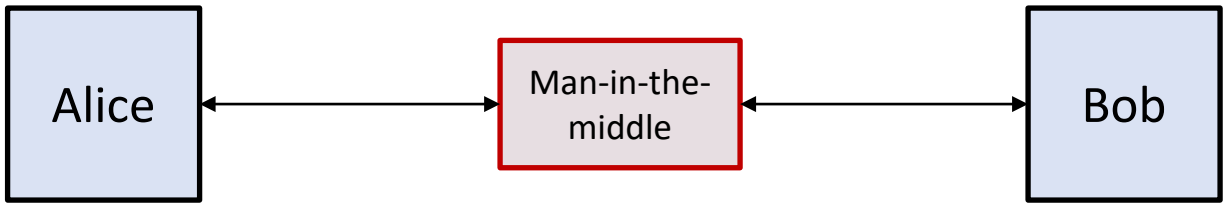
There could be multiple rounds of send/read

- `open_connect` would carry out a TCP 3-way handshake between the two nodes
- `send` would construct the corresponding *IP packet* to be sent using data-link layer protocol:

- If the message is too long, multiple IP packets will be formed
- TCP is ***reliable***: it has mechanisms to **re-transmit, re-order, acknowledge packets** so that the destination can receive the sent messages in the correct order

TCP Reliability and Its Security

- TCP is ***reliable*** for message-delivery purposes, but it *not* "***secure***"
- Intermediate nodes along the communication route can still **read and modify data** in the header and payload of the packets
- The intermediate nodes can act as a **man-in-the-middle** at the IP layer: this can be addressed by using TLS, e.g. HTTPS

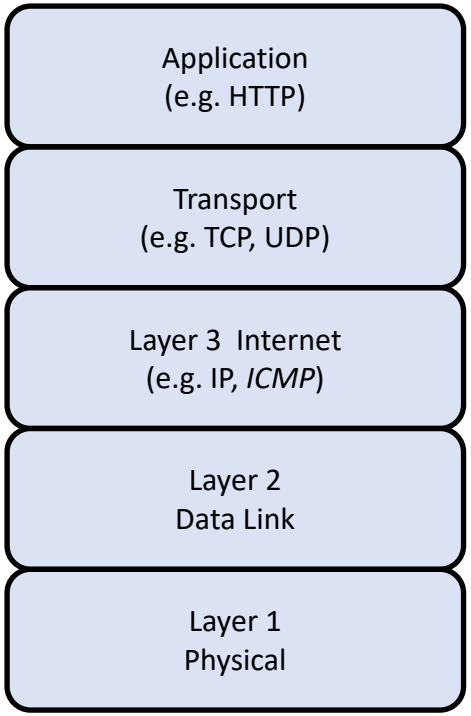


6.2 Name Resolution and Attacks

Naming Schemes and Resolution

- Each peer entity has a **name**
- On a single node, at different layer, the name can be **different**

	www.comp.nus.edu.sg	Domain name
Port number plays little role in name resolution attack	80	Port
	132.127.12.3	IP address
	01:02:03:04:05:06	MAC address



Naming Schemes and Resolution

- When a peer entity uses the virtual connection in the layer below, it needs to find out the corresponding ***name mapping***
- Example: finding the IP address of a domain name
- **Protocols** that perform name mappings are known as "***resolution***" protocols
- Many initial design of resolution protocols **didn't** take security into account, and thus *easy* for attackers to **manipulate the outcome**

Resolution Protocols

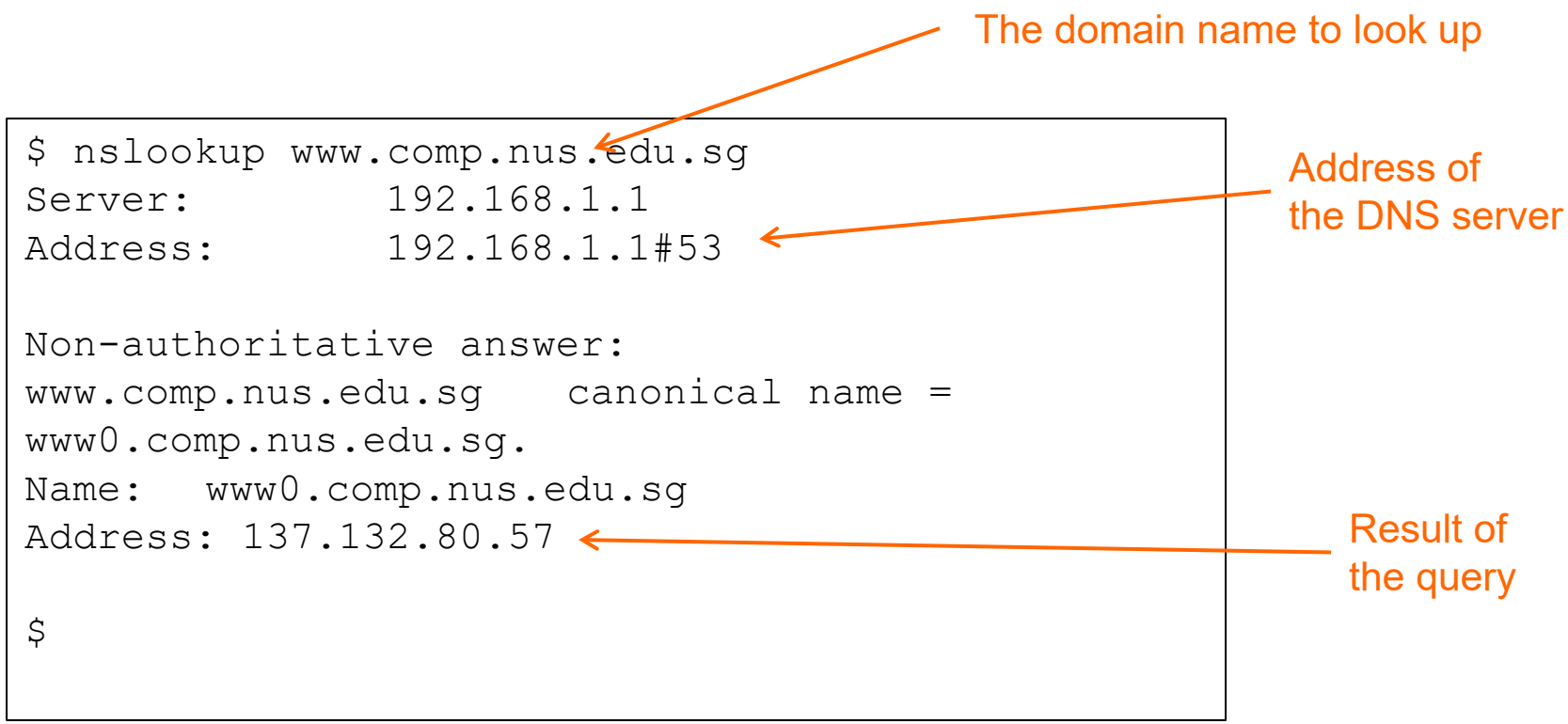
- **Domain Name System (DNS)**:
  - Maps **domain name** to **IP address**
  - A hierarchical decentralized naming system
  - An attacker can target the association of domain name with IP address

In this module, we only consider a *basic* type of DNS attack

- **Address Resolution Protocol (ARP)**:
  - Associate/map **IP address** (logical address) with/to **MAC address** (physical address)
  - Use a *broadcast mechanism* on a local network
  - An attacker **on the local network** can target the association

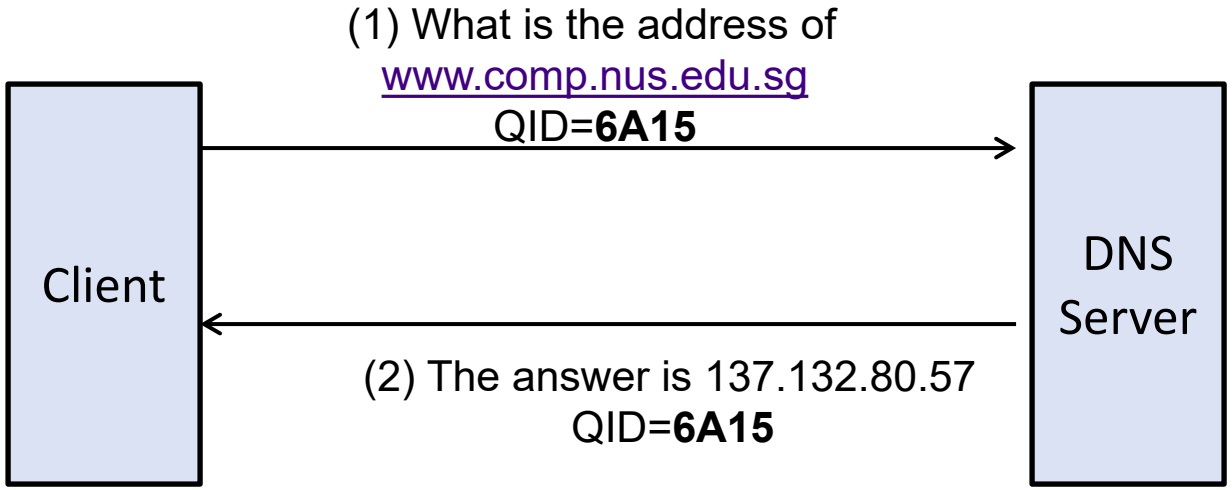
DNS (Domain Name System)

- Given a domain name (e.g. www.comp.nus.edu.sg), its IP address can be found by either looking up a locally stored **host table**, or by **querying a DNS server**. The process is known as ***name resolution***.
- The entity (a.k.a client) that initiates the query is called the ***resolver***
- If the address is found, we say that the domain name is ***resolved***



(Lightweight) Authentication of DNS Query

- Each query contains a 16-bit number, known as **Query ID (QID)**
- The response from the name server must also contains a QID
- If the QID in the response doesn't match the QID in the query, the client rejects the answer
- Note that **no** encryption/MAC is involved



Remark: In the original design consideration, the QID is probably *not* meant for authentication, but as an efficient way to match multiple queries

Local DNS Attack Scenario

Alice:

- is using a café's free/open (without protection) WiFi to surf the web
- wants to visit the webpage www.comp.nus.edu.sg
- types the domain name into the browser's address bar

- **Alice's browser:**
  - makes a query to a DNS server to determine the IP address
  - then connects to the IP address (after the browser obtains the IP address)

Local DNS Attack Scenario

Active Attacker (Mallory):

- We consider an attacker at the ***physical layer***:
  for example, he/she can be another person in the café
- Since the WiFi is not protected, the attacker can:
  - Sniff data from the communication channel
  - Spoof data into the communication channel
- Attacker, however, can't remove/modify data already sent by Alice
- Attacker also owns a **web server** (e.g. with IP address 100.100.100.3), which is a spoofed SoC website

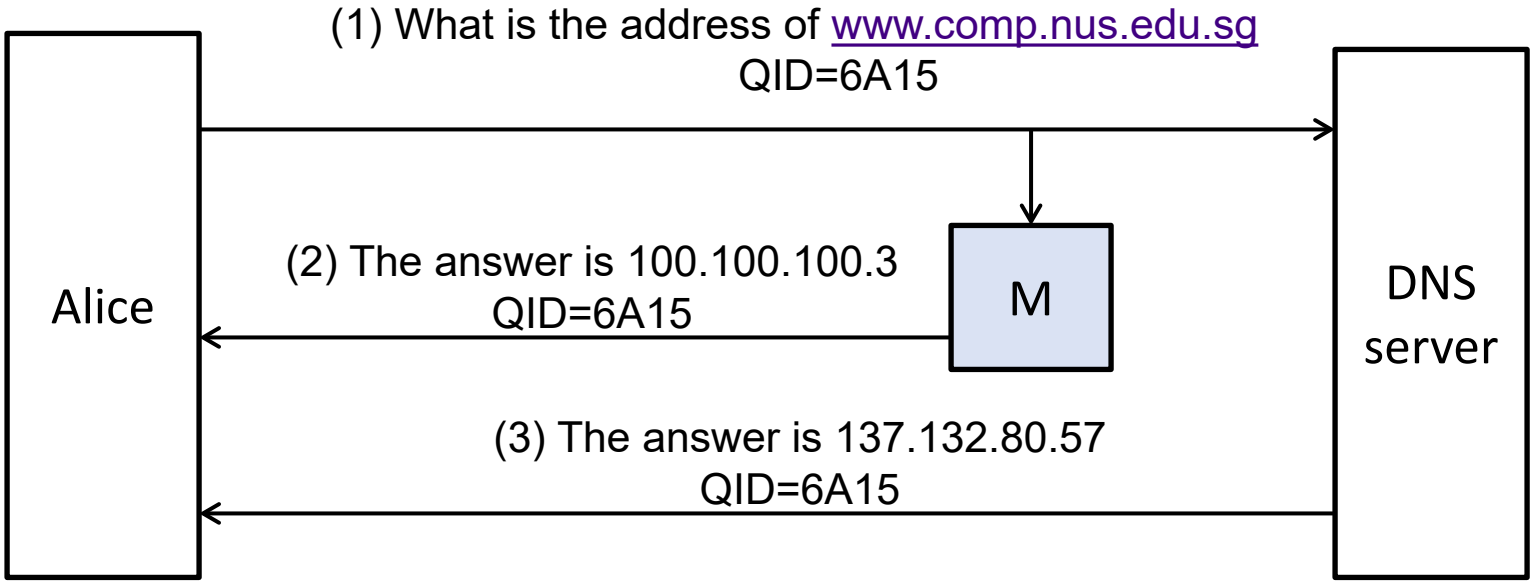
The Attack (See [PF] page 409)

(1) Alice asks for the address.

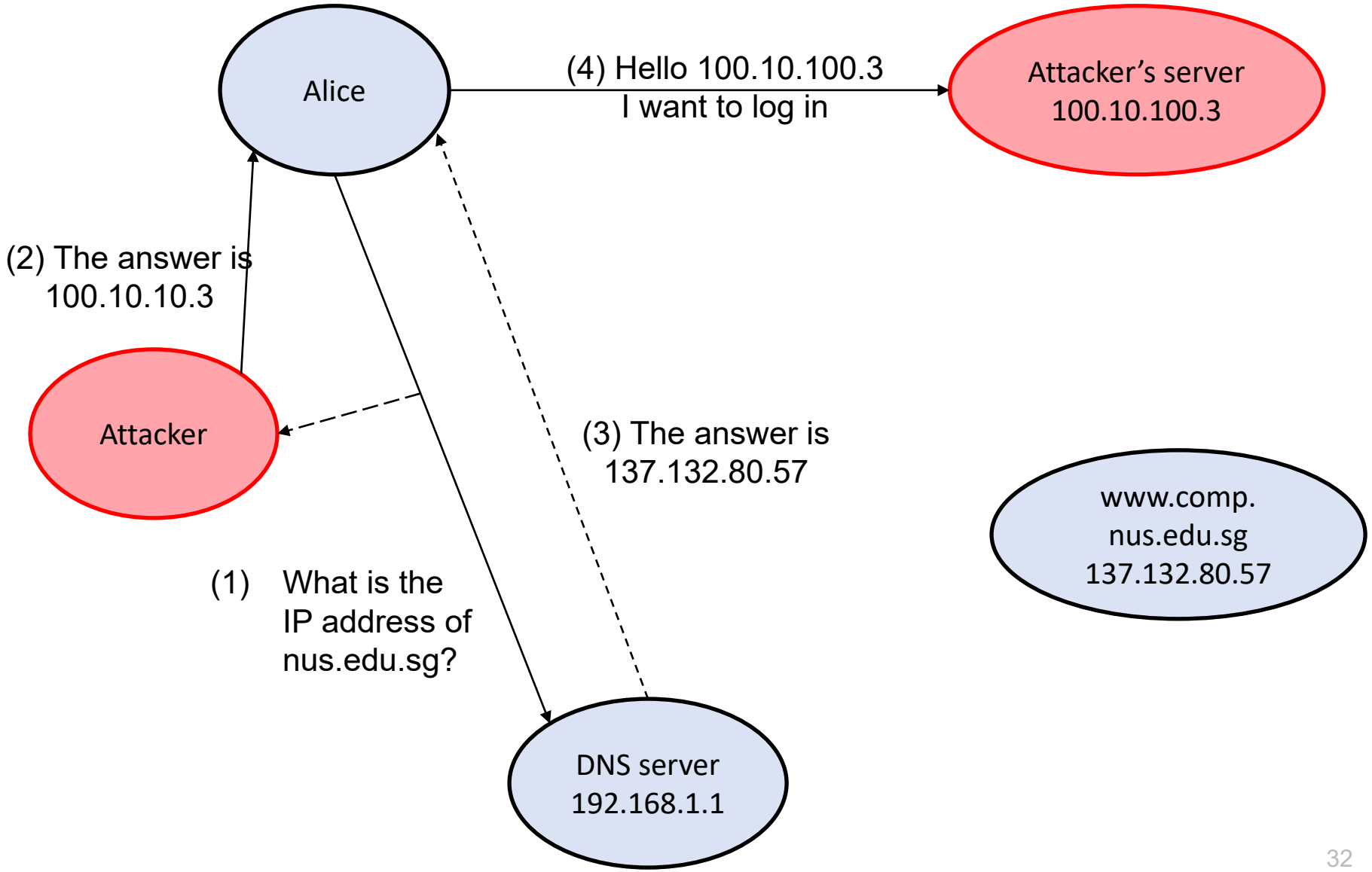
(2) Mallory sniffs and knows about it.
She quickly spoofs a reply with the same QID.

(3) DNS server also sends a reply.
Since Mallory is closer to Alice,
Mallory's reply is likely to reach Alice first.

Alice takes the *first* reply as answer, and connects to 100.100.100.3.



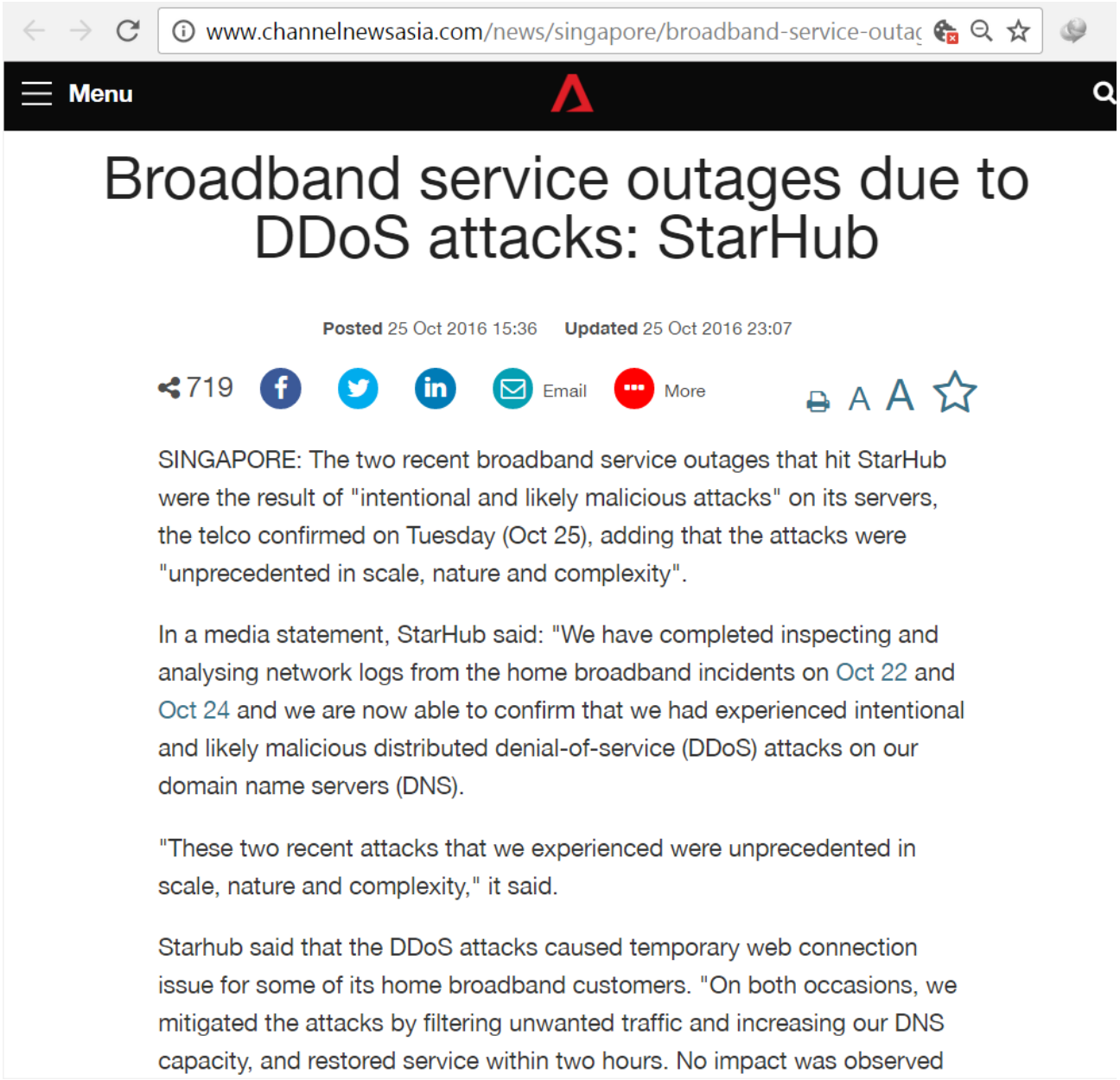
DNS Spoofing: Overall Attack Scenario



Some Remarks

- DNS operates at the **application layer**.
- Although the attacker is at the physical layer, for ease of analysis, we can assume that the attacker is **just below the application layer**: That is, there exists some virtual connection that can send the message across.
  Hence, the previous slide doesn't mention about the MAC and IP addresses, etc., of the DNS server.
- The DNS is an important component as it resolves the domain name. Hence, an DNS server can be the "**single-point-of-failure**" for the network.
- A DoS attacks, instead of attacking a Web server, could attack the **DNS server** instead.
  E.g. see attack on WikiLeak (See [PF6.5] pg. 414, [PF] pg. 485),
  StarHub attack 2016 (next slide).

Recent DNS Attacks



www.channelnewsasia.com/news/singapore/broadband-service-outag

Menu

Broadband service outages due to DDoS attacks: StarHub

Posted 25 Oct 2016 15:36 **Updated** 25 Oct 2016 23:07

719 Email More A A

SINGAPORE: The two recent broadband service outages that hit StarHub were the result of "intentional and likely malicious attacks" on its servers, the telco confirmed on Tuesday (Oct 25), adding that the attacks were "unprecedented in scale, nature and complexity".

In a media statement, StarHub said: "We have completed inspecting and analysing network logs from the home broadband incidents on Oct 22 and Oct 24 and we are now able to confirm that we had experienced intentional and likely malicious distributed denial-of-service (DDoS) attacks on our domain name servers (DNS).

"These two recent attacks that we experienced were unprecedented in scale, nature and complexity," it said.

Starhub said that the DDoS attacks caused temporary web connection issue for some of its home broadband customers. "On both occasions, we mitigated the attacks by filtering unwanted traffic and increasing our DNS capacity, and restored service within two hours. No impact was observed

Channel News Asia, 25 Oct 2016

6.3 Denial of Service Attacks

DOS Attacks

- **Availability**: the property of being accessible and usable upon demand by an authorized entity
- **Denial of service (DoS)**:
  - The prevention of authorized access to resources or the delaying of time-critical operations
  - An attack on availability
- Types of **DoS attacks**:

	Stopping Service	**Exhausting Resources**
Local Attack	• Process killing • Process crashing • System reconfiguring	• Spawning processes • Filling up file system
Remote Attack	Sending malformed packet attacks	***Packet flooding***

DoS Attack Types

- **Local attacks** can be more easily tracked
- Sending **malformed attack** remotely does not usually work on updated OSes
- **Packet flooding** attacks:
  - Many effective DOS attacks simply remotely **flood** the victims with overwhelming requests/data
  - The attacker can ***amplify*** small traffic to obtain large traffic, typically by using available **public servers** (Internet infrastructure), such as DNS, NTP, CharGen

ICMP/Smurf Flood Attack [PF] page 404

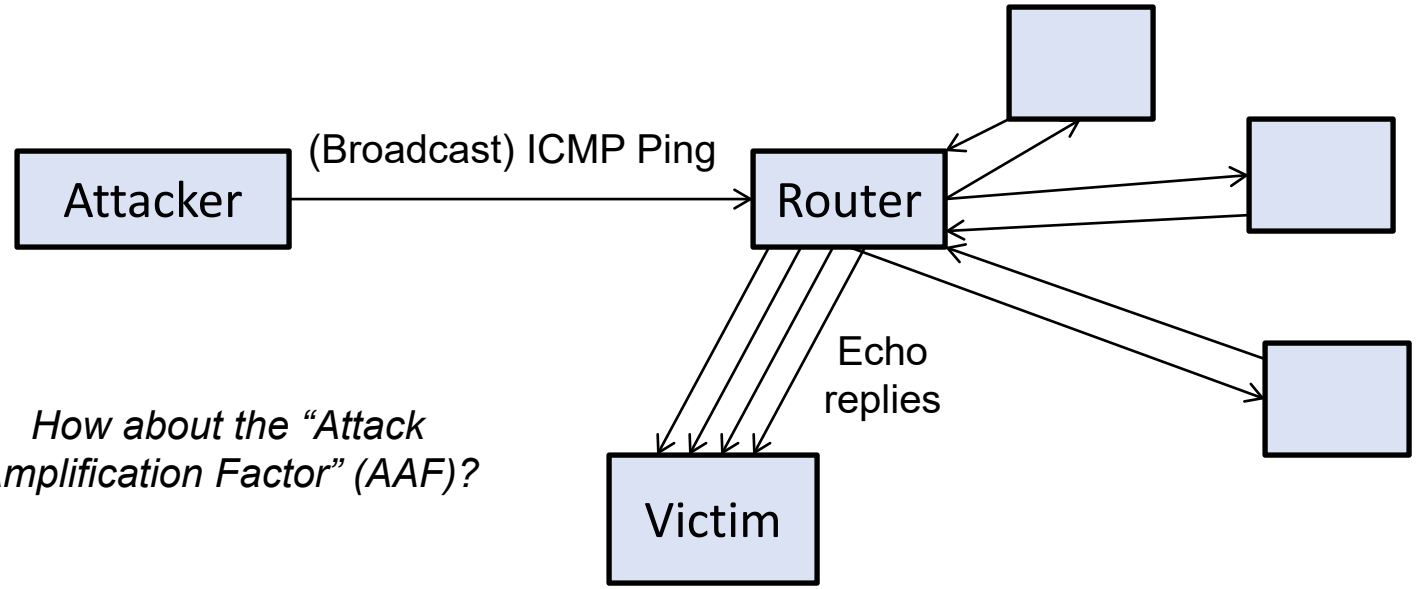
(1) An attacker sends the "**ICMP PING**" request to a router, instructing the router to **broadcast** this request to all local nodes. The request' source IP address is spoofed with the victim IP address.

(2) The router broadcasts this **Echo request**.

(3) Each entity who has received this request, replies to it by sending an "**Echo reply**" to the source, which is the victim

The victim is thus overwhelmed with "**Echo reply**" from the entire network.

The attacker takes advantage of the ***amplification*** effect



How about the "Attack Amplification Factor" (AAF)?

ICMP/Smurf Flood Attack: Preventive Measures

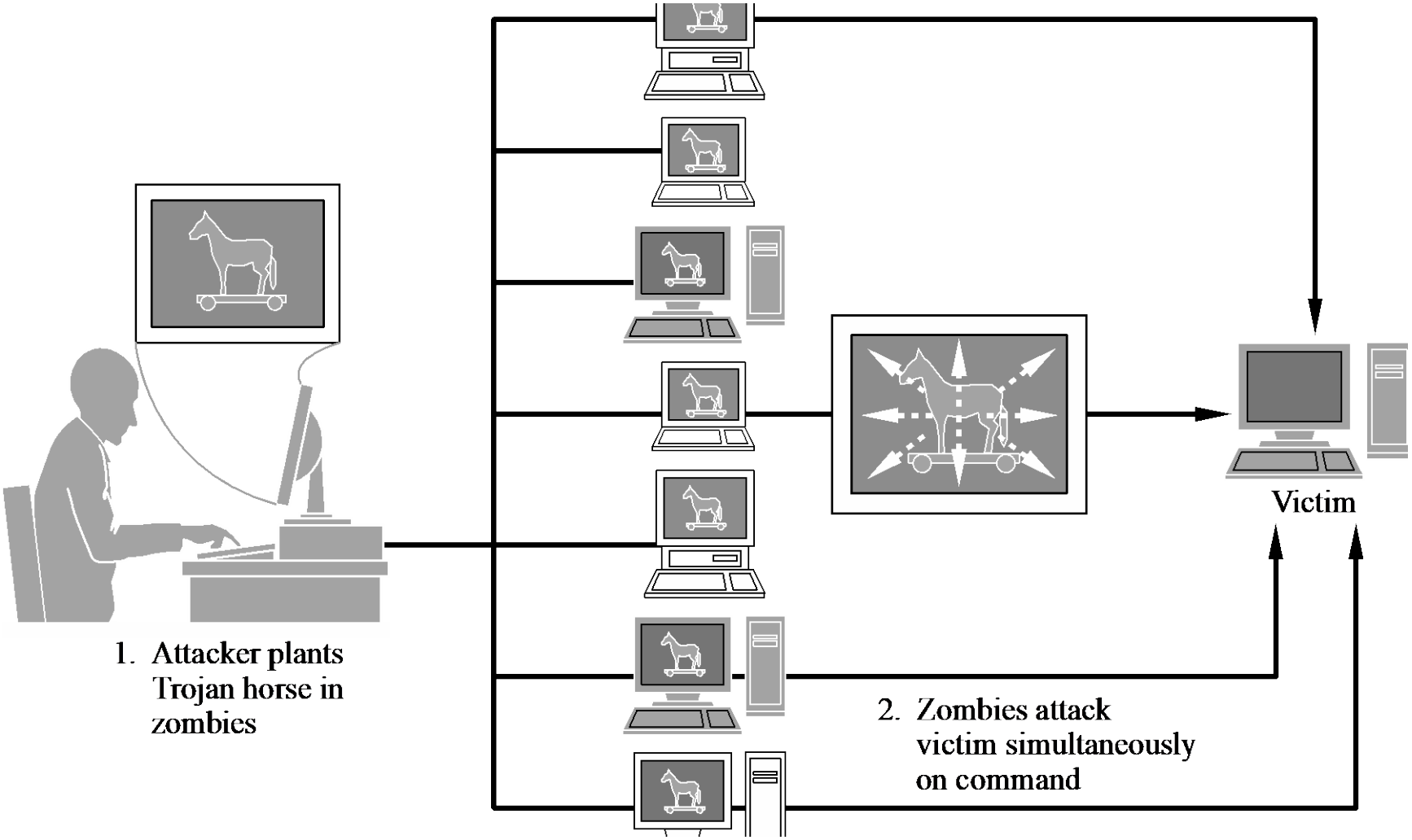
- *Question: Is this attack technique still effective?*
- No!
- *Why not?*
- Most routers are now configured **not** to broadcast the requests
- To prevent the attack, this measure simply **disables** a feature that was previously thought to be useful

See: IP broadcasting, http://en.wikipedia.org/wiki/Broadcast_address

Example of Application-Layer DoS Attack (HTTP Get)

- Simply flood a **Web server** with **HTTP requests**
- Example: **MyDoom worm**, which targeted SCO's website.
  Attacks started on Feb 12, 2004.
  This is after the SCO's legal actions and public statements against Linux.
- For this attack to be effective, **a large number** of attackers are required.
  (Since each attacker can send requests at a low rate only).
- When DoS is carried out by large number of attackers, this is called **Distributed Denial of Service (DDoS)**.

Distributed Denial of Service (DDoS)

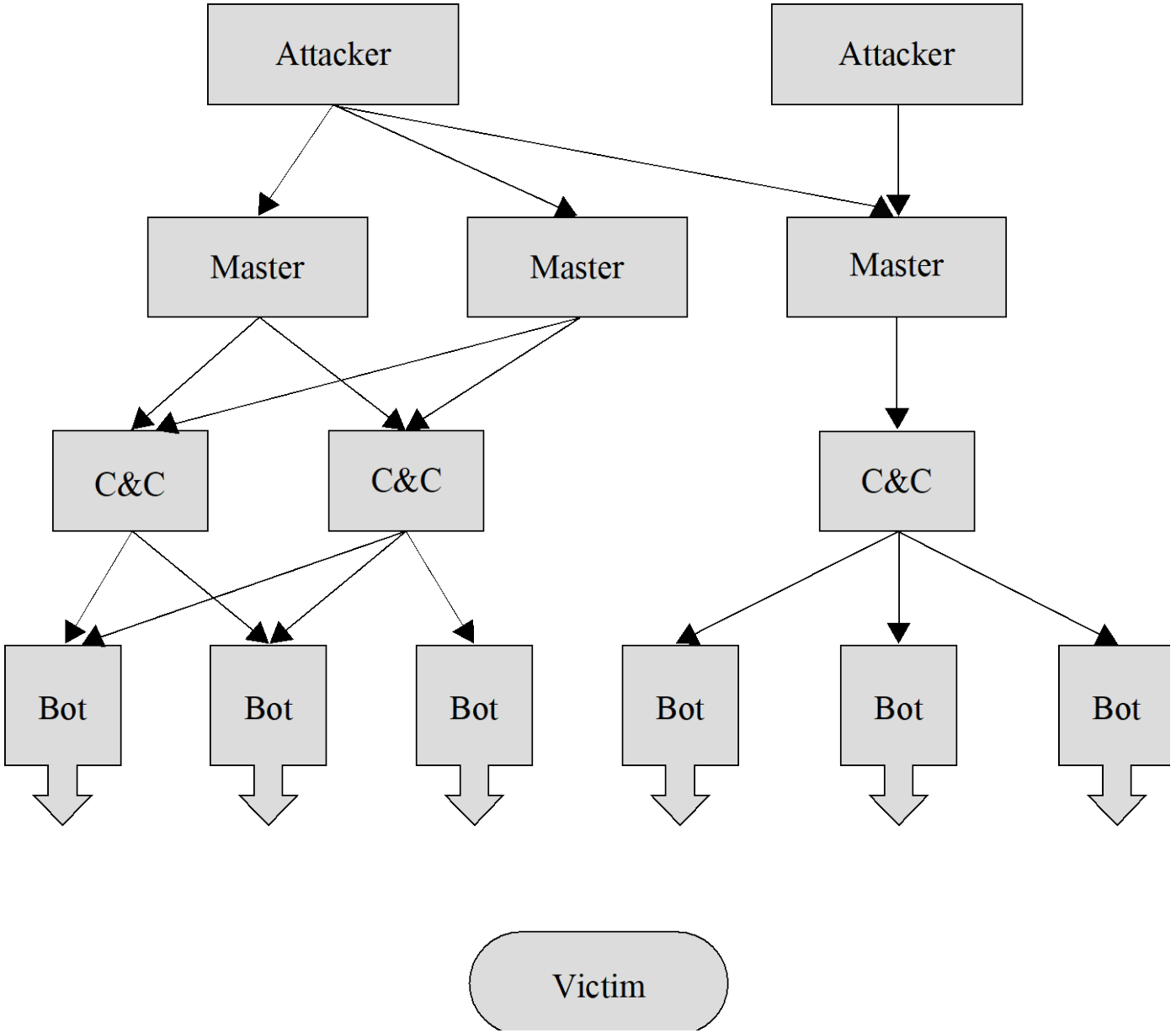


Botnet

- A ***bot*** (aka ***zombie***) is a compromised machine
- A ***botnet*** (aka ***zombie army***) is a large collection of connected bots, communicating via covert channels
- A botnet has a **command-and-control** mechanism, and thus can be control by an individual to carry out DDOS
- Possible **usages** of a botnet:
  - DDoS flooding, vulnerability scanning, anonymizing HTTP proxy, email address harvesting, cipher breaking!
- See Wiki for the size of known botnets: http://en.wikipedia.org/wiki/Botnet

Question: Why **covert** channels are used by a botnet?

Botnet



IoT Devices: a New Trend

TECHNOLOGY

M1 launches commercial IoT network in boost to Singapore's Smart Nation drive

All Sections

By Kevin Kwang
@KevinKwangCNA

07 Aug 2017 06:20PM
(Updated: 07 Aug 2017 09:28PM)

Smart rubbish bins enabled with sensors will trigger an alert to cleaners to clear them after a certain level is met. (Photo: M1)

Channel News Asia,
7 Aug 2017

Botnet

Optional

- Size of known botnets:
  see http://en.wikipedia.org/wiki/Botnet

Date created	Date dismantled	Name	Estimated no. of bots	Spam capacity (bn/day)	Aliases
2009 (May)	November 2010 (not complete)	BredoLab	30,000,000[51]	3.6	Oficla
2008 (around)	2009-Dec	Mariposa	12,000,000[46]		
2008 (November)		Conficker	10,500,000+[47]	10	DownUp, DownAndUp, DownAdUp, Kido
		Marina Botnet	6,215,000[37]	92	Damon Briant, BOB.dc, Cotmonger, Hacktool.Spammer, Kraken
2010 (around)		TDL4	4,500,000[56]		TDSS, Alureon
		Zeus	3,600,000 (US only)[57]		Zbot, PRG, Wsnpoem, Gorhax, Kneber
2011 or earlier	2015-02	Ramnit	3,000,000[58]		
2007 (around)		Cutwail	1,500,000[42]	74	Pandex, Mutant (related to: Wigon, Pushdo)

Other Examples of Reflection Attack

Optional

DNS reflection attack:

"During a **DNS amplification attack**, the perpetrator sends out a **DNS query** with a forged IP address (the victim's) to an open DNS resolver, prompting it to reply back to that address with a **DNS response**. With numerous fake queries being sent out, and with several DNS resolvers **replying back simultaneously**, the victim's network can easily be overwhelmed by the sheer number of DNS responses."

https://security.stackexchange.com/questions/93820/dns-reflection-attack-vs-dns-amplification-attack

- For UDP-based attacks, ***Bandwidth Amplification Factor (BAF)*** is the number of UDP payload bytes that an amplifier sends to answer a request, compared to the number of UDP payload bytes of the request
- See the table on the right for **different BAFs**
- What was the **largest DDoS attacks**?
  → an unnamed customer of the US-based service provider Arbor Networks, reaching a peak of about **1.7 terabits per second**

UDP-based Amplification Attacks

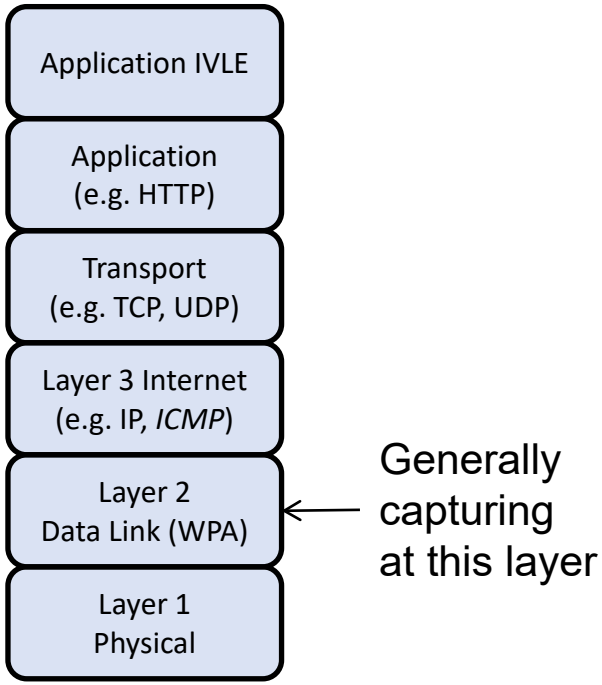
Protocol	Bandwidth Amplification Factor
Memcache	50000
NTP	556.9
CharGen	358.8
DNS	up to 179 [50]
QOTD	140.3
Quake Network Protocol	63.9
BitTorrent	4.0 - 54.3 [51]
SSDP	30.8
Kad	16.3
SNMPv2	6.3
Steam Protocol	5.5
NetBIOS	3.8

https://en.wikipedia.org/wiki/Denial-of-service_attack

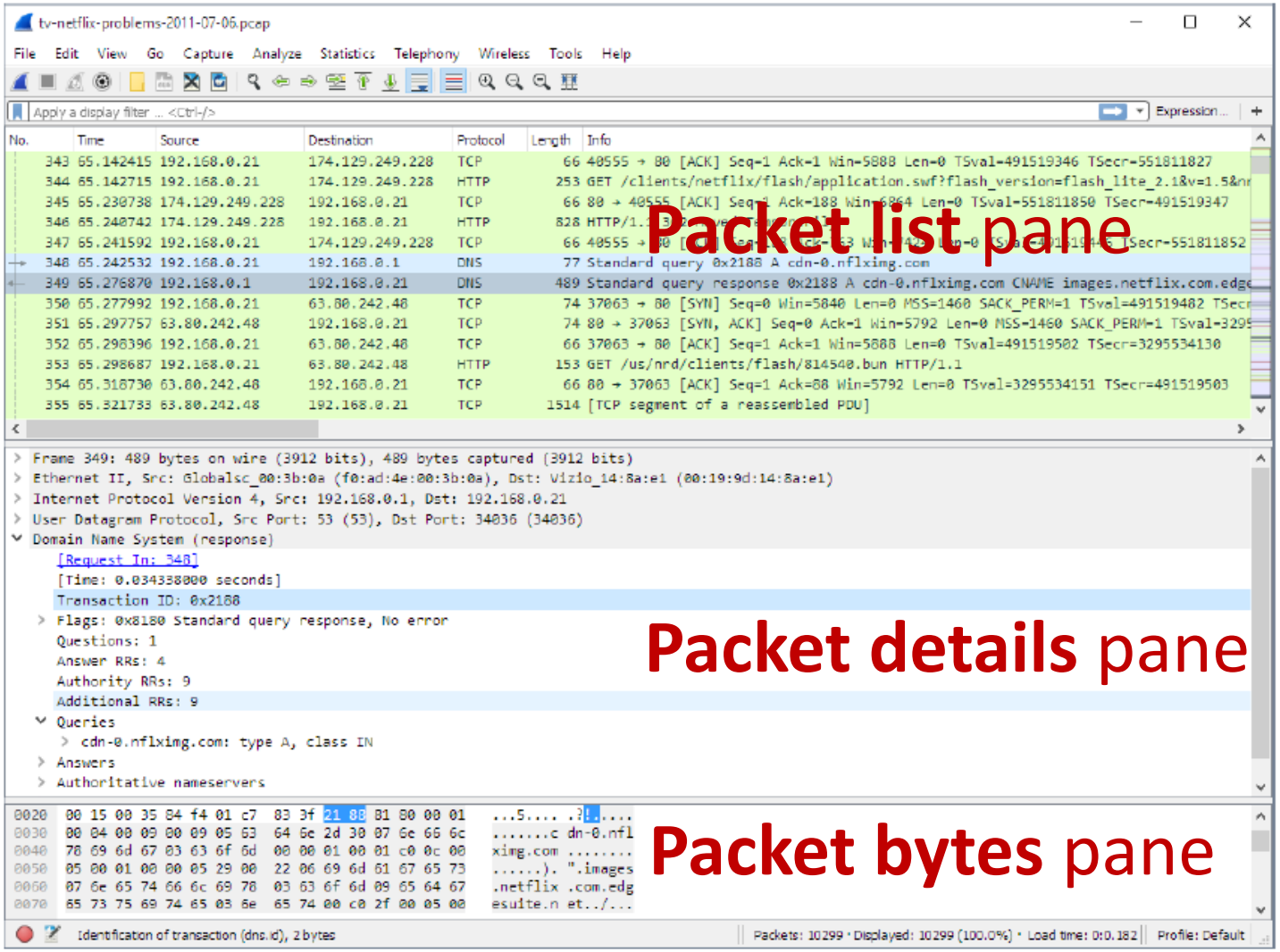
6.4 Useful Tools

Wireshark (a Packets Analyzer)

- **Wireshark**: a popular free open-source network packet analyzer, https://www.wireshark.org/.
- What does Wireshark exactly capture?
  - Generally, it performs capturing at the **link layer**
  - "The capturing framework is placed **between** the NIC driver and higher layer protocols in the kernel (e.g. TCP/IP)"
  - See the following FAQ for more info: https://ask.wireshark.org/questions/22956/where-exactly-wireshark-does-captures-packets

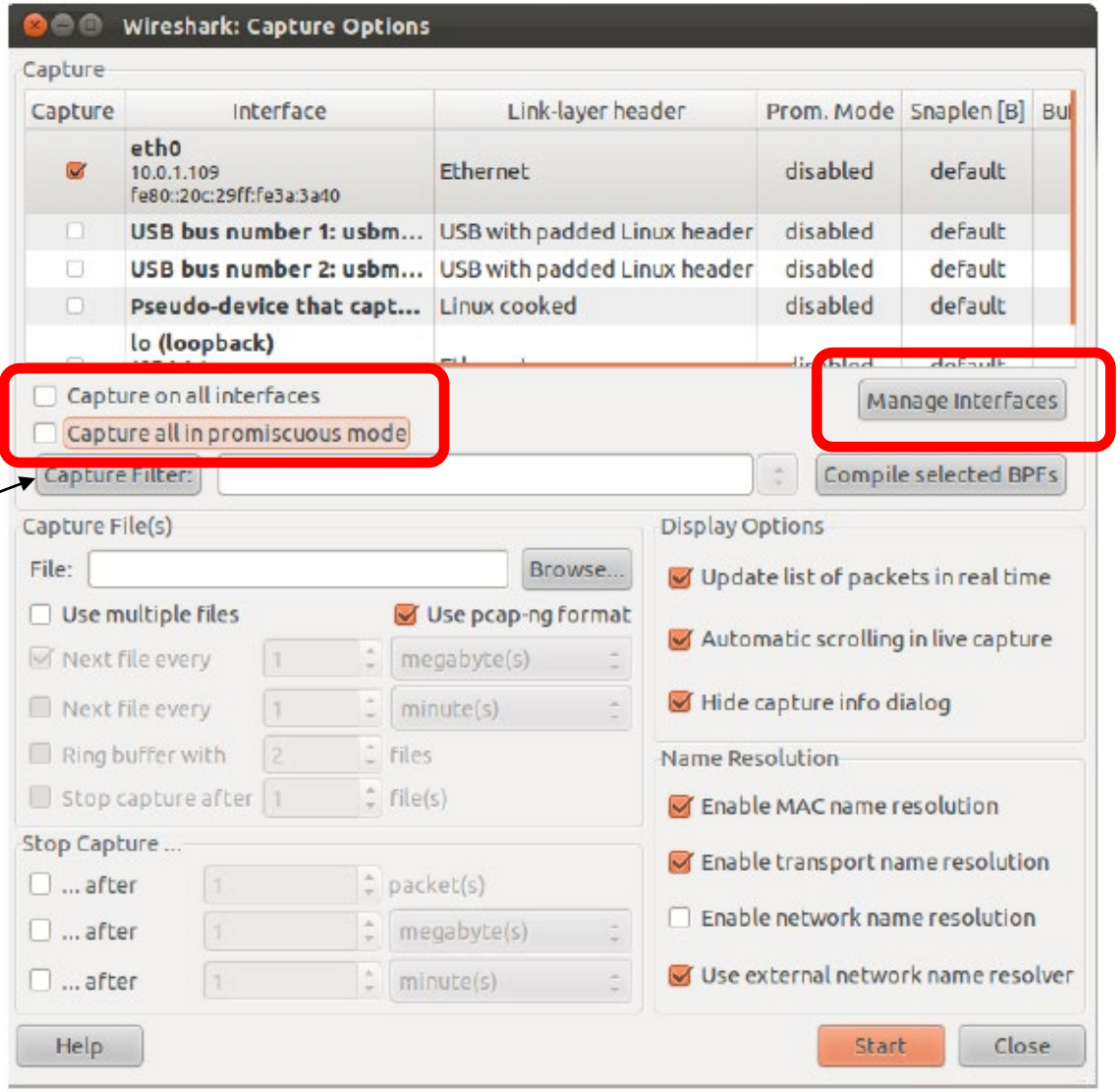


Wireshark (a Packets Analyzer)



For Wireshark usage, see: Wireshark User's Guide, https://www.wireshark.org/docs/wsug_html/

Sniffing using Wireshark: Capture Options



Capture filter

Source: Wireshark User's Guide

Sniffing using Wireshark: Display Filter

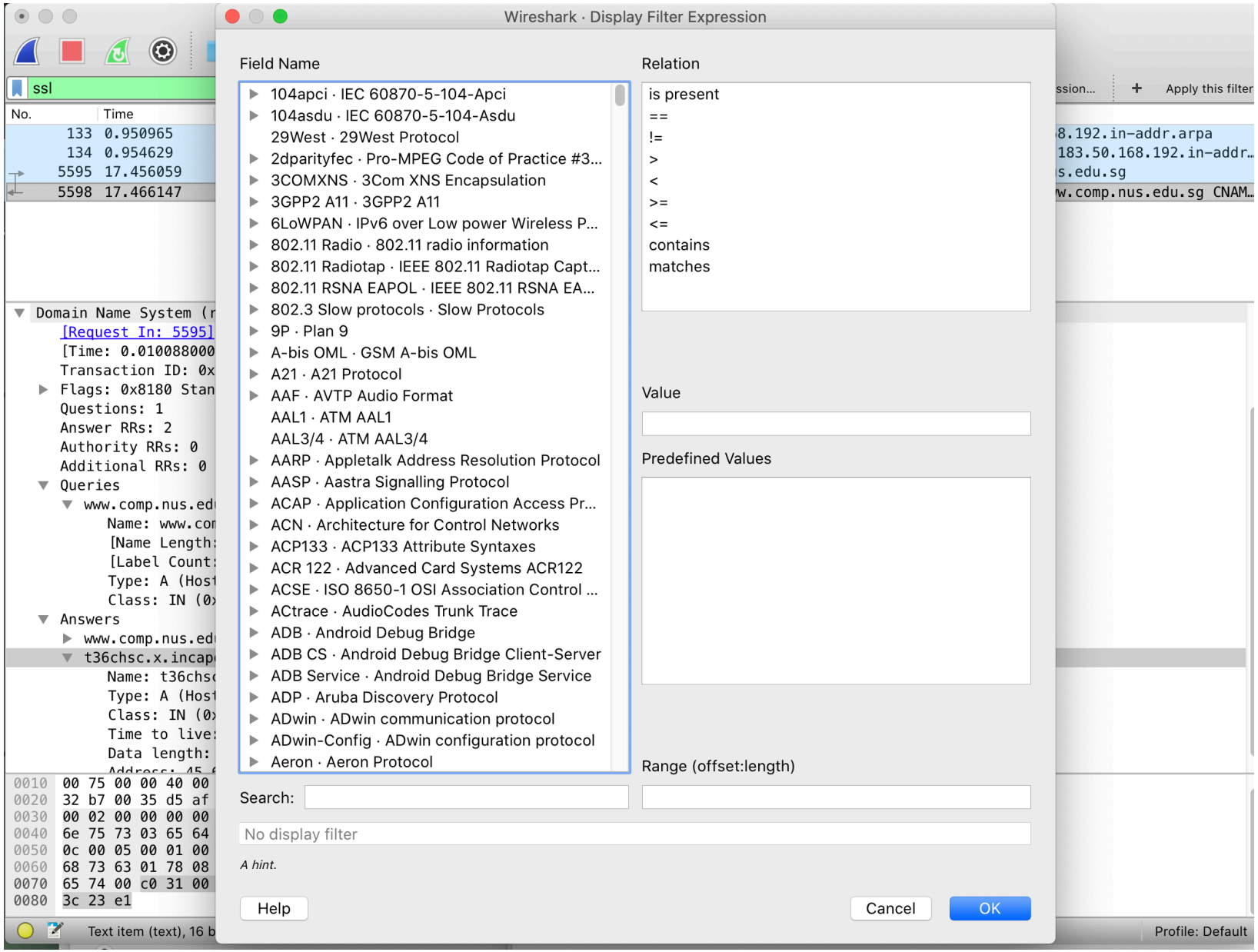
You need to specify a good **display filter**:

Filter comparison operators

Table 20. Display Filter comparison operators

English	C-like	Description and example
eq	==	Equal. `ip.src==10.0.0.5`
ne	!=	Not equal. `ip.src!=10.0.0.5`
gt	>	Greater than. `frame.len > 10`
lt	<	Less than. `frame.len < 128`
ge	>=	Greater than or equal to. `frame.len ge 0x100`
le	<=	Less than or equal to. `frame.len <= 0x20`
contains		Protocol, field or slice contains a value. `sip.To contains "a1762"`
matches	~	Protocol or text field match Perl regualar expression. `http.host matches "acme\.(org|com|net)"`
bitwise_and	&	Compare bit field value. `tcp.flags & 0x02`

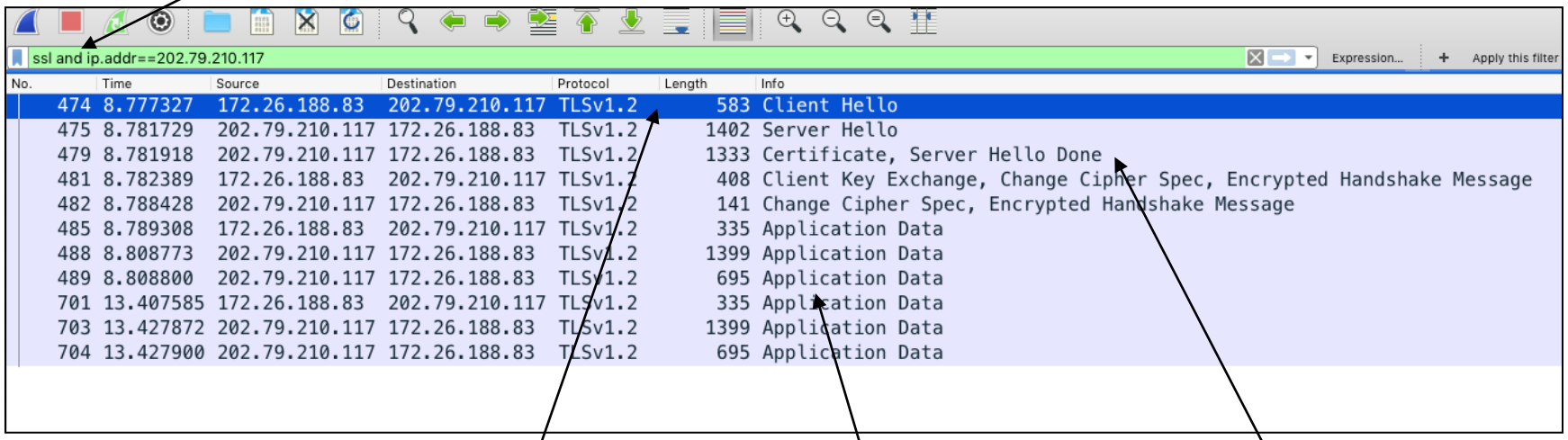
Sniffing using Wireshark: Display Filter



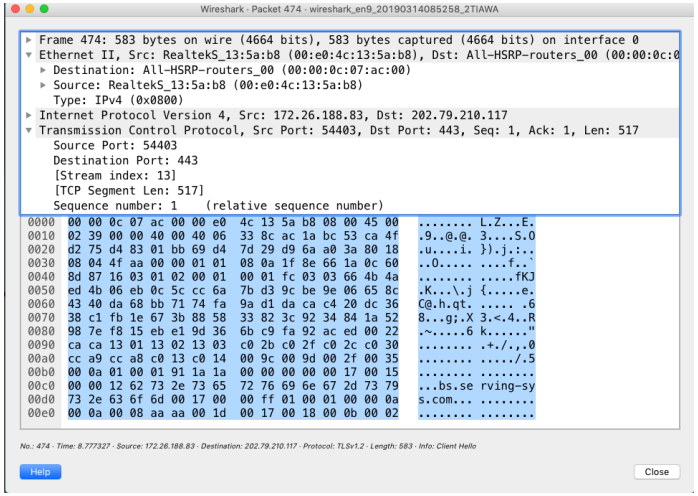
Display Filter on SSL/TLS

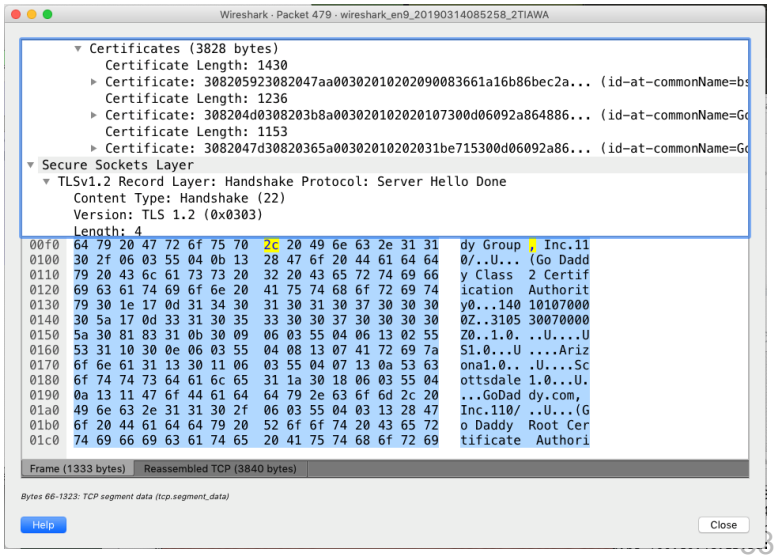
Display filter

Note: there are two types of filter in Wireshark: display and capture filters (don't get confused)



Encrypted application data





Sniffing using Wireshark: Popup Menu

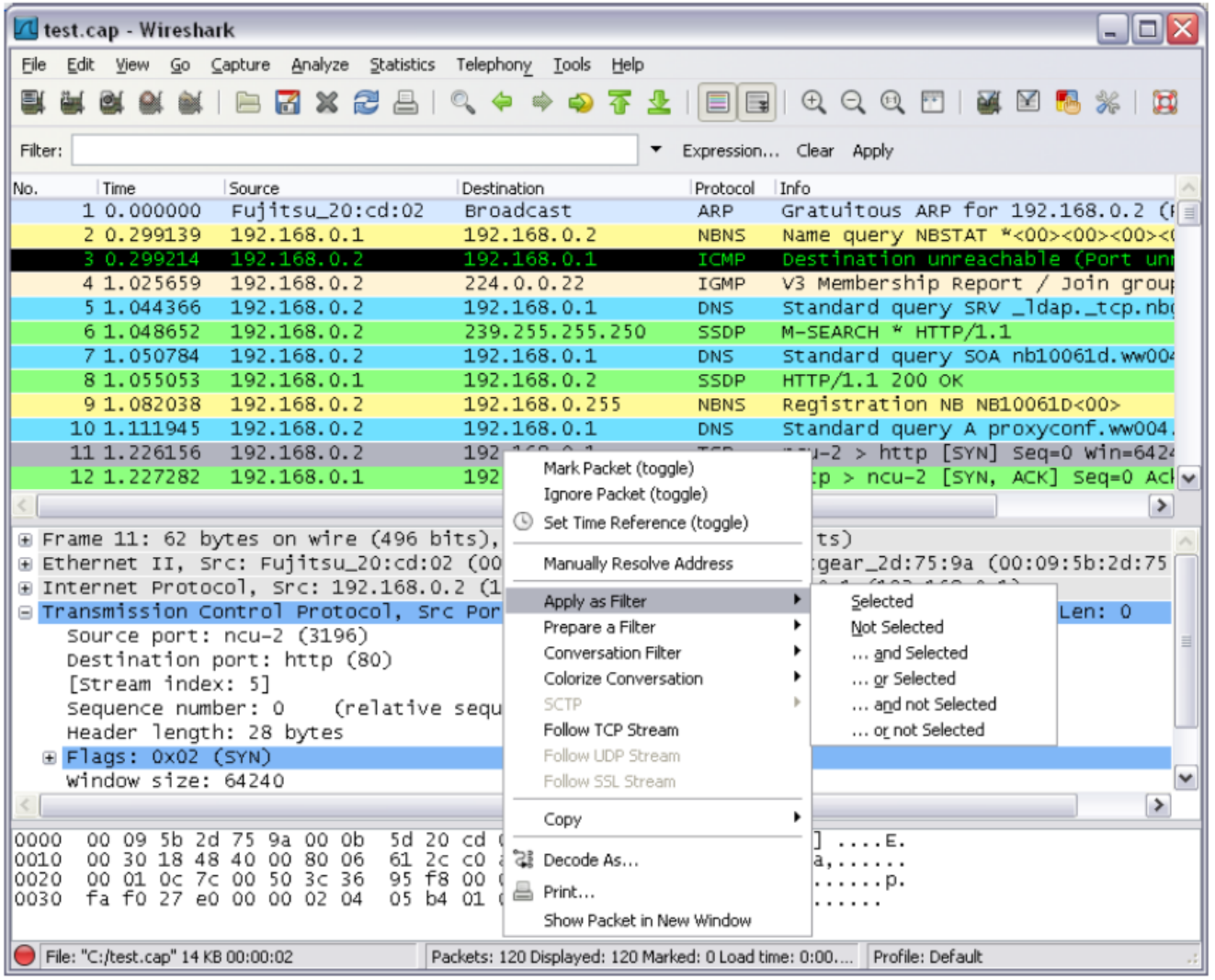


Figure 59. Pop-up menu of the "Packet List" pane

Source: Wireshark User's Guide

Sniffing using Wireshark: Follow TCP Stream

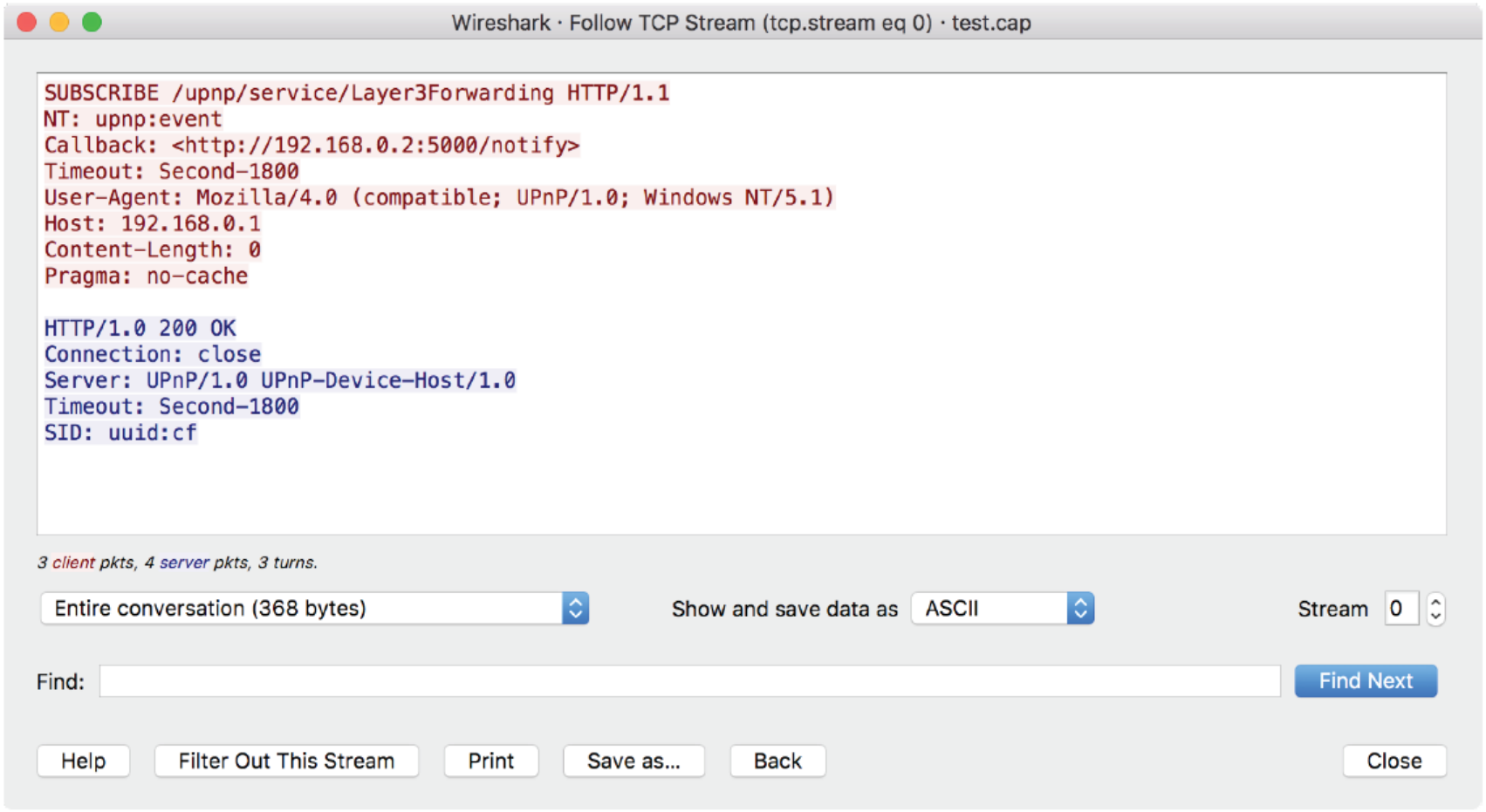


Figure 67. The "Follow TCP Stream" dialog box

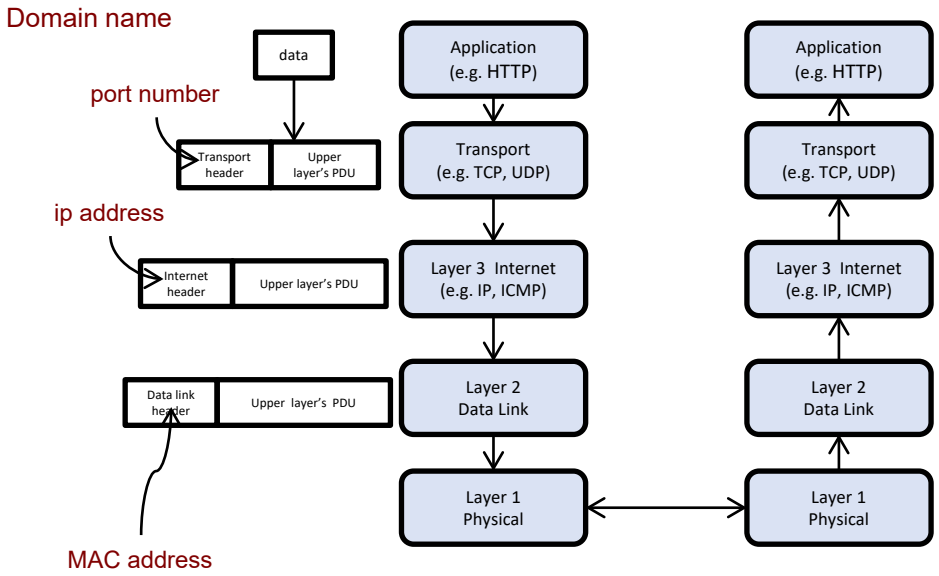
Source: Wireshark User's Guide

Demo Time!

Wireshark Demo

Nmap (Port Scanner)

- What is a ***port***?



- When a server receives an incoming packet, it will decide **which application process** to handle that packet based on the port no
- By saying that a process/service is "***listening***" to a particular port, we mean that the process **is running and ready to process** packets with that particular port no

Nmap (Port Scanner)

- When a port is "***open***", there exist such a process running in the server
- See **well-known port numbers**: https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers#Well-known_ports.
- ***Port scanning***: the process of determining which ports are open on hosts in a network
- Ports are "doors" into each machine, hence port scanning is like **knocking at the doors**
- ***Port scanner***:
  - A tool for performing port scanning
  - Is useful for both attacker and network administrator to scan for vulnerabilities
  - E.g. **Nmap** (very popular!)

Nmap (Port Scanner)

- **Nmap** is a full featured port-scanning tool:
  - Command-line tool, with GUI frontend
  - Installation:
    `sudo apt-get install nmap, zenmap`
  - Usage: `nmap` [Scan Type(s)] [Options] {target specification}
  - Examples:

    TCP ACK scan (a stealthier scan): `nmap -sA`

    OS fingerprinting: `nmap -O`

    Service/version detection: `nmap -sV`

Nmap: Sample Output

```
Nmap scan report
192.168.1.1 / somehost.com (online) ping results
address: 192.168.1.1 (ipv4)
hostnames: somehost.com (user)
The 83 ports scanned but not shown below are in state: closed
Port       State      Service  Reason       Product  Version  Extra info
21   tcp   open       ftp      syn-ack      ProFTPD  1.3.1
22   tcp   filtered   ssh      no-response
25   tcp   filtered   smtp     no-response
80   tcp   open       http     syn-ack      Apache   2.2.3    (CentOS)
106  tcp   open       pop3pw   syn-ack      poppassd
110  tcp   open       pop3     syn-ack      Courier pop3d
111  tcp   filtered   rpcbind  no-response
113  tcp   filtered   auth     no-response
143  tcp   open        imap     syn-ack      Courier Imapd       released
2004
443  tcp   open       http     syn-ack      Apache  2.2.3       (CentOS)
465  tcp   open       unknown  syn-ack
646  tcp   filtered   ldp      no-response
993  tcp   open       imap     syn-ack      Courier Imapd       released
2004
995  tcp   open                syn-ack
2049 tcp   filtered   nfs      no-response
3306 tcp   open       mysql    syn-ack      MySQL   5.0.45
8443 tcp   open       unknown  syn-ack
34 sec. scanned
1 host(s) scanned
1 host(s) online
0 host(s) offline
```

Demo Time (Again)!

Nmap Demo

Just for Fun, or If You are Curious

Optional

- *Question: Is port scanning illegal?*
- Read this if you want to use it: https://nmap.org/book/legal-issues.html

Figure 1.3. Strong opinions on port scanning legality and morality

Source: https://nmap.org/book/legal-issues.html

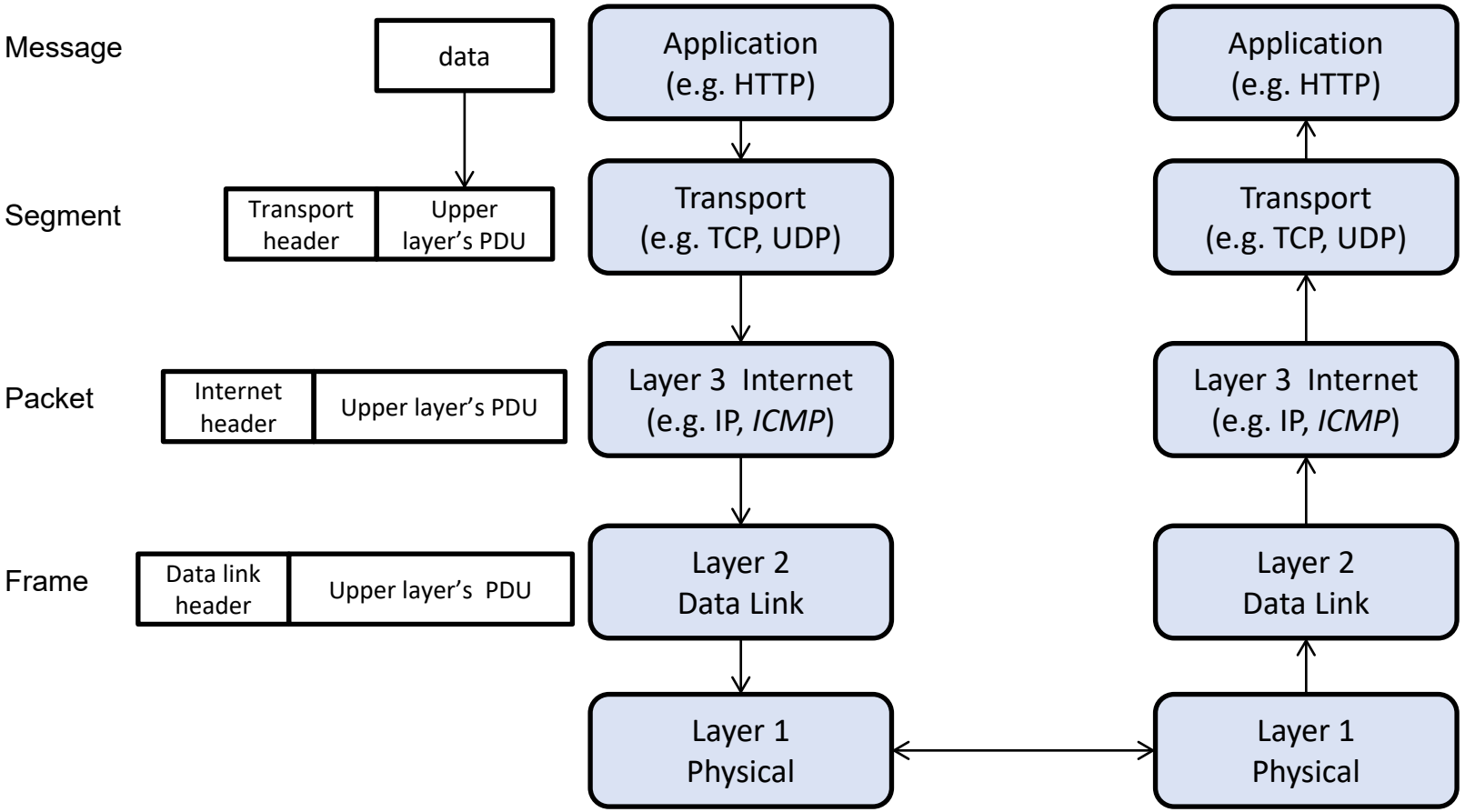
6.5 Protection: Securing the Communication Channel using Cryptography

Cryptography for Securing Network Communication

- Several cryptographic techniques to achieve **confidentiality** (encryption) and **authenticity** (MAC, PKI, strong authentication) over a public communication channel, even if the adversary can sniff & spoof data
- There are **many** security protocols that essentially achieve that, but operates at **different "layers"**
- Prominent **protocols**:
  - TLS/SSL
  - Wi-Fi Protected Access II (WPA2)
  - Internet Protocol Security (IPsec)

Security Protocols at Different Layers

- Recall the network layering shown previously
- TLS/SSL, WPA, IPSEC protect **different** layers



Remarks on Security Protocols and Network Layering

- Very often, when referring to a **security protocol**, we indicate the "**layer**" that the protocol **targets to protect**
- **Complication**: some protections *span across* multiple layers, or do *not* provide full protection of the targeted layer

- When analyzing an **attack**, it is also insightful to figure out at **what layer the attacker resides**
- **Complication**: likewise, some attacks *span across* multiple layers. In such situations, trying hard to pinpoint the layer could sometimes be very confusing

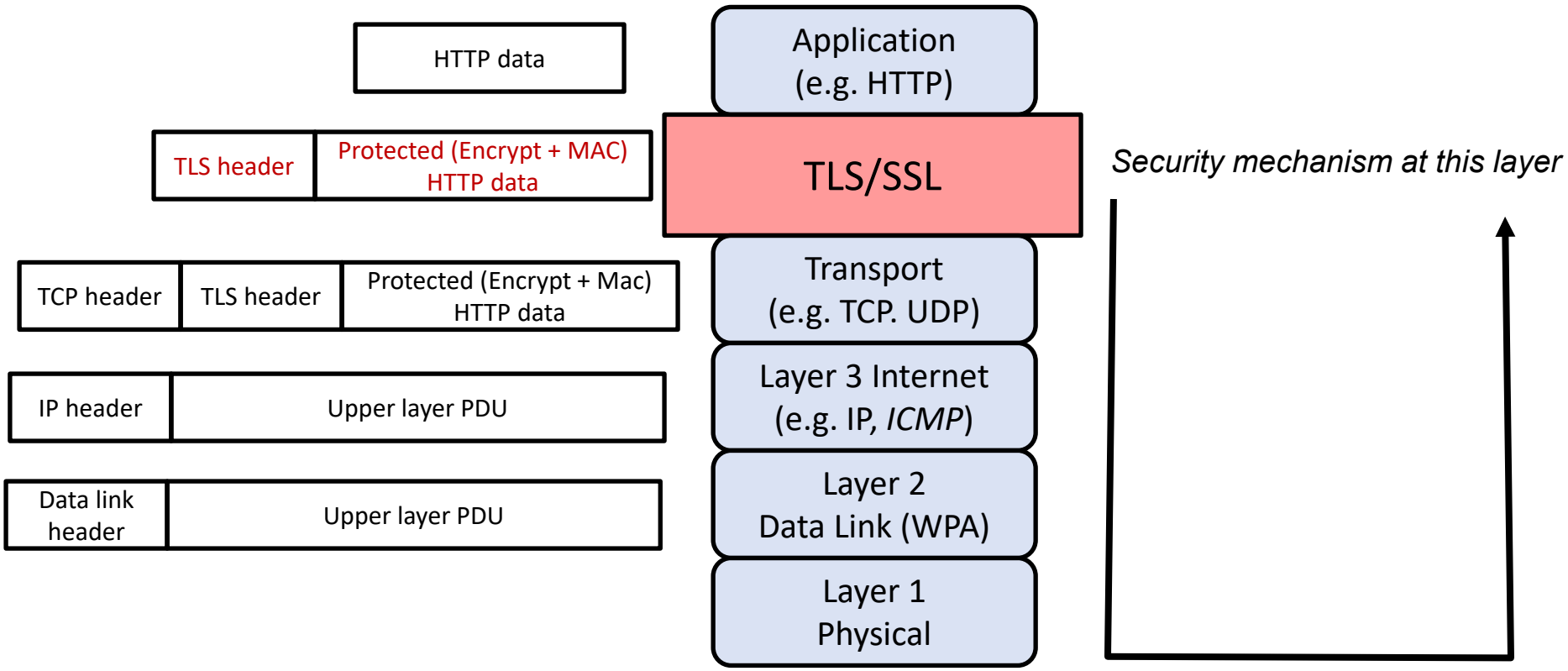
Remark on Security Protocols and Network Layering

- Below are the **general guideline** and example
- "A security protocol that **protects layer *k***
  would protect information from **that layer and above**
  against an attacker sitting at **layer *k*-1 and below**"
- **Example**: what happens if an attacker resides at layer 1,
  and there is a security protocol that protects layer 3*?*
- What is **protected** by the security protocol:
  the information generated in layer 3 and above
- What is ***not* protected**:
  the information generated in layer 2

1. SSL/TLS

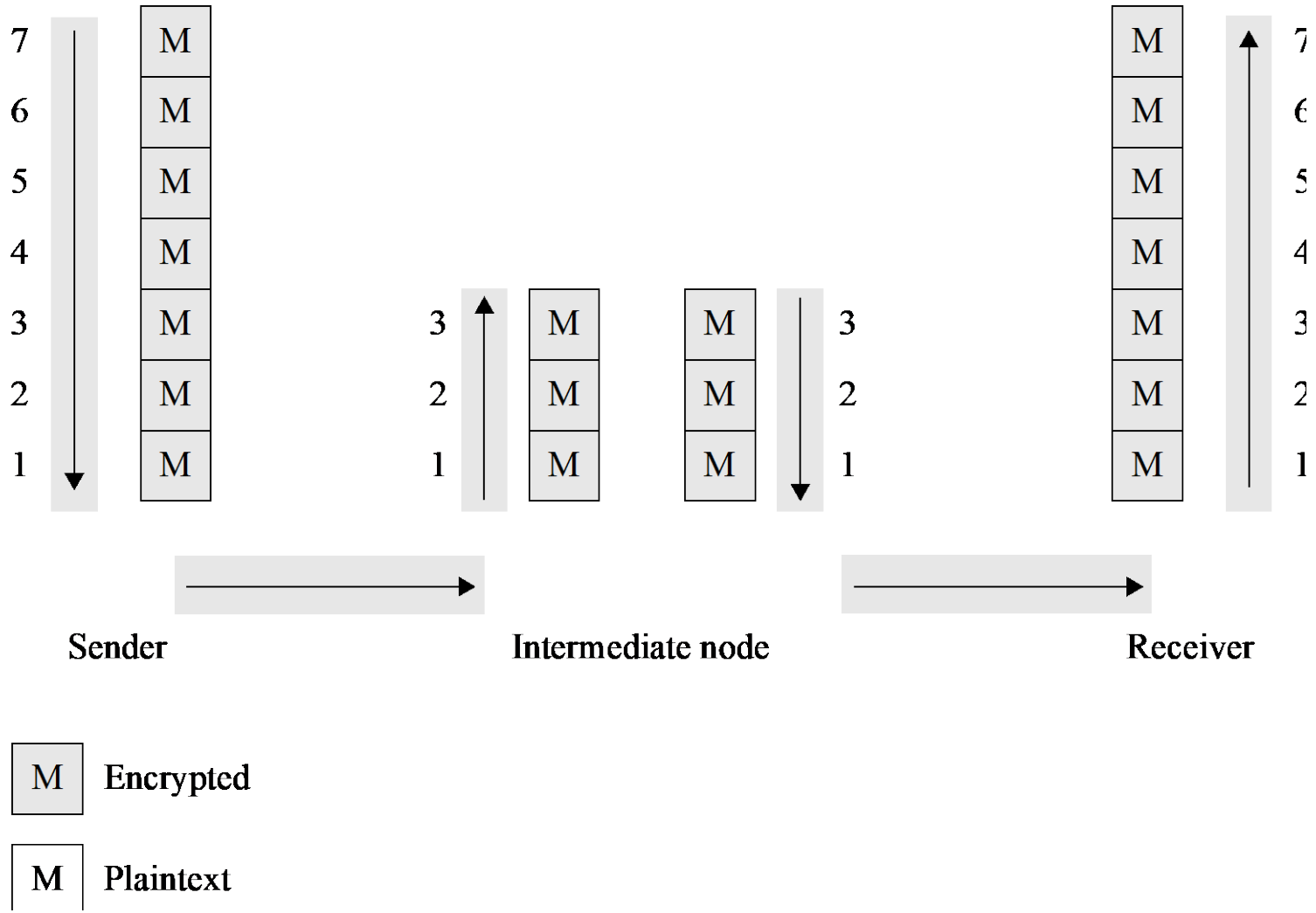
- The SSL/TLS sit on top of **transport layer**
- We can imagine that, when an application (e.g. browser or email agent) wants to send data to the other end point, it first pass the data and the destination IP address to SSL/TLS
- Next, SSL/TLS first “protects” the data using **encryption** (for confidentiality) and **MAC** (for authenticity), and then instructs the transport layer to send the protected data
- An **end-to-end encryption** is performed

SSL/TLS Location

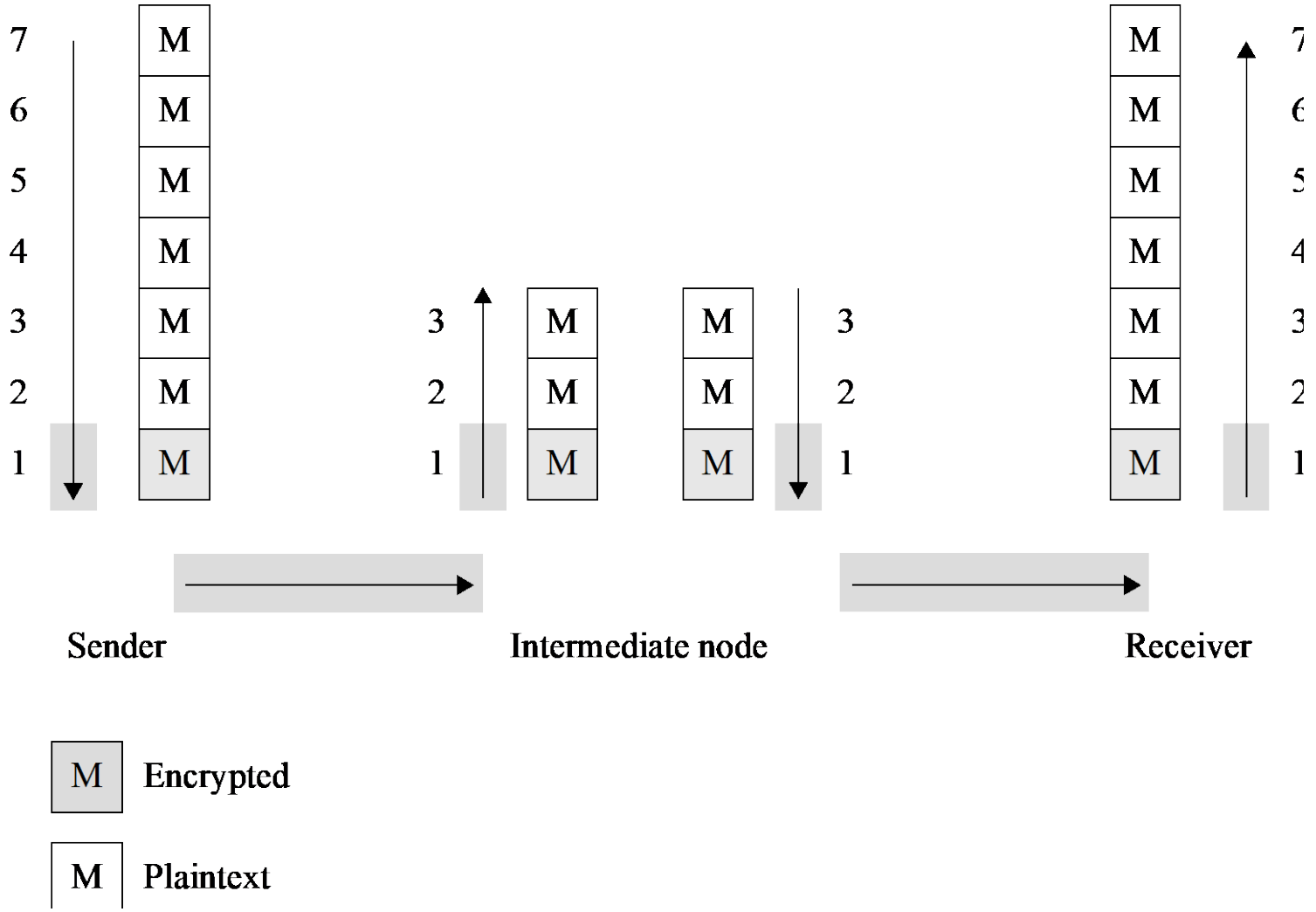


The receiver end-point **decrypts** the received data at the corresponding layer

Terminology: End-to-End Encryption



Terminology: Link (Hop-by-Hop) Encryption



Sample Usage Scenario

- Alice accesses LumiNUS web application to upload her report `a.pdf` to the LumiNUS server
- Note that LumiNUS uses **HTTPS**, which in turn employs **SSL/TLS**

Alice's machine carries the following:

1. The "LumiNUS client" passes the file `a.pdf` to HTTPS, and then to TLS
2. TLS protects the data by encryption and MAC
3. TLS passes the protected data to the transport layer

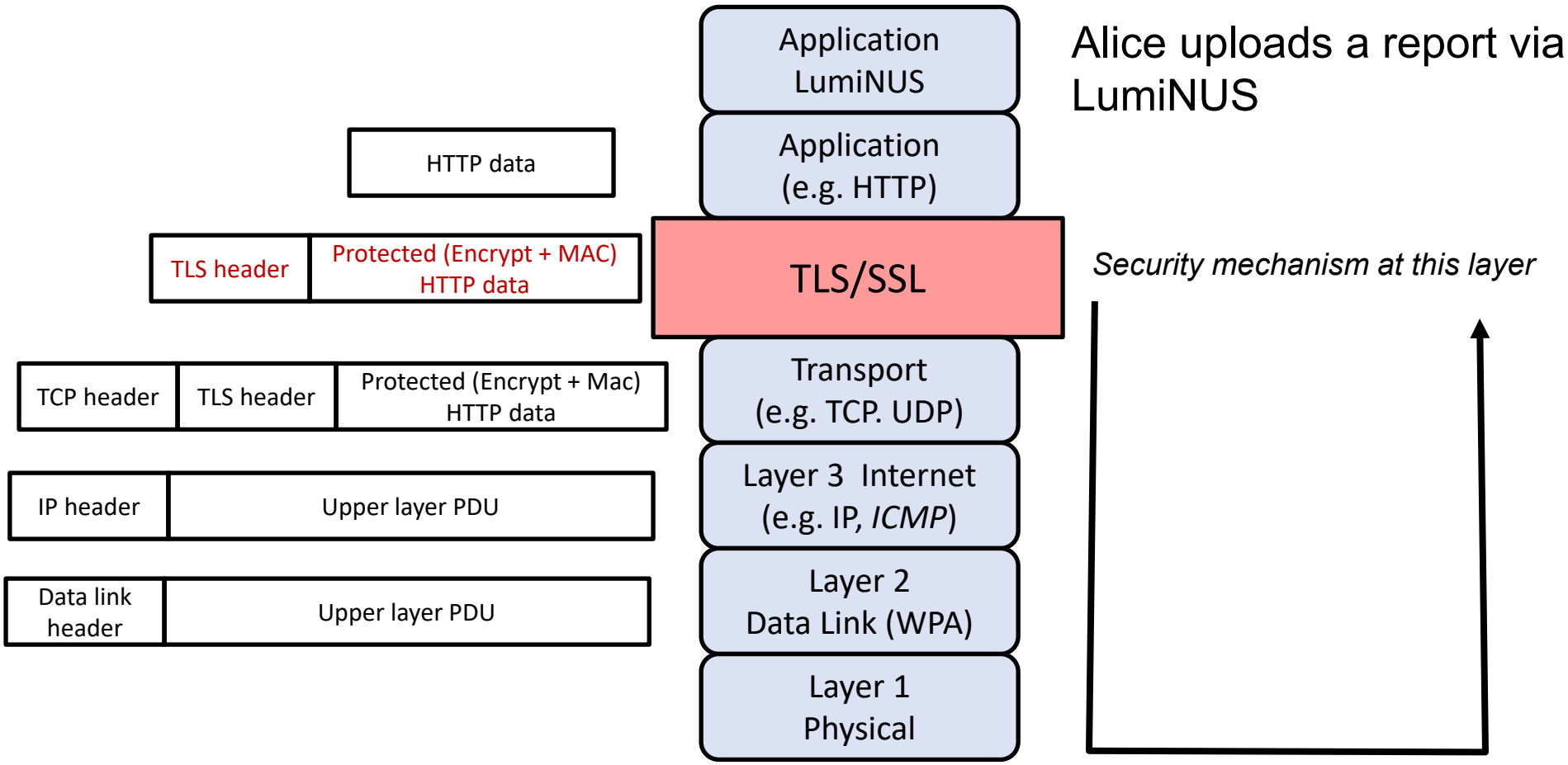
Sample Usage Scenario

The **LumiNUS server** carries out the following:

1. The transport layer passes the protected data to TLS
2. TLS decrypt the data and verify the MAC for integrity
3. TLS passes the decrypted data to LumiNUS application

- ***Remark***: Many details are omitted in the description. For instance, the “handshaking”, whereby the two parties establishing the session keys.

Sample Usage Scenario



The receiver end-point **decrypts** the received data at the corresponding layer

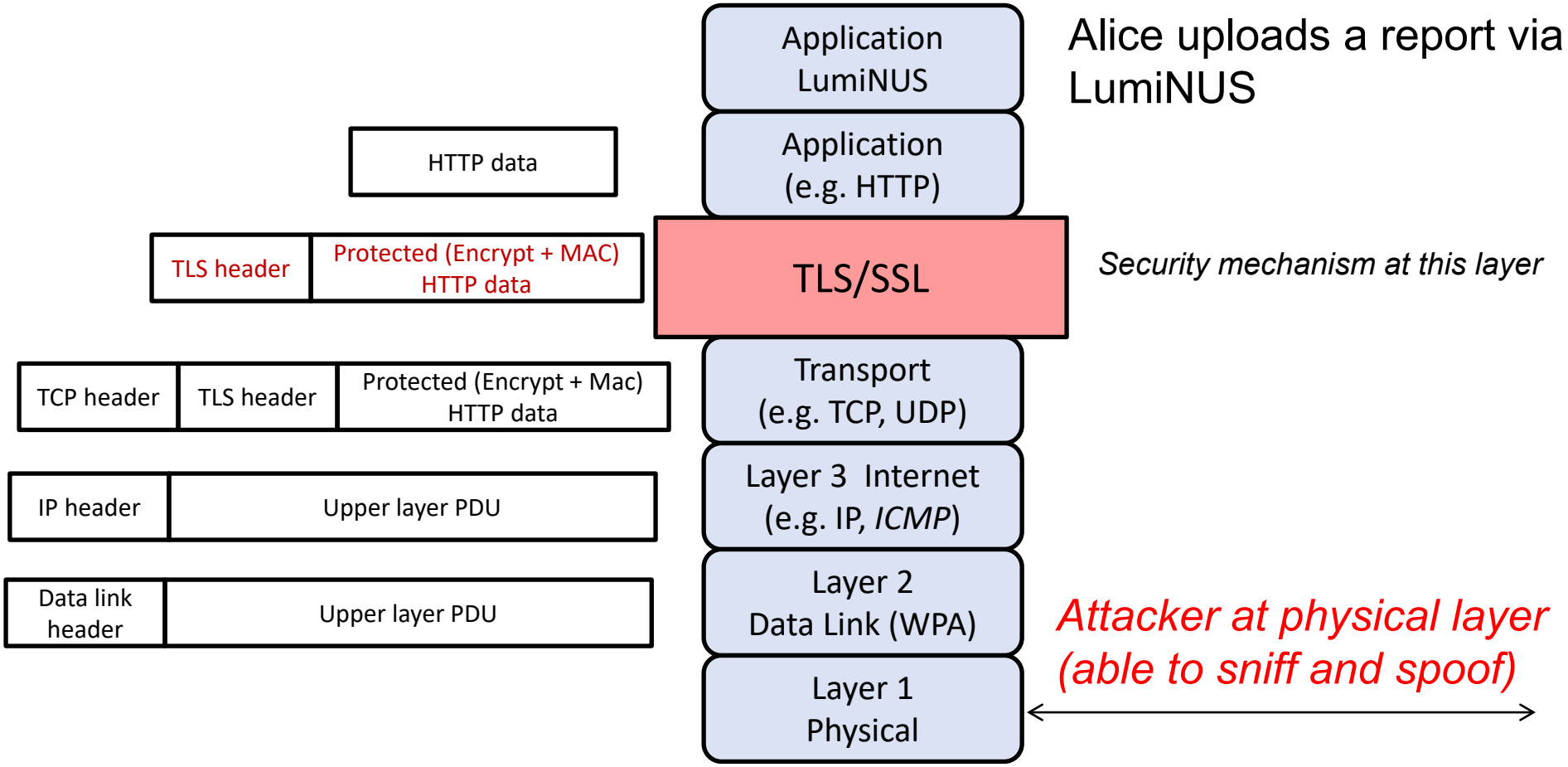
Attack Scenario 1-1: SSL/TLS with Attacker at Physical Layer

- Suppose that there is an attacker **at the physical layer**, who can sniff and spoof message at that layer
- **For example**, Alice uploads her report in a cafe using a free/open WiFi (without WPA protection).
  Hence, anyone in the café has access to the physical layer, and thus can sniff and spoof messages in that layer.

Question: Can the attacker learn:

1. Alice's uploaded report?
2. The fact that Alice is visiting LumiNUS website (i.e. can the attacker learn the website's IP address)?

Attack Scenario 1-1: SSL/TLS with Attacker at Physical Layer



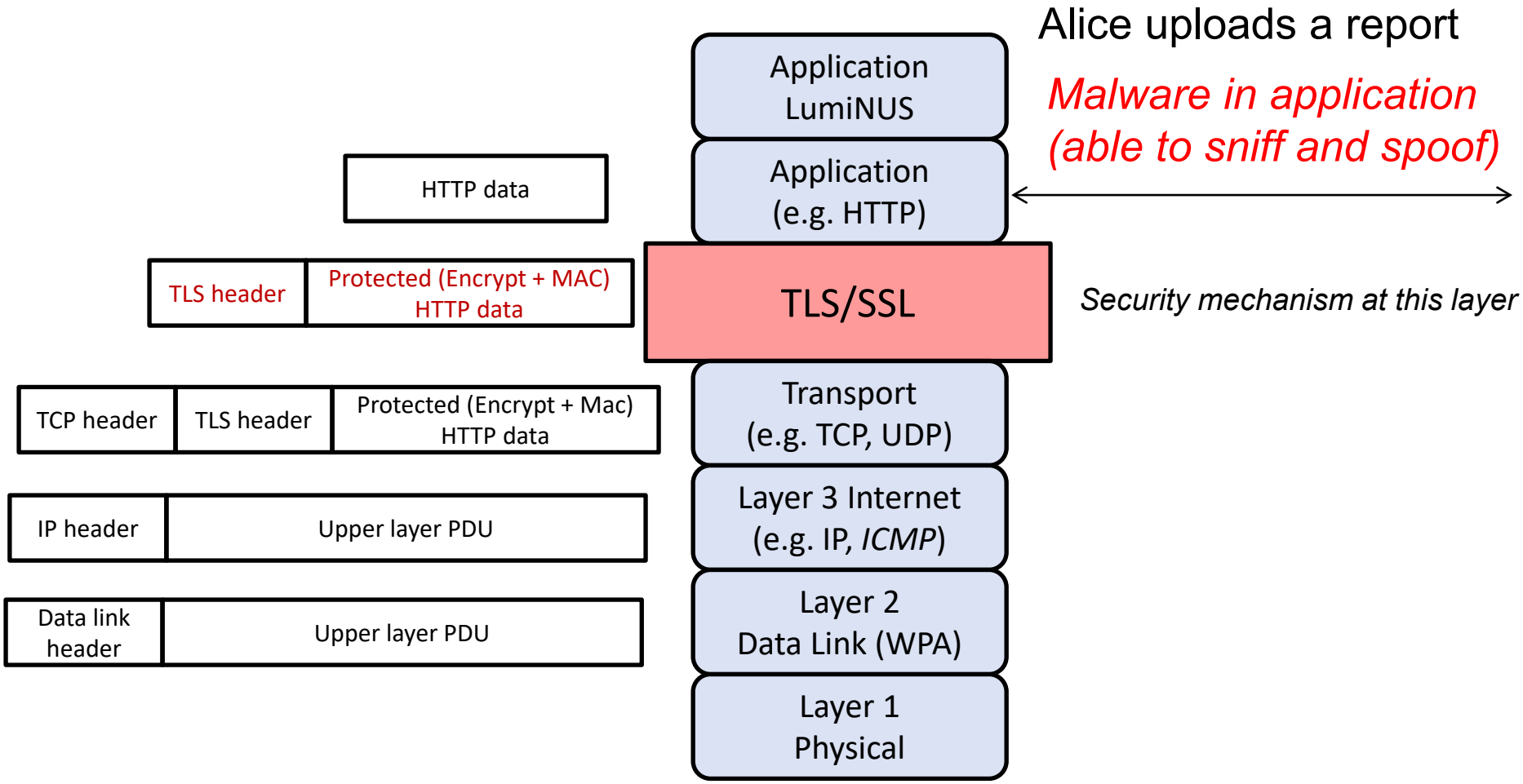
Attack Scenario 1-2: SSL/TLS with Attacker at Application Layer

- Suppose that there is an adversary at the **application layer**
- For example, a **malicious JavaScript** is injected into LumiNUS and being executed by Alice's **browser** → "*Man in the Browser (MitB)*"

Question: Can the malicious script learn:

1. Alice's report?
2. *Alice's MAC address?* (an interesting issue)

Attack Scenario 1-2: SSL/TLS with Attacker at Application Layer



2. WPA2

WiFi Protected Access II (WPA2):

- A popular protocol employed in home WiFi access point
- More secure than WEP (broken), WPA

The protections provided:

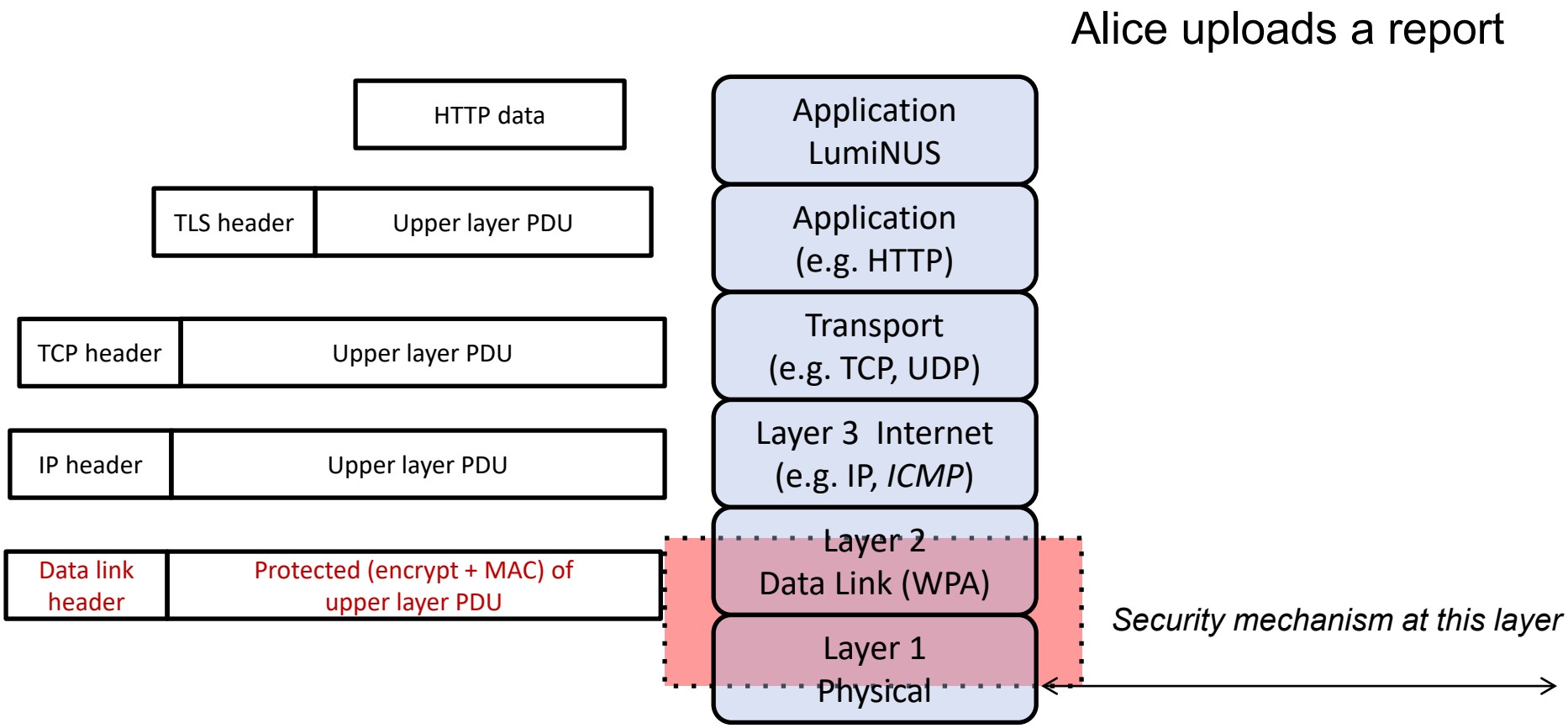
- WPA2 provides protection at **layer 2 (link) and layer 1 (physical)**
- Note: *not* all information in layer 2 are protected

See a recent attack on WPA2:

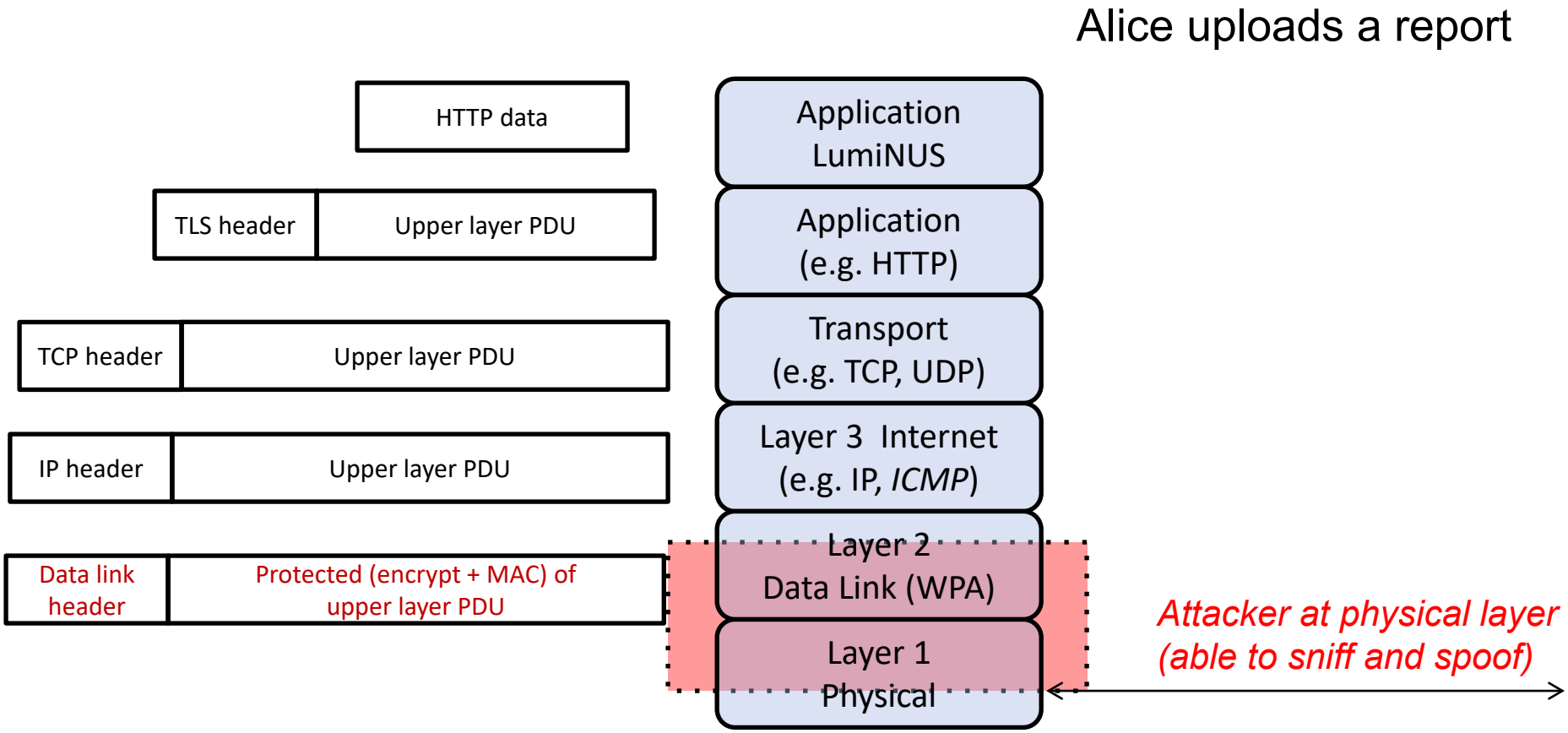
Key Reinstallation Attacks: Breaking WPA2 by forcing nonce reuse
https://www.krackattacks.com/

The research paper: https://papers.mathyvanhoef.com/ccs2017.pdf

WPA2 and Network Layers



Attacker at Physical Layer

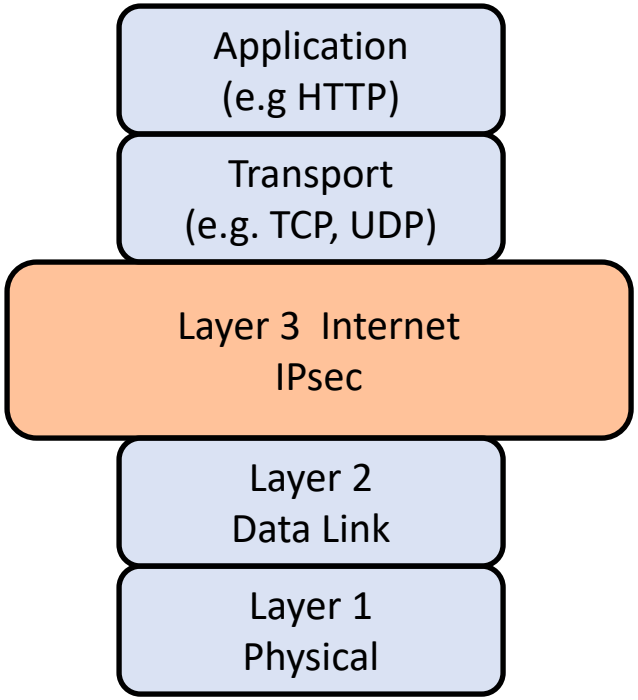


Question: Can the attacker learn:

1. *Alice's report?*
2. *The fact that Alice is visiting LumiNUS website?*
3. *The MAC address (link layer)?* not clear

3. IPsec

- IPsec provides **integrity/authenticity** protection of IP addresses, but ***not*** **their confidentiality:**
  - Hence, attackers are unable to “spoof” the source IP address
  - But they can still learn the source and destination IP addresses of the sniffed packets



Remarks on IPsec

IPsec is a mechanism whose goal is **to protect the IP layer**

Its description:

- "**Internet Protocol Security** (**IPsec**) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used in protecting data flows between a pair of hosts (*host-to-host*), between a pair of security gateways (*network-to-network*), or between a security gateway and a host (*network-to-host*).[1]
- Internet Protocol security (IPsec) uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection.
- IPsec is an end-to-end security scheme operating in the Internet Layer of the Internet Protocol Suite, while some other Internet security systems in widespread use, such as Transport Layer Security (TLS) and Secure Shell (SSH), operate in the upper layers at Application layer. Hence, only IPsec protects any application traffic over an IP network. Applications can be automatically secured by IPsec at the IP layer."

-Wiki

Question

Question: Explain the underlined sentences

- "**Internet Protocol Security** (**IPsec**) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used in protecting data flows between a pair of hosts (*host-to-host*), between a pair of security gateways (*network-to-network*), or between a security gateway and a host (*network-to-host*).[1]
- Internet Protocol security (IPsec) uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection.
- IPsec is an end-to-end security scheme operating in the Internet Layer of the Internet Protocol Suite, while some other Internet security systems in widespread use, such as Transport Layer Security (TLS) and Secure Shell (SSH), operate in the upper layers at Application layer. Hence, only IPsec protects any application traffic over an IP network. Applications can be automatically secured by IPsec at the IP layer."

-Wiki

6.6 Protection: Firewall

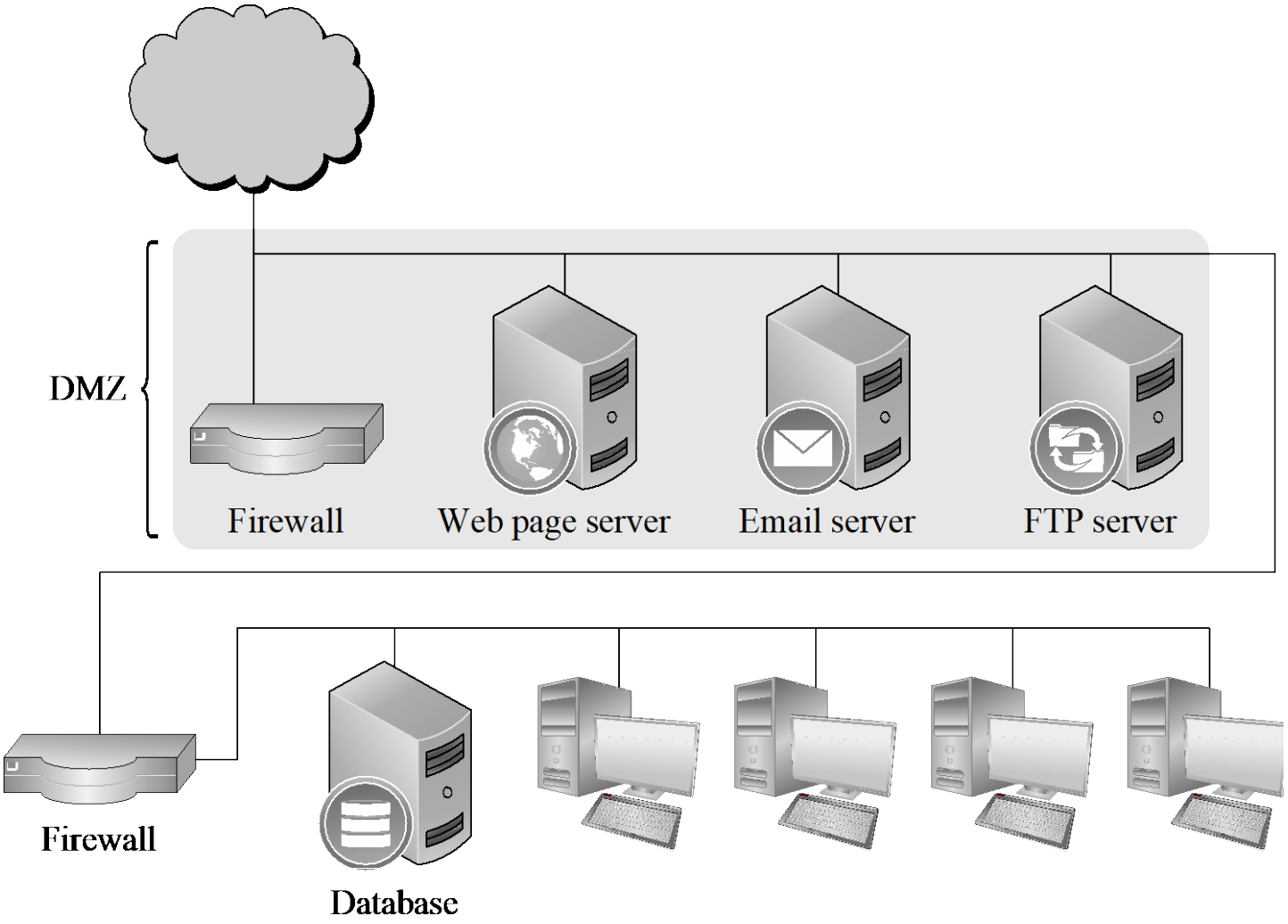
Motivation

- Having SSL/TLS and WPA2 is still ***not*** sufficient to protect the network:
  - There are concerns of **DoS** that they can't prevent
  - Many services and applications **can't be protected** by SSL/TLS & WPA2: e.g. **DNS spoofing**
    (It is not practical, due to efficiency, to establish SSL/TLS to the DNS server for DNS query)
- There is a need to control **the flow of traffic** between networks, especially between the ***untrusted*** public network (Internet) and the ***trusted*** internal network
- Even within the internal network, we still need to divide it into different **network segments** and deny unnecessary access: **Principle of least privilege, compartmentalization**

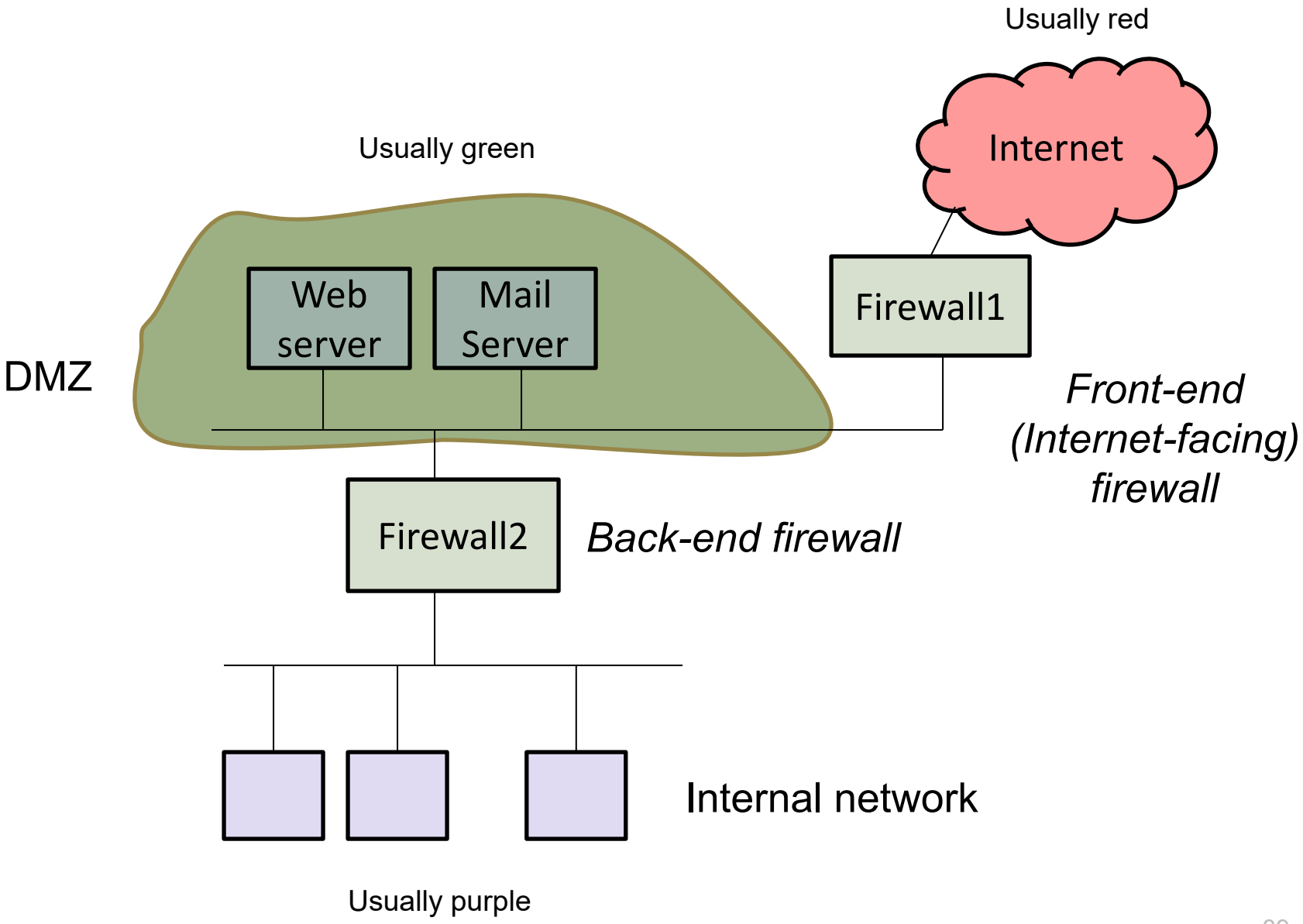
Firewall

- ***Firewall***:
  - Sits at border between networks
  - Looks at addresses, services, other characteristics of traffic
  - Controls what traffic is allowed to enter the network (ingress filtering), or leave the network (egress filtering)
- **Definition**: *"Firewall are devices or programs that control the flow of network traffic between networks or hosts that employ differing security postures."*
  (From "Guidelines on Firewalls and Firewall Policy", NIST, special publication 800-41 http://csrc.nist.gov/publications/nistpubs/800-41-Rev1/sp800-41-rev1.pdf.)
- ***Demilitarized Zone* (*DMZ*)**:
  - A small **sub-network** that exposes the organization's external service to the (untrusted) Internet
  - The original military term: an area between states in which military operations are not permitted

Demilitarized Zone (DMZ)



Typical 2-Firewall Setting



Firewall Design (Read [PF] pg 453)

- A firewall enforces a **set of rules** provided by the network administrator
- Examples of rules for Firewall2 (back-end firewall):
  - Block HTTP
  - Allow from Internal network to Mail Server: SMTP, POP3
- Examples of rules for Firewall1 (front-end firewall):
  - Allow from anywhere to Mail Server: SMTP only
- How the rules are to be specified
  differs on different devices and software
- The next slide gives a typical design/configuration
  (from [PF] pg. 453)

Sample Firewall Configuration

Rule No	Protocol Type	Source Address	Destination Address	Source Port	Designation Port	Action
1	TCP	*	192.168.1.*	*	25	Permit
2	TCP	*	192.168.1.*	*	69	Permit
3	TCP	192.168.1.*	*	*	80	Permit
4	TCP	*	192.168.1.18	*	80	Permit
5	TCP	*	192.168.1.*	*	*	Deny
6	UDP	*	192.168.1.*	*	*	Deny
n	*	*	*	*	*	Deny

Matching condition

* (means “any”), which matches **any** values

The table is processed in a **top-down manner**

The **first matching rule** determines the **action taken**

Hence: put your most specific rule *first*, and put your most general rule *last*

Types of Firewall

The textbook [PF] lists **6** types of firewalls

The literature, including NIST's document (NIST 800-41), usually groups firewalls into **3 types**:

1. **(*Traditional*) *packet filters***:
   - Filters packets based on information in **packet headers**
2. ***Stateful inspection***:
   - Maintains a ***state table*** of all active connections
   - Filters packets based on **active connection states**
3. ***Application proxy***:
   - Understands application logic
   - Acts as a relay of application-level traffic

(Details are not required for this module)

Comparison of Firewall Types [PF]

Optional

Packet Filter	Stateful Inspection	Application Proxy	Circuit Gateway	Guard	Personal Firewall
Simplest decision-making rules, packet by packet	Correlates data across packets	Simulates effect of an application program	Joins two subnetworks	Implements any conditions that can be programmed	Similar to packet filter, but getting more complex
Sees only addresses and service protocol type	Can see addresses and data	Sees and analyzes full data portion of pack	Sees addresses and data	Sees and analyzes full content of data	Can see full data portion
Auditing limited because of speed limitations	Auditing possible	Auditing likely	Auditing likely	Auditing likely	Auditing likely
Screens based on connection rules	Screens based on information across multiple packets—in either headers or data	Screens based on behavior of application	Screens based on address	Screens based on interpretation of content	Typically, screens based on content of each packet individually, based on address or content
Complex addressing rules can make configuration tricky	Usually preconfigured to detect certain attack signatures	Simple proxies can substitute for complex decision rules, but proxies must be aware of application's behavior	Relatively simple addressing rules; make configuration straightforward	Complex guard functionality; can be difficult to define and program accurately	Usually starts in mode to deny all inbound traffic; adds addresses and functions to trust as they arise

6.7 Protection: Network Security Management

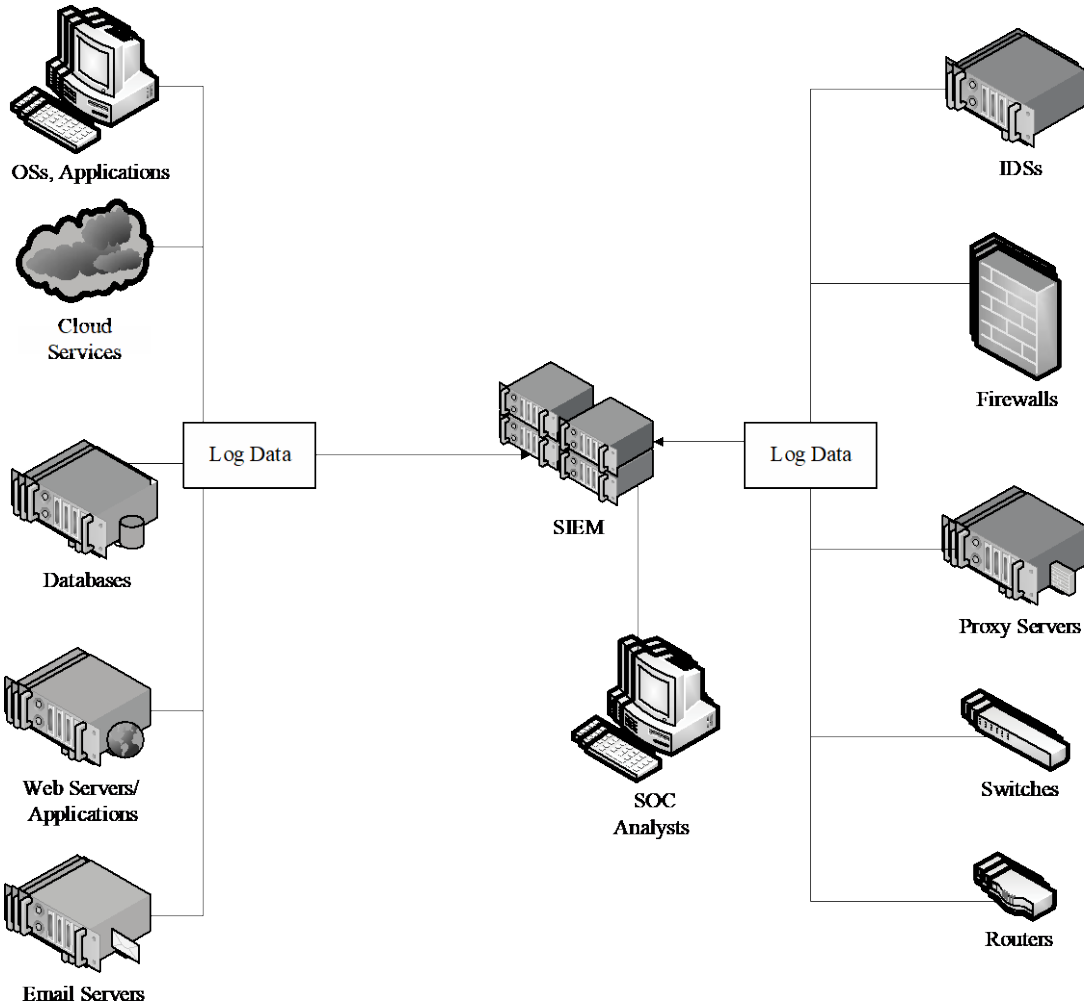
Network Security Management

There is a need to **continuously** monitor and adjust network characteristics
(Details on this are omitted. See [PF6.9])

Some terms:

- ***Security Operations Center* (*SOC*)**:
  - A centralized unit in an organization that monitors the IT systems and deals with security issues
- ***Security Information and Event Management* (*SIEM*)**:
  - Pronounced as “SIM”
  - Provides **real-time analysis** of security alerts generated by network hardware and applications
  - May include the following **capabilities**:
    data aggregation & correlation, event alerting,
    compliance report generation, forensic analysis

Security Information and Event Management (SIEM)



Summary & Takeaways

- Even with cryptography securing an end-to-end communication channel, there are many other issues, e.g. network security issues
- Network layering
- Name-resolution issues (DNS attack, ARP attack)
- DoS & DDoS attacks
- Protection at various layers (WPA, TLS, IPsec)
- Access control (firewall) and monitoring (SOC, SIEM)
- Some useful tools: Wireshark, Nmap