Topic 5: Network Security

- Background
  - Layering
  - Port listening
- MITM positions in the network
- DNS, ARP, DoS attacks
- Securing the channel using Cryptography
  - TLS, IPSec, WPA2, VPN
  - Firewalls and IDS
- Useful tools (self explore)

1 Recap

Recap

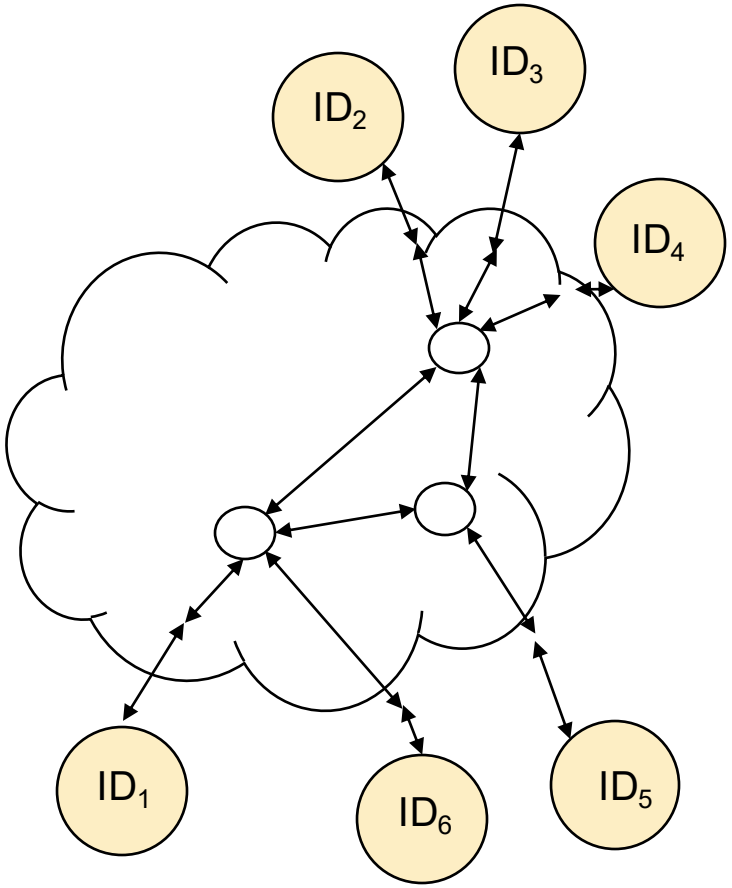
- Cryptographic techniques + PKI, we can ***secure a channel*** to achieve authenticity & confidentiality, even in the presence of a Mallory.
- **Narrow focus**:
  - Emphasizes securing a specific communication channel ***between two endpoints*** (e.g., server and client).
  - It is over-simplified in a network with ***large number of entities***.
- **Broad scope**: focuses on protecting an entire network or segments of a network - Routers, switches, firewalls, etc
- This leads to next topic: Network Security.

2 Background: Networking

Computer Network

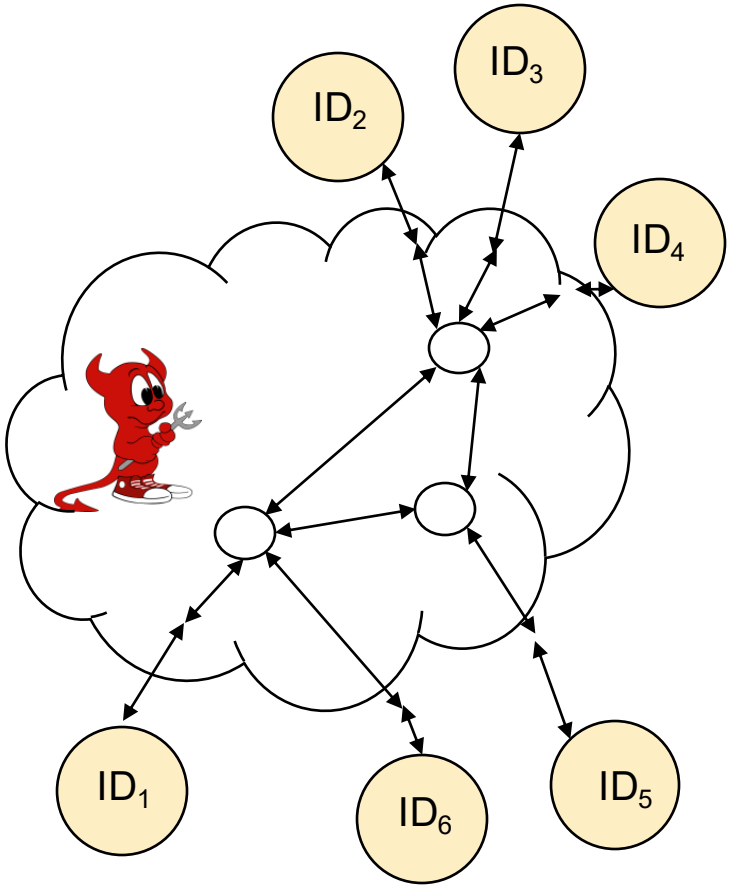
- **Computer network**: a collection of ***interconnected devices*** (e.g., computers, servers, routers, switches) that can communicate with one another

- **Packet switching:** To transmit data over the network, instead of having dedicated line between any two nodes, packet switching is deployed:
  - Messages are broken into “packets/frames”.
  - Messages are “routed” via multiple switches and routers.



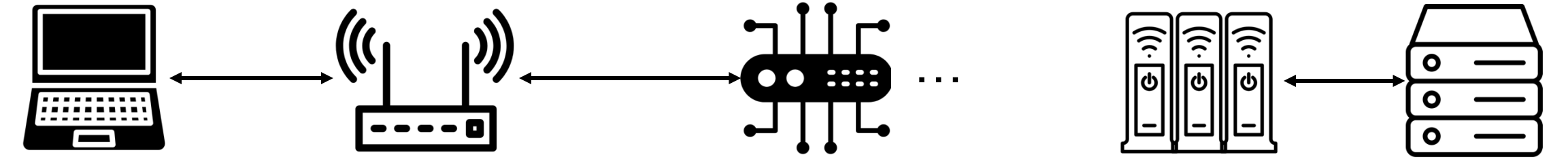
Network Security

- Networking focuses on ***how to route messages*** via the intermediate nodes.
- In contrast, network security focuses on the ***attackers among the intermediate nodes***.
- An attacker wants to steal, modify, disrupt
  - Attackers modify the routing information so that traffic being routed to wrong places, or to nowhere.
  - Attackers guess who-is-talking-to-who.



Multiple Hops

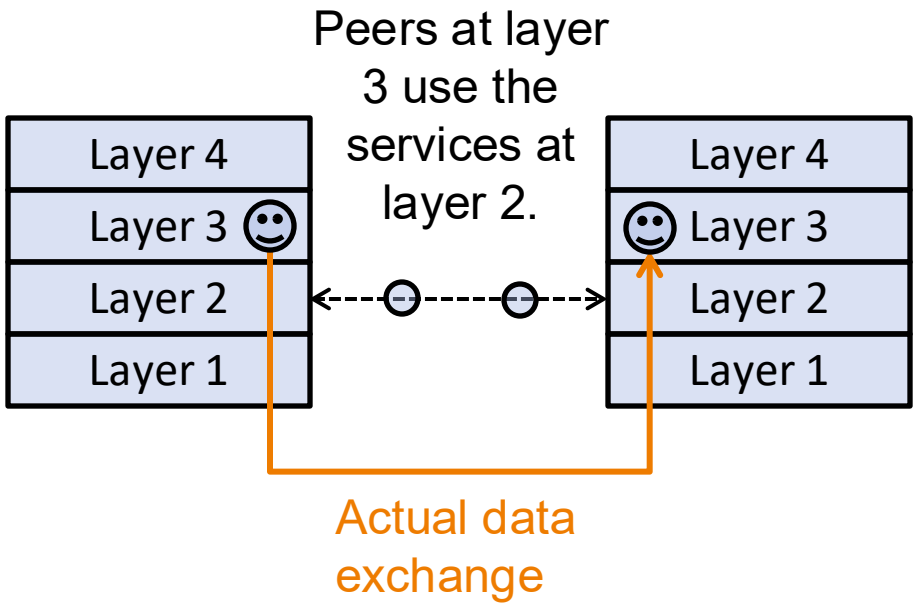
- Suppose Alice is using her laptop at home to browse the website `bbc.com`.
- The data go through multiple hops via intermediate nodes.
- At each intermediate node, routing data are being read and modified
  - Use "`traceroute`" command to see the hops.



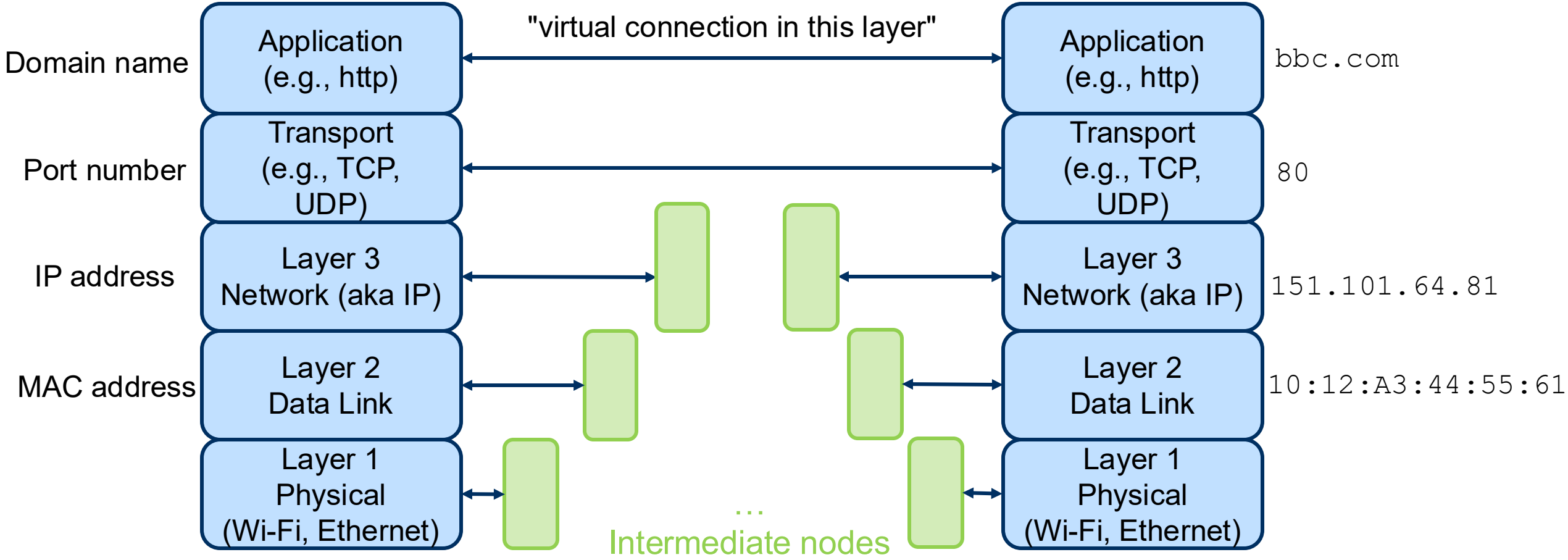
the images are by Singlar, zaenul yahya, Naba A'la Lail on The Noun Project

Network Layers

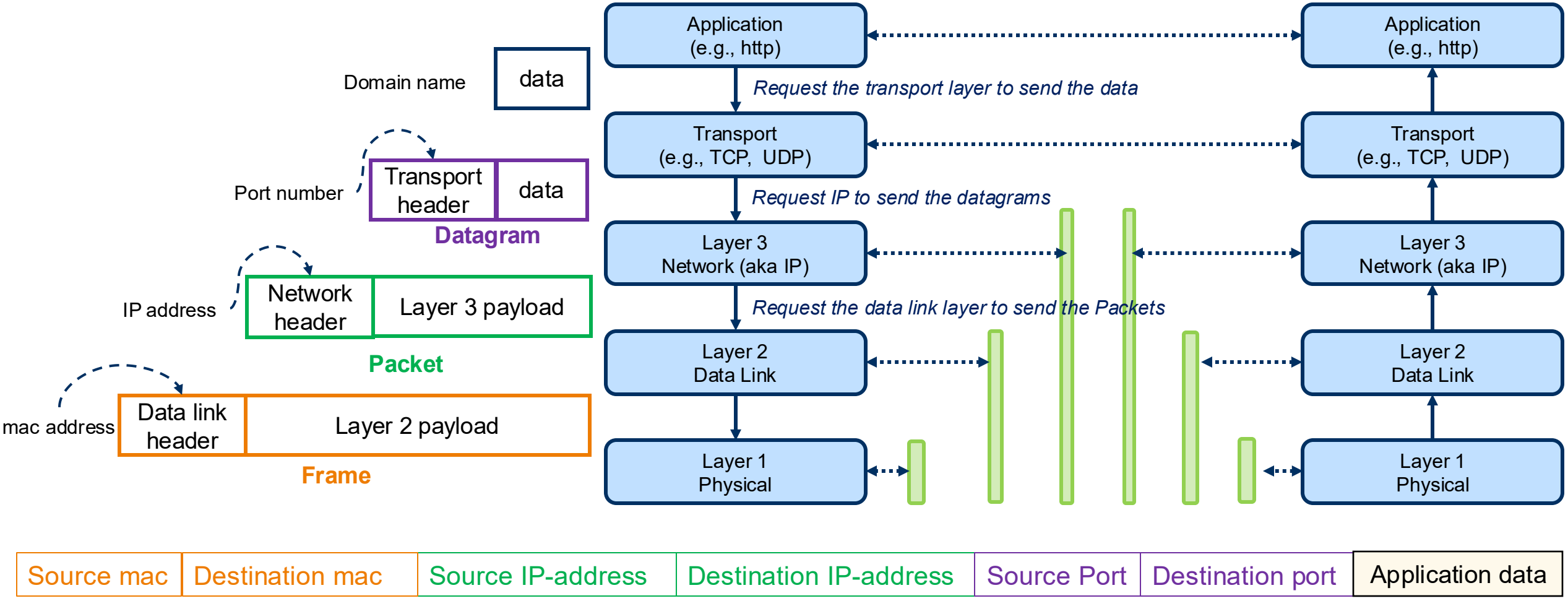
- How to organize the structure of the network?
- **Network Layering**:
  - **Modularity**: each layer can be developed and modified independently.
  - **Abstraction**: a specific set of functions and hides the underlying complexities from the layers above.
  - **Isolation**: potential issues or failures can be contained and addressed within a specific layer
- Conceptually, virtual connection between layers at different host



Internet layers (TCP/IP Model):

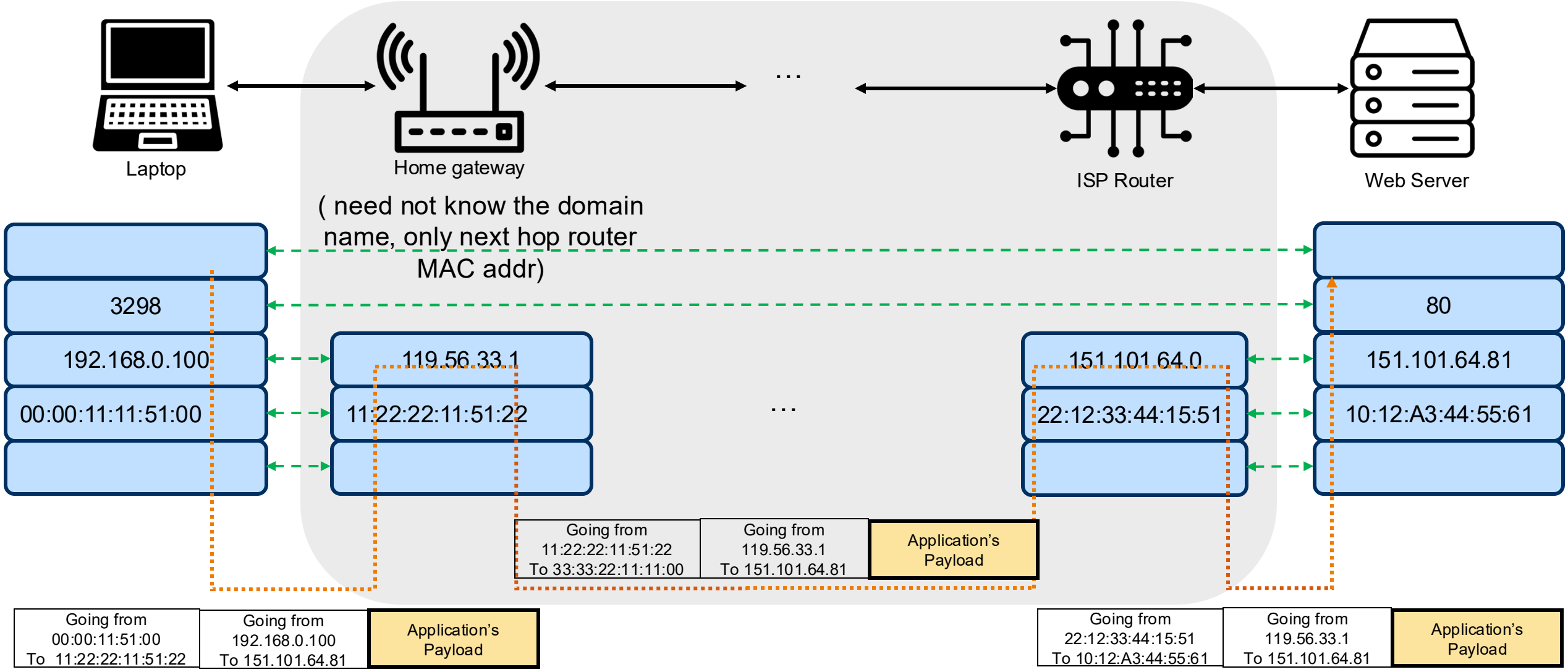


Data Flow Across the Layers Between Two Nodes



Source mac	Destination mac	Source IP-address	Destination IP-address	Source Port	Destination port	Application data

Multiple Hops and Network Layers

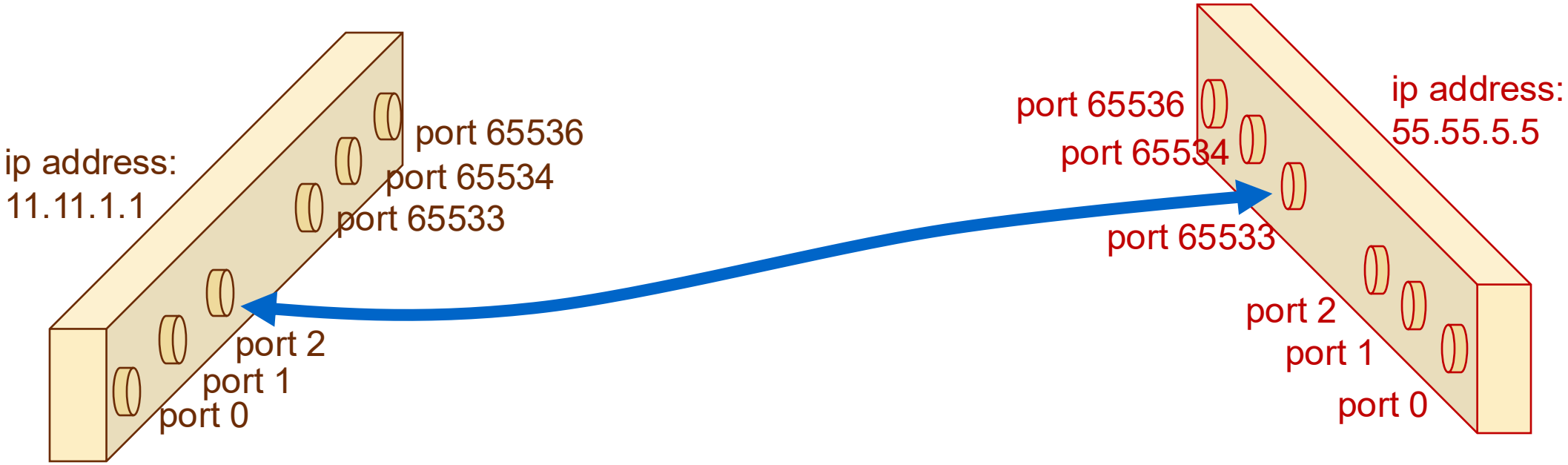


Remark: The ip-address might be translated (gateway change 192.168.0.100 to 119.56.33.1) as shown in this example. In this class, for simplicity, we don't consider translation.

Transport + IP layer

- Very often, the transport layer and IP layer are treated as one single layer.
- In this combined layer, the address of a communicating entity is an ip-address and a port - `151.101.64.81:80`
- Each node in the network has a total 65535 ports.
- A communication channel between two nodes is established by connecting two ports.
- Two protocol in transport layer:
  - **TCP** (Transmission Control Protocol): connection-oriented, reliable data transfer
  - **UDP** (User Datagram Protocol): connectionless, non-reliable data transfer

Comm Channel: 11.11.1.1:2 and 55.55.5.5:65533.

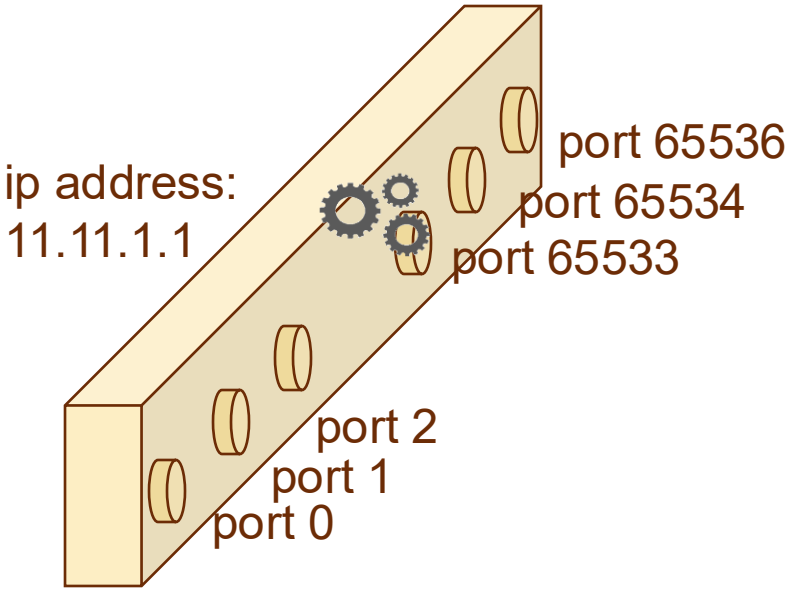


Going from left to right:
src ip: 11.11.1.1
src port: 2
dest ip: 55.55.5.5
Dest port: 65533

Going from right to left:
src ip: 55.55.5.5
src port: 65533
dest ip: 11.11.1.1
Dest port: 2

What do we mean by “listening to a port”, “closed port”?

- We can imagine that behind certain ports (e.g., 65533) there are applications waiting to process data coming via the respective port.
- In such cases, we say that the node/process is “listening” to the port, and the port is a “listening port”.
- If the port is not listening, then it is a “closed port”.
- Data sent to a closed port will be dropped.



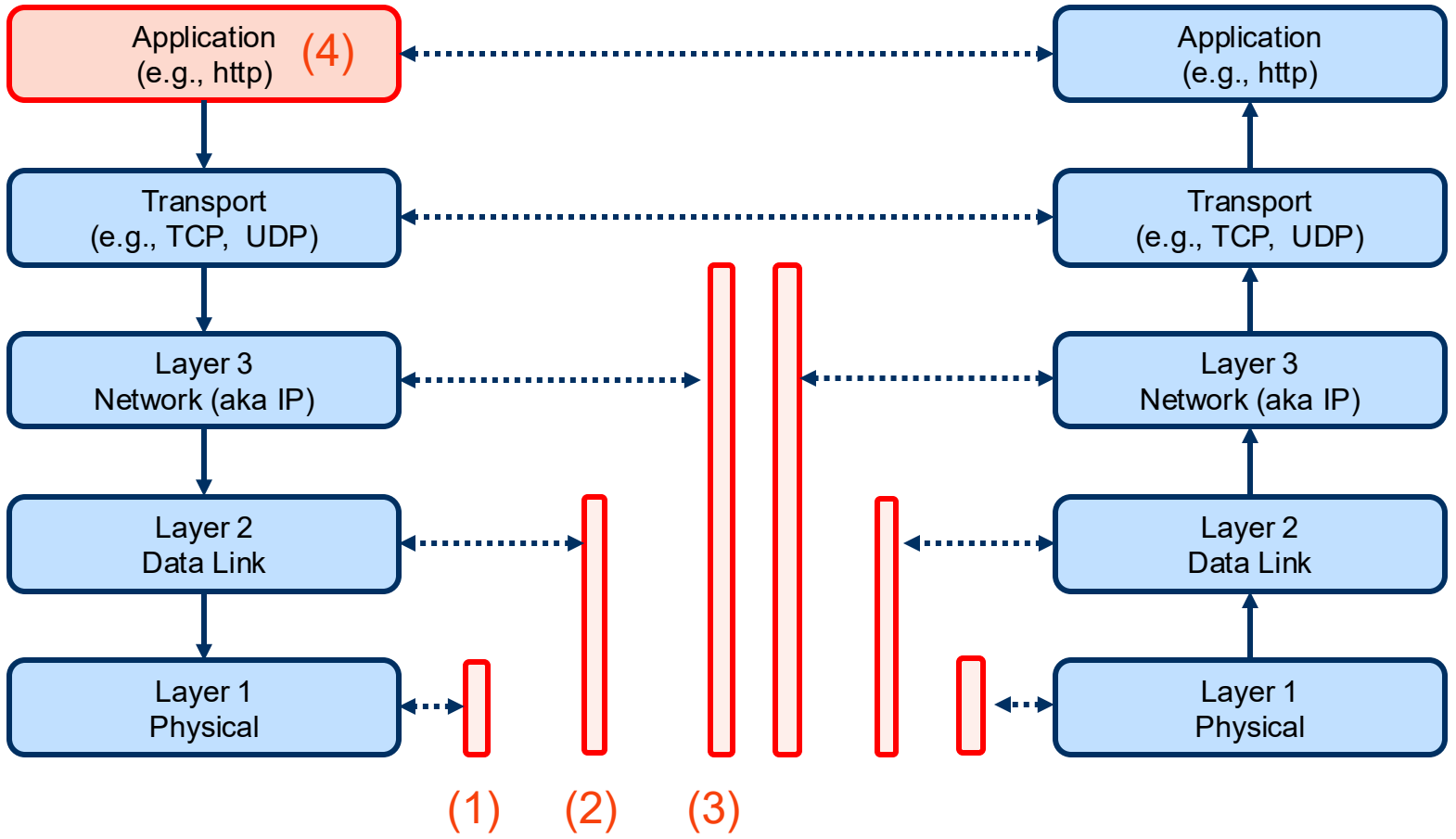
3 Network Attacker

Who is the Man-In-The-Middle?

- TCP/IP is reliable but not “secure”.
  - Malicious intermediate nodes along the communication route can modify data in the header and payload – can be MITM
  - E.g., spoof an IP packet to inform one node to close the connection; reorder the packets;
- Unless otherwise stated, the MITM can sniff, spoof, modify, drop the header and payload.
- The ***MITM in `layer x`*** means MITM along the `layer x` virtual connection.
  - The MITM can see and ***modify data unit of that layer***.

Where is the Man-In-The-Middle?

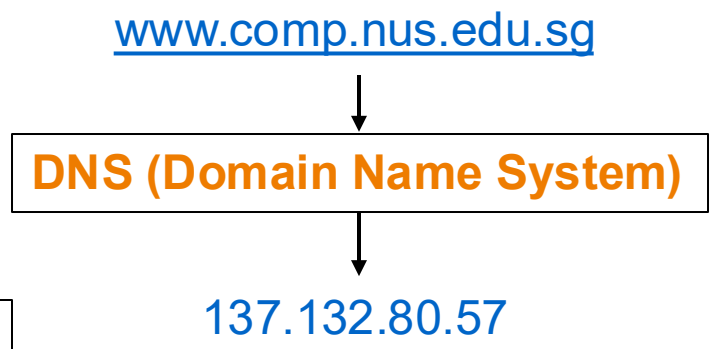
1) MITM in the physical layer.
   - E.g. Tap into the Internet cable, sniff the wireless communication in café
2) MITM in the link layer.
   - E.g. Malicious café owner who offer the Wi-Fi.
3) MITM in the IP/Transport layer.
   - E.g. ISP (such as Singtel)
4) MITM in the application.
   - E.g. a malware in the browser.



4 DNS Attacks

Domain Name System (DNS)

- Given a domain name (e.g., www.comp.nus.edu.sg), its ip address can be found by querying a remote DNS server.
- The client who initiates the query is called the *resolver*.
  - If the address is found, we say that the domain name is *resolved.*




```
$ nslookup www.comp.nus.edu.sg
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
www.comp.nus.edu.sg canonical name = www0.comp.nus.edu.sg.
Name:   www.comp.nus.edu.sg
Address: 137.132.80.57
```

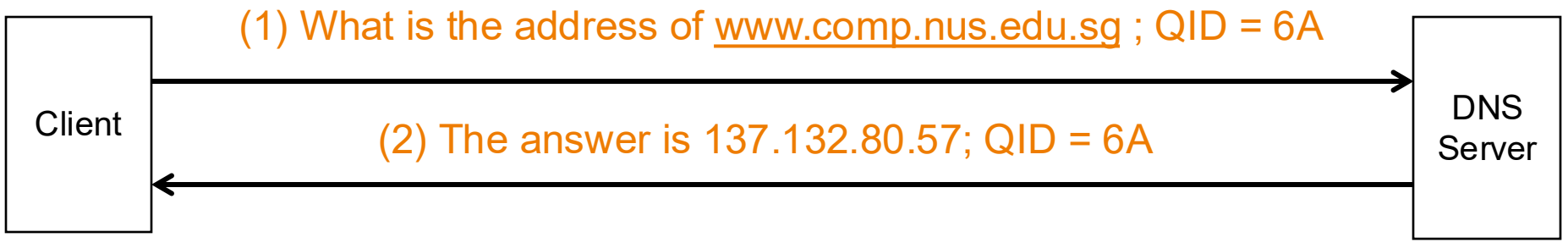
← The domain name to lookup
← Address of the DNS server
← result of the query

Security problems?

- Confidentiality?
- Integrity?
- Availability?

DNS Query and Answer

- Client sends a query to DNS Server (using UDP protocol)
- DNS sends the answer back (using UDP protocol)



- The query contains a ***16-bit number***, known as QID (query ID).
  - The response from the 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.

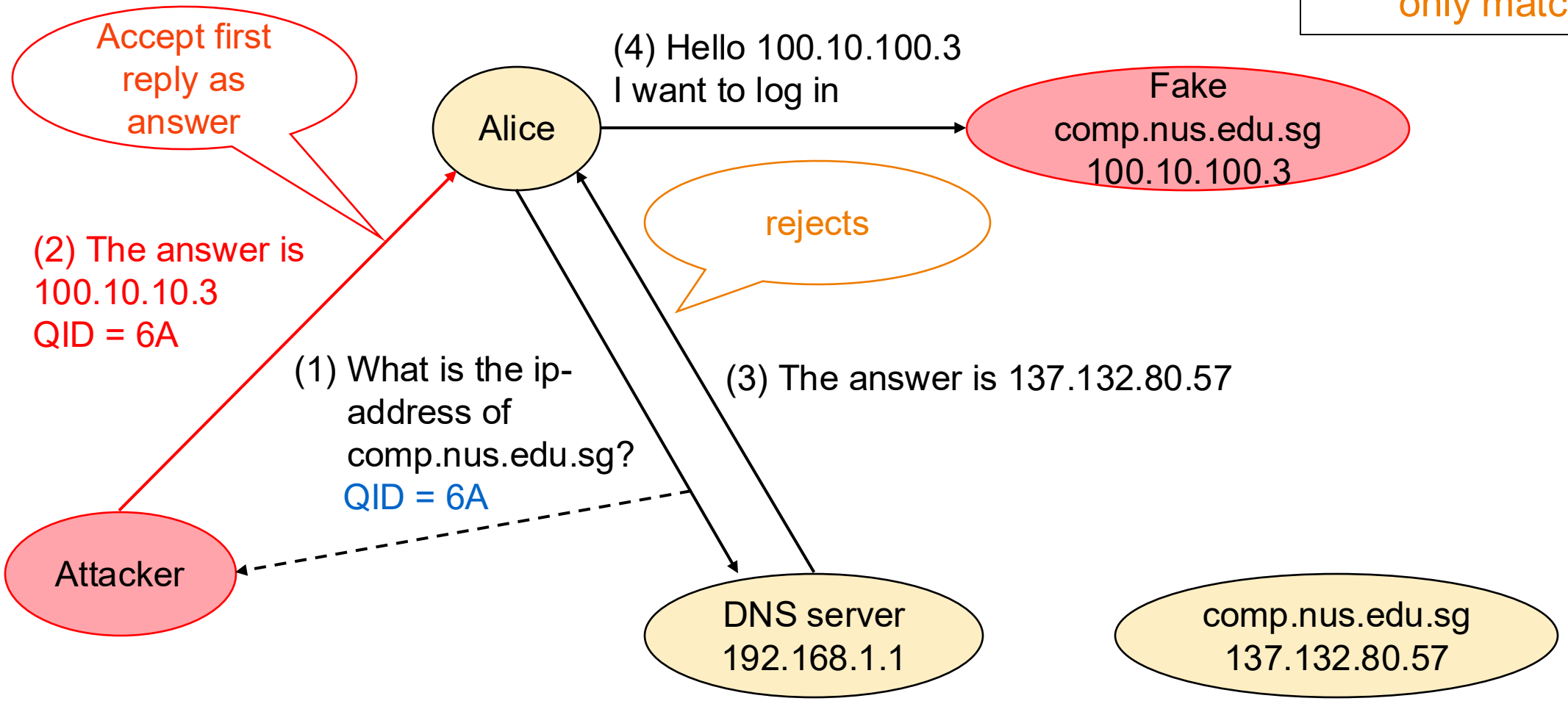
Vulnerable?

DNS Spoofing: Attack Scenario

- Alice is using a café ***free Wi-Fi*** to surf the web.
- Alice wants to visit and login to comp.nus.edu.sg
- We consider an attacker who is also in the café. Since the Wi-Fi is not protected, the attacker can
  - Sniff data from the communication channel.
  - Inject spoofed data into the communication channel.
- However, the attacker can't remove/modify data sent by Alice.
- Attacker owns a webserver at 100.10.10.3 which is a spoofed NUS website.

DNS Spoofing

Cannot verify if response is coming from correct source or has not been modified – only matches QID



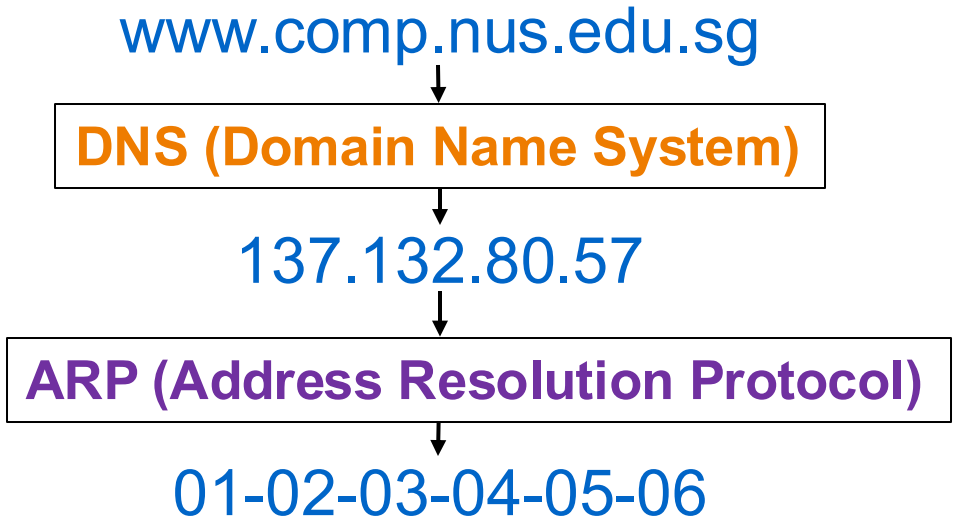
Denial of Service on DNS

- DNS is an important component as it resolves the domain name.
- Hence, an DNS server can be a "***single-point-of-failure***" of the network.
- A denial of service (DoS) attacks can be launched on a web service (i.e., DNS server), instead of directly attack the web server.
- When DNS server is down, the web service is not longer reachable.
- E.g., see attack on wikiLeak
- **Countermeasure**: redundant servers, rate limiting, etc.

5 Poisoning ARP Table

Address Resolution Protocol (ARP)

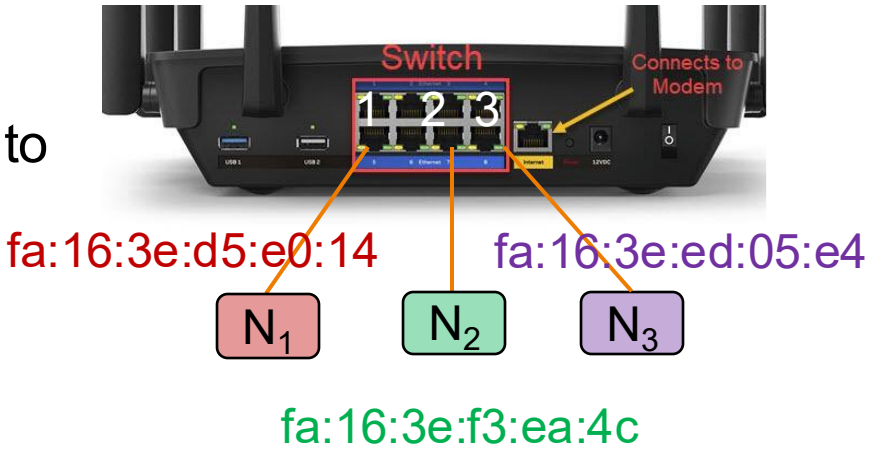
- Resolution of ***IP address to mac address***
- Data Link Layer
- When a device knows the the IP address of the ***next hop router on the same network*** but needs the corresponding MAC address
- ARP resolves the router's IP address to its MAC address to allow packet ***forwarding at the Data Link Layer***



Switch

- Direct data packets between devices or ***nodes on the same local network*** using MAC addresses
- The switch ***keeps a table*** that associates the port to the mac-addresses.
  - N1 directs a data frame with destination MAC address of N3 to switch
  - Switch does a lookup to identify the port (i.e., port 3), then forward it to that port.
- Switch doesn't understand IP-addresses and doesn't store IP-addresses.

Switch's port	Mac-address
1	fa:16:3e:d5:e0:14
2	fa:16:3e:f3:ea:4c
3	fa:16:3e:ed:05:e4



Address Resolution Protocol (ARP) Table

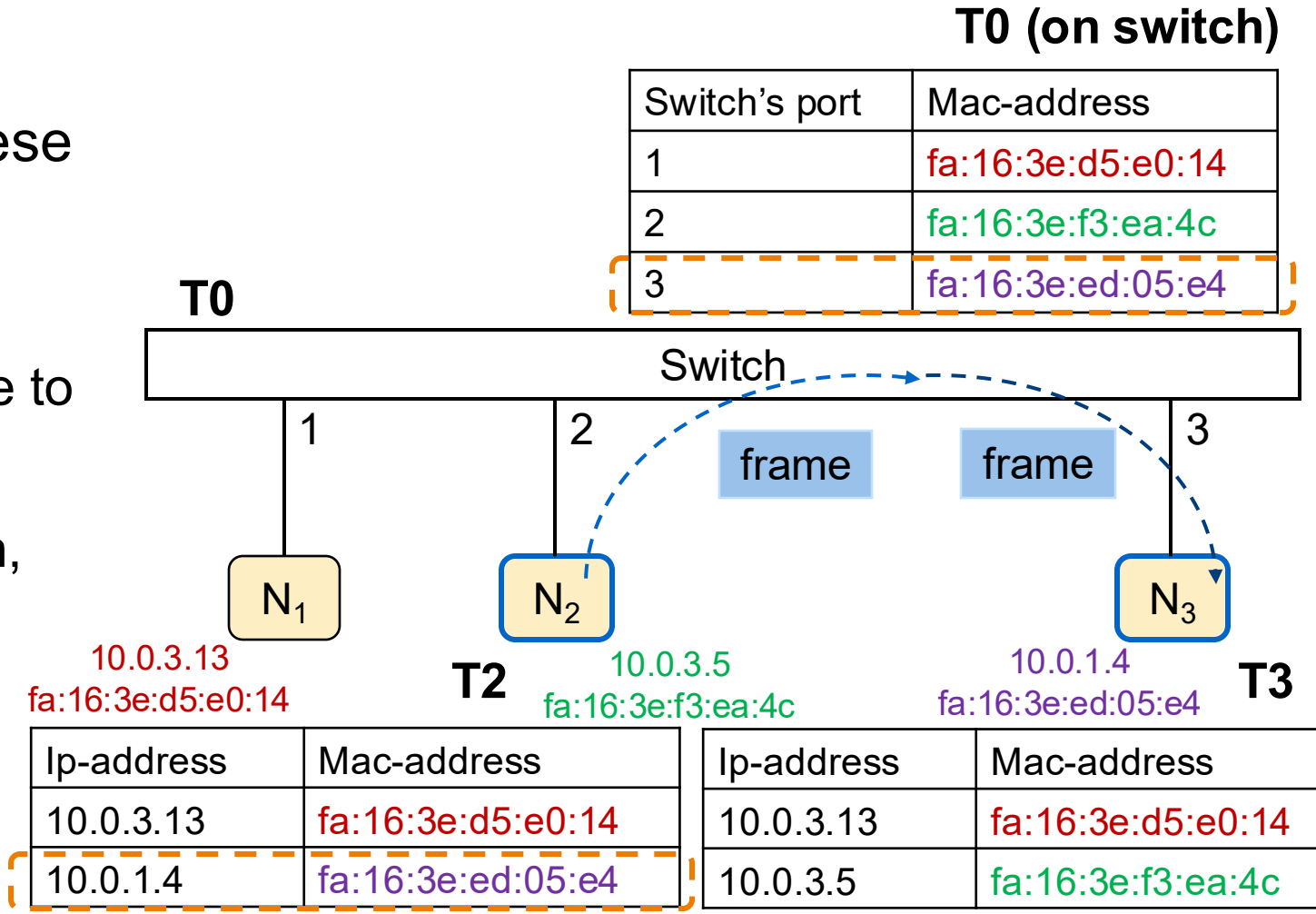
- An ARP table is a database maintained by each device or nodes on a subnet,
  - It stores mappings between ***IP addresses and MAC addresses***
- Resolution of ip-address to mac-address is done by the nodes
- The nodes update each others table using ARP.
- ARP poisoning is an attack that modifies (aka “poisons”) the tables so as to gain Man-In-The-Middle access.

N_1

ip-address	Mac-address
10.0.3.13	fa:16:3e:d5:e0:14
10.0.1.4	fa:16:3e:ed:05:e4

When N_2 want to send to IP address 10.0.1.4

- Under normal circumstances, these are carried out when N2 sends a packet to 10.0.1.4
  - N2 looks up the table T2. Resolve to fa:16:3e:ed:05:e4
  - N2 sends the frame to the switch, specifying destination fa:16:3e:ed:05:e4
  - Switch looks up the table T0, redirect the frame to port 3.



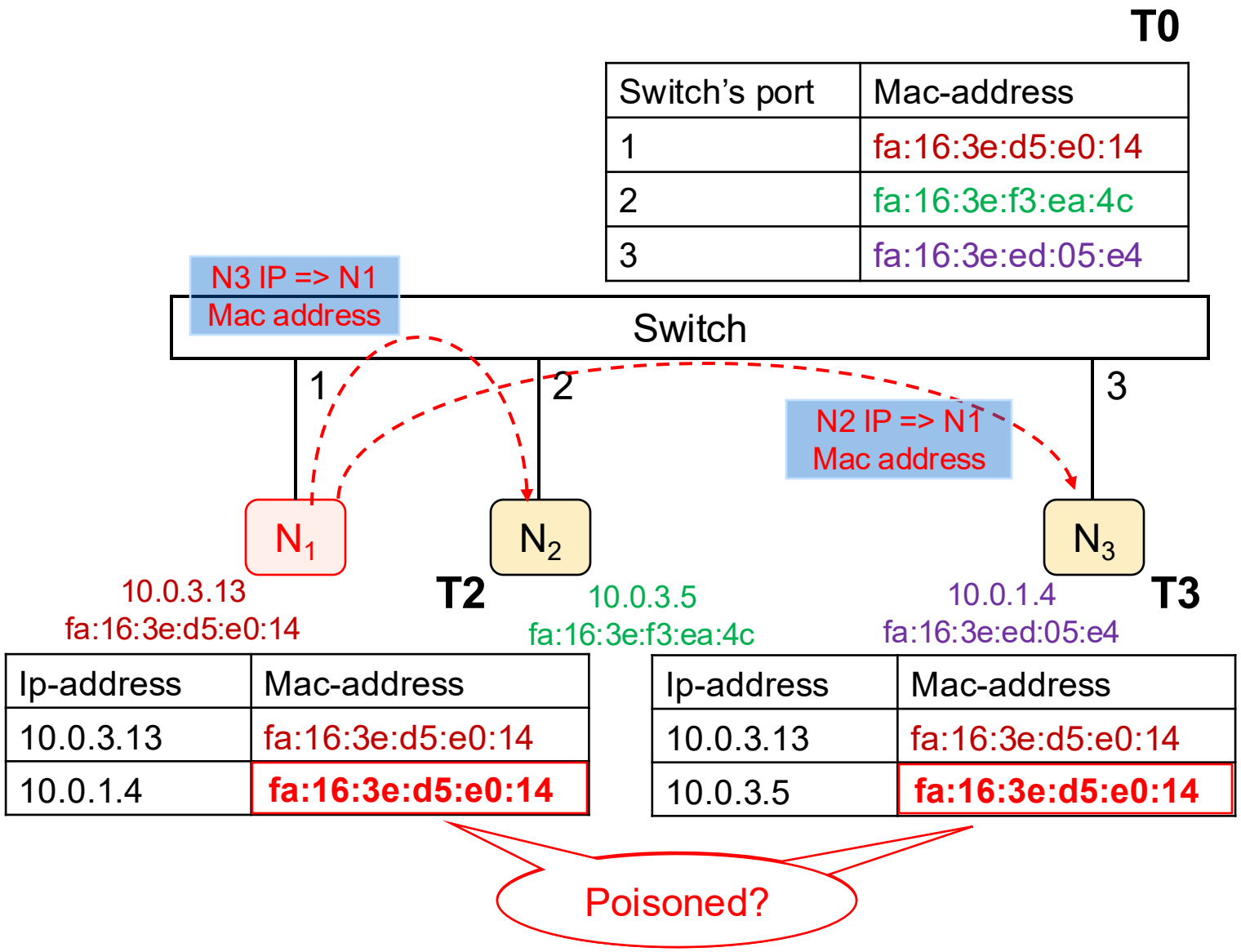
ARP Table Update

- If T2 ***does not have info*** of a particular ip-address
  - N2 broadcasts an ARP ***request*** packet to all devices on the local network.
  - The request essentially asks, "***Who has IP address X.X.X.X? Tell me your MAC address***."
- The device with the requested IP address receives the ARP request and ***reply*** with an ARP reply packet.
  - This reply includes the sender's IP address and corresponding MAC address.
  - Upon receiving the ARP reply, the requesting node updates its ARP table with the new IP-to-MAC address mapping.
- Any node, say N1, can also broadcast the info or to a specific node, ***even if there isn't such a request*** (reply with out a request).

Vulnerable?

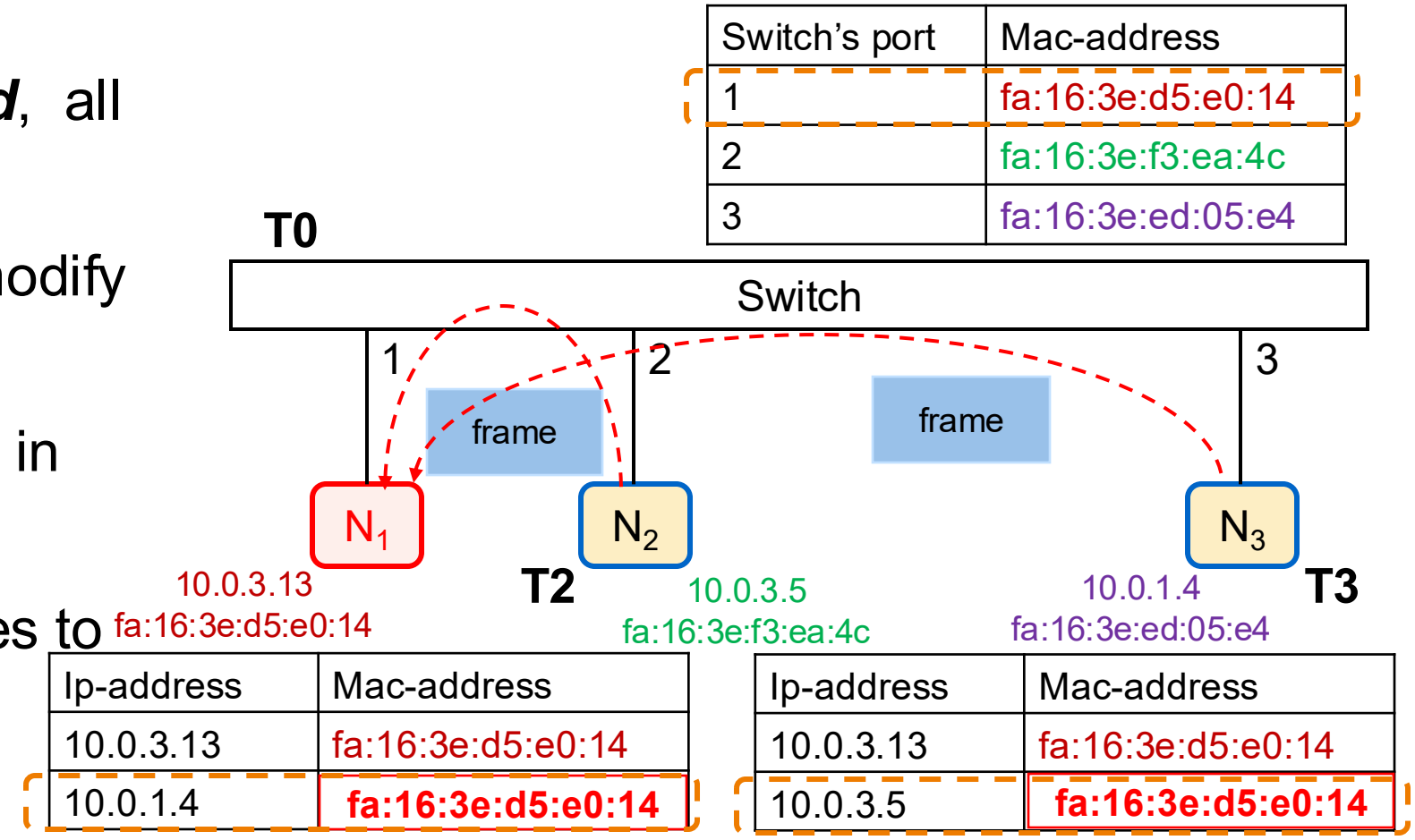
Attack: N_1 wants to be MITM between 10.0.3.5 and 10.0.1.4

- N_1 informs N_2 that mac address of 10.0.1.4 (i.e., N3's) is fa:16:3e:d5:e0:14 (i.e., N1's)
  - N3's IP => N1 mac
  - 10.0.1.4 => fa:16:3e:d5:e0:14
- N_1 informs N_3 that mac address of 10.0.3.5 (i.e., N2's) is fa:16:3e:d5:e0:14 (i.e., N1's)
  - N2's IP => N1 mac
  - 10.0.3.4 => fa:16:3e:d5:e0:14



When N_2 want to send to IP address 10.0.1.4

- After the tables are ***poisoned***, all frames will be sent to N_1.
- N_1 can relay the frames, or modify the frames before relaying.
- Hence, N_1 become the MITM in layer 2.
- Similar when N3 sends frames to N2



6 Denial of Service Attack

Denial-of-Service (DoS) Attacks

- Attempt to ***disrupt the normal functioning*** of a targeted server, service, or network => effect availability
- Many successful DoS attacks simply flood the victims with overwhelming requests/data.
- For DoS to be effective, large number of attackers are required
  - A single attacker can send requests only at a low rate
- When DoS is carried out by a large number of attackers, this is called **DDoS: Distributed Denial of Service**

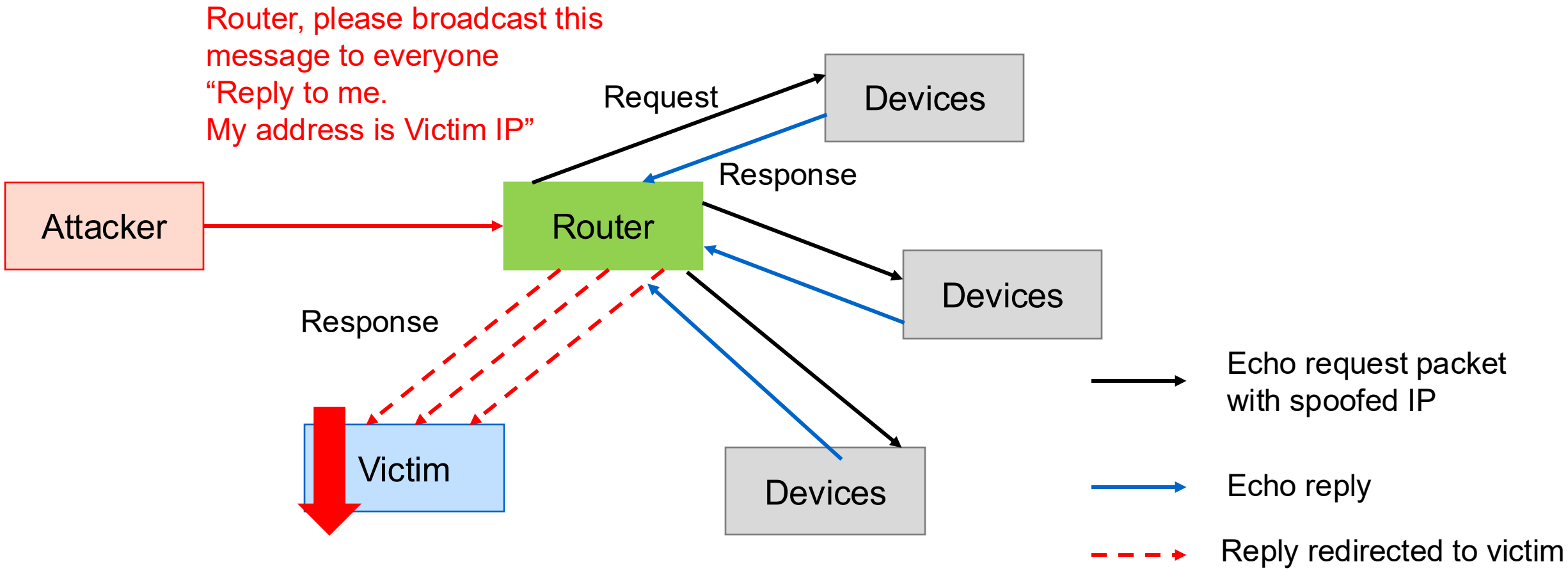
Reflection Attack

- A type of DoS in which the attackers send requests to intermediate nodes, which in turn ***send overwhelming traffic to the victim***.
- Attacker ***spoofs the victim's IP address*** as the source.
- The intermediate nodes then sends its response back to the spoofed IP (the victim)
- Indirect, and thus more difficult to trace.

Reflection Attacks: ICMP/Smurf flood

- An attacker sends the request "***ICMP PING***" to a router, instructing the router to broadcast this request.
  - The ***source ip-address of this request is spoofed*** with the victim ip address.
- The router broadcasts this request.
- Each entity who has received this request, replies to it by sending an ***"echo reply" to the source***, which is the victim.
- The victim's network is overwhelmed with the "echo reply".

Reflection Attacks: ICMP/Smurf flood



Smurf Flood Preventive Measure

- Most router are now configured not to broadcast by defaults.
  - This attack is no longer effective.

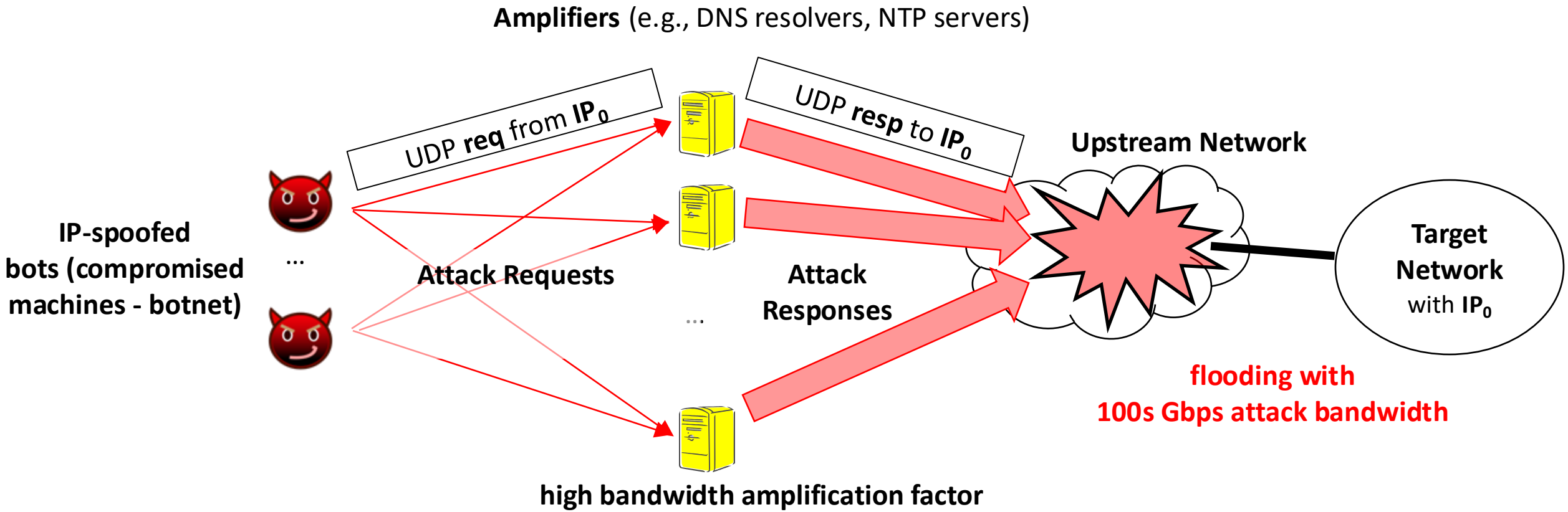
- Configure firewalls to block incoming ICMP Echo Request packets directed at broadcast addresses.

Amplification Attack

- A variation of reflection attacks where the ***intermediate nodes response is significantly larger*** than the attacker's request.
  - its amplification factor, which is the size of traffic the victim received over the size of traffic sent by the attacker.
- A single request could trigger multiple responses from the intermediate nodes.
- This difference in request-versus-response size "amplifies" the traffic directed at the victim.

DDoS Amplification Attack

- Exploit UDP services (e.g., DNS, NTP) to amplify a relatively small volume of traffic into a larger volume



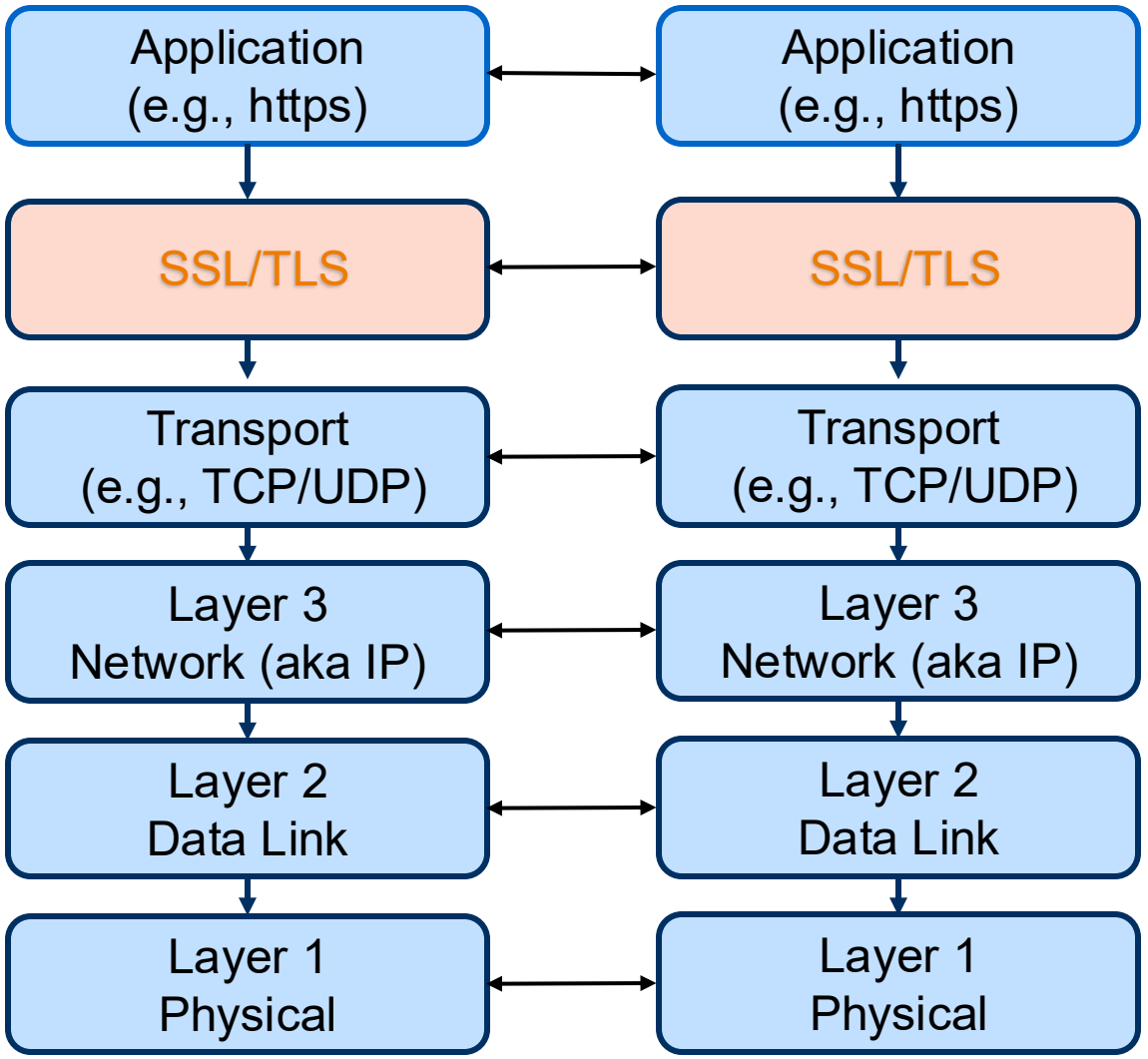
7 Securing Different Layers

Securing the Communication Channel

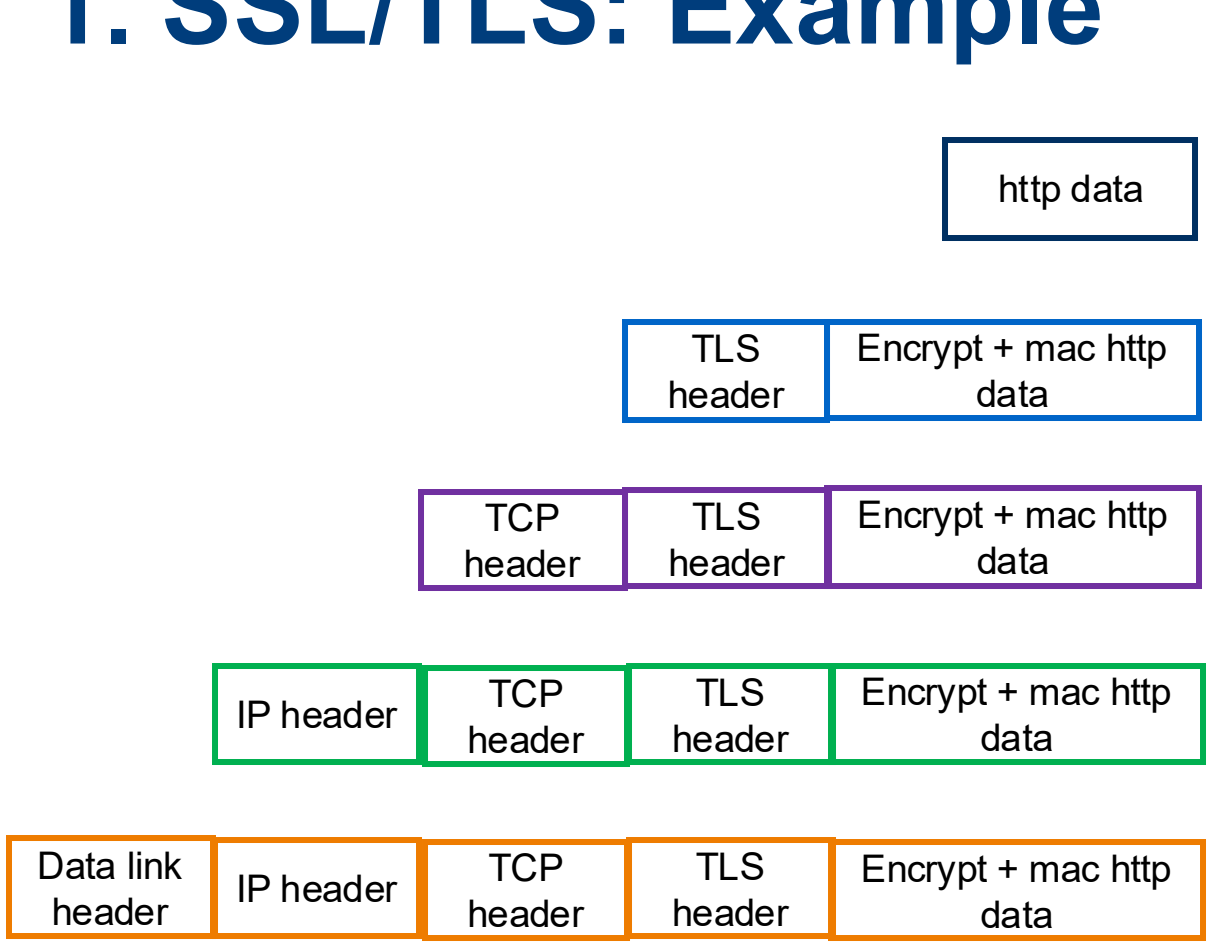
- In the past lectures, we have illustrated that cryptography techniques can be deployed to achieve confidentiality and authenticity over a public communication channel, even if the adversary can sniff & spoof data.
- There are many security protocols achieving that but operates at different “layers”.
- The well-known TLS/SSL, WPA, IPSEC protect different layers.

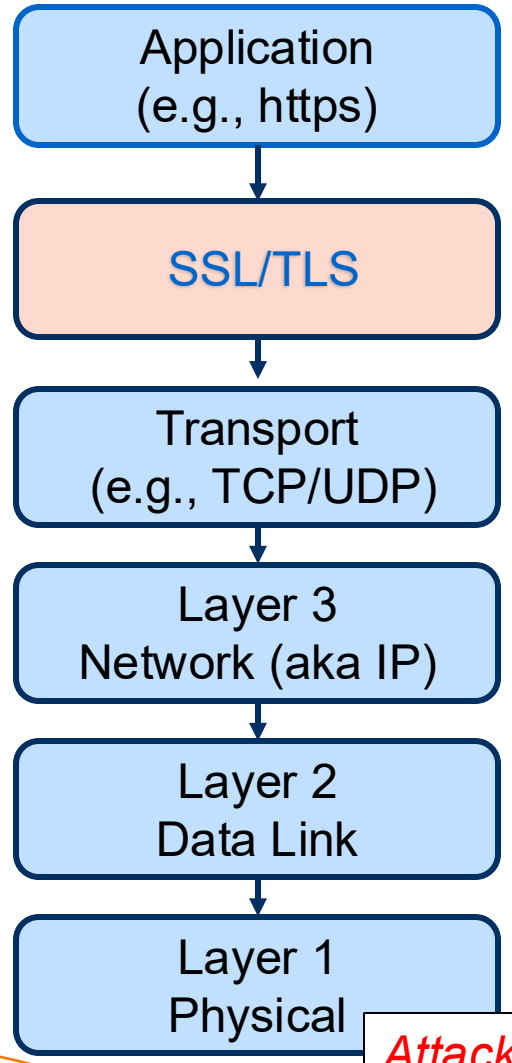
1. SSL/TLS

- The SSL/TLS sit on top of Transport layer.
- When an application (say browser) wants to send data to the other end point, it first passes the data and the address (ip address, port) to SSL/TLS.
- Next SSL/TLS first “protects” the data using encryption and mac.



1. SSL/TLS: Example





Alice uploads a report to Canvas

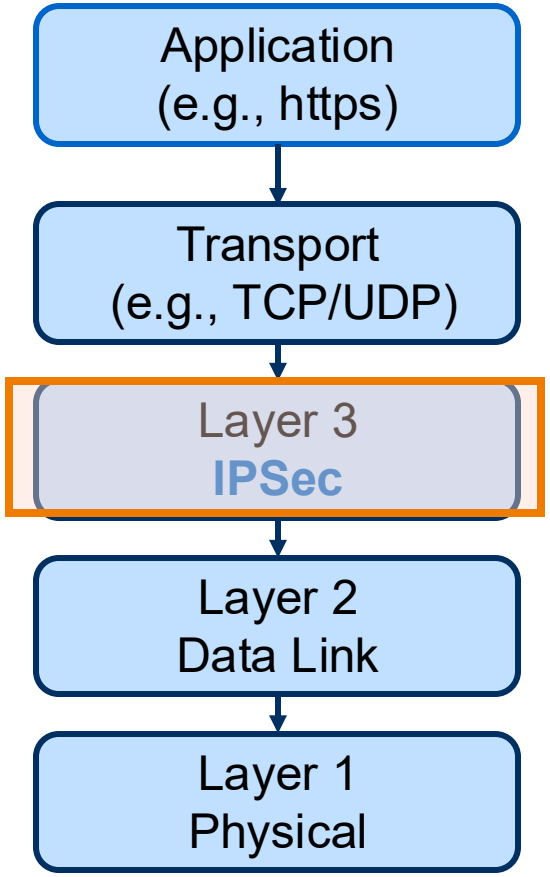
The receiver end-point decrypts the received data at the corresponding layer.

Can attacker know Alice is visiting Canvas website ?

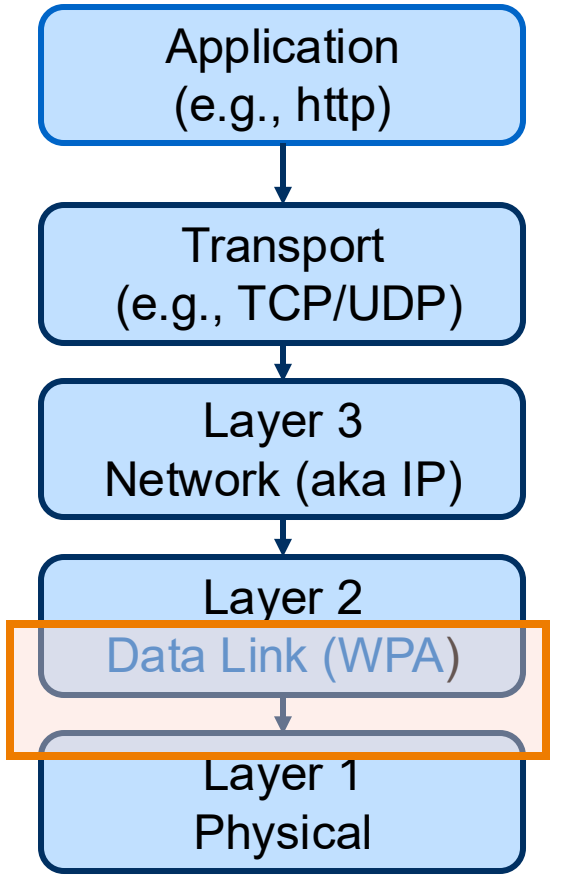
Attacker at physical layer. (able to sniff and spoof) => MITM

2. Internet Protocol Security (IPSec)

- IPSec is a mechanism whose goal is to protect the IP layer - secure all IP traffic between endpoints
- Securing network connections between host-to-host, network-to-network (gateways) or network-to-host (gateway and host)



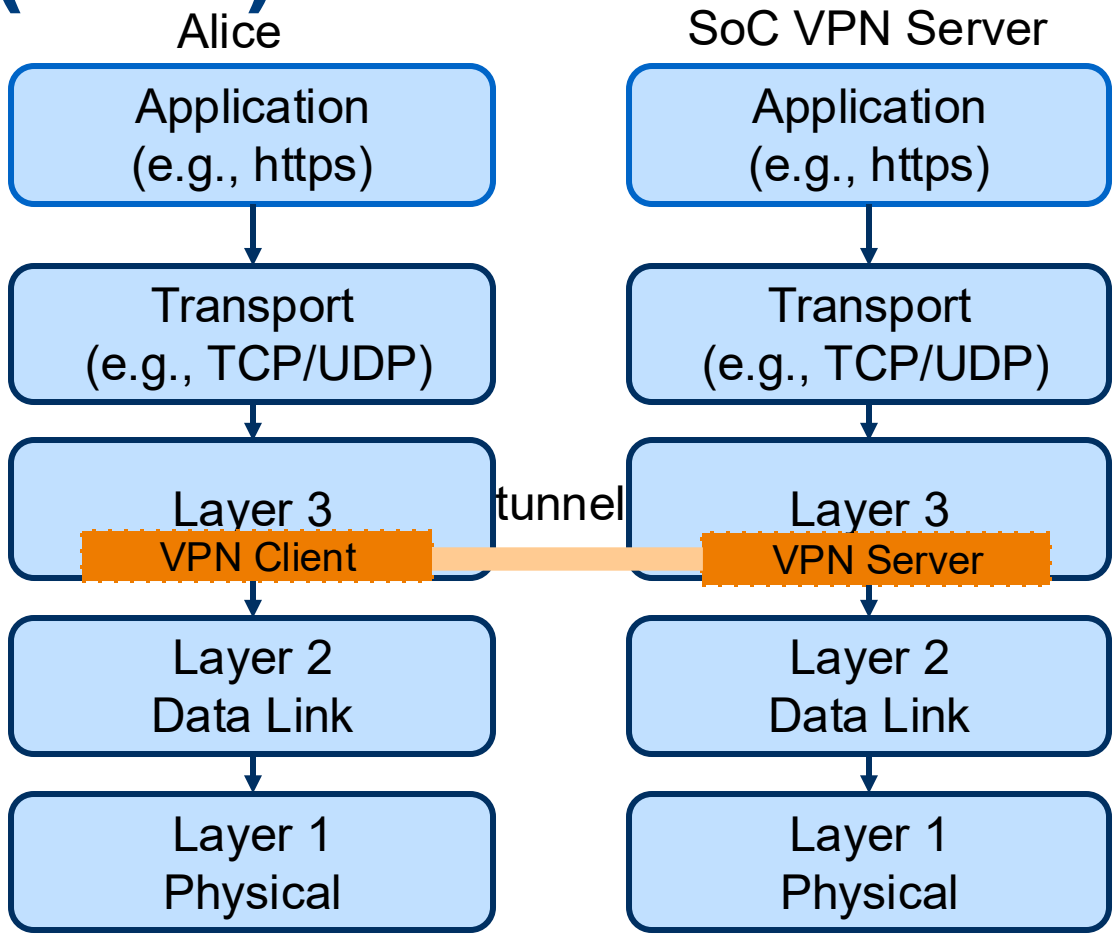
3. Wi-Fi Protected Access II (WPA2)

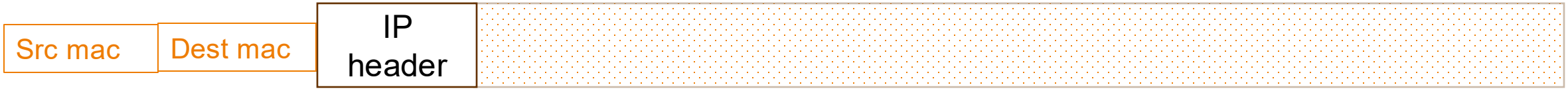


- A popular protocol to protect data transmitted over Wi-Fi networks
- WPA2 provides protection in layer 2 (Link) and layer 1 (Physical).
  - data traveling between a ***wireless device and the access point*** is confidential and protected from eavesdropping.
- Not all information in layer 2 are protected.

4. Virtual Private Network (VPN)

- Enable remote user to securely connect to private network
- First, VPN client and the VPN server establish a connection, which is called a “tunnel” => authentication and verification done, establish session keys.
- While Alice communicating with another node, say Bob, the VPN client encrypt the entire payload and add a new IP header
- From Bob’s point of view, Alice is communicating with the virtual node with ip-address in NUS. (Thus, not knowing Alice actual IP address).





4. VPN

- The “tunnel” can be established in many ways.
  - **Use TLS/SSL or use IPSec**

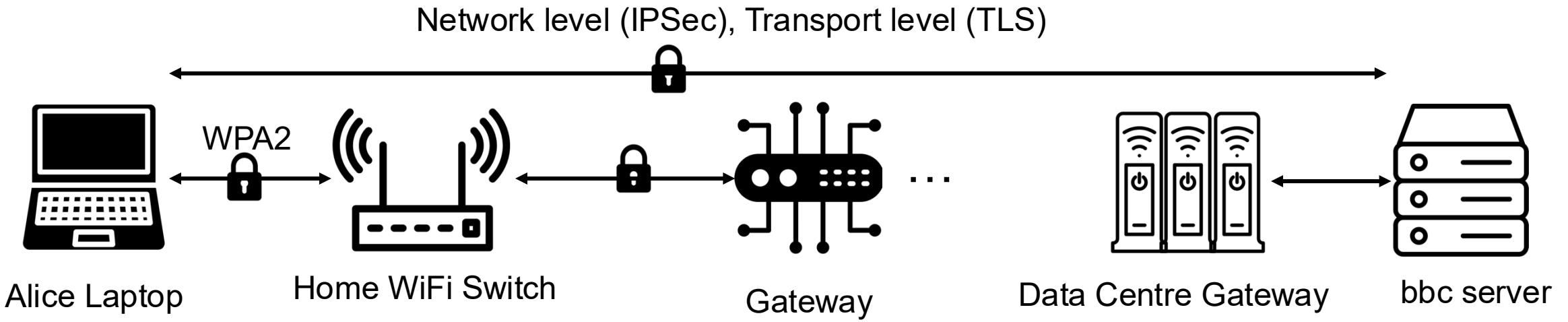
- Since layer 3 & 4 are often treated as one, if the client is in layer 4, many documents still say that it is in layer 3, and call it Layer 3 VPN.

- There are many different configurations and variants.
  - E.g., for efficiency, some VPN clients might choose not to tunnel DNS queries.

Which Layer Should We Protect?

- From the previous, it seems that by protecting the lowest layer, we can protect information in all layer – not feasible
- Why?
  - Intermediate node need to access some higher layer info, e.g., ip-address, and sit in higher layer.
  - Hence, a malicious intermediate node could be a MITM in higher layer.
- Typical implementation use TLS/SSL + WPA2
- IPSec turns out to be expensive to implement and difficult to deploy.

Which Layer Should We Protect?



Simplified

8 Firewalls and IDS

Motivation

- Consider the computer network in an organization.

- Some nodes ***contain more sensitive information*** than other.
  - Example, Student examination record database server, vs public workstations in lecture hall.

- Some nodes are ***more "secure"***.
  - E.g., When a patch is available, it might take some time (e.g., days, weeks, months) to patch all the systems and need to be prioritized.
  - E.g., Certain nodes operate in a more hostile environment, lecture theatre's workstation.

Motivation

- We need to divide the computer network into segments and deny unnecessary access.
  - **Principle of least privilege:** control access to the network.
  - **Compartmentalization**: Keep things separated to limit the impact of any single failure or attack.

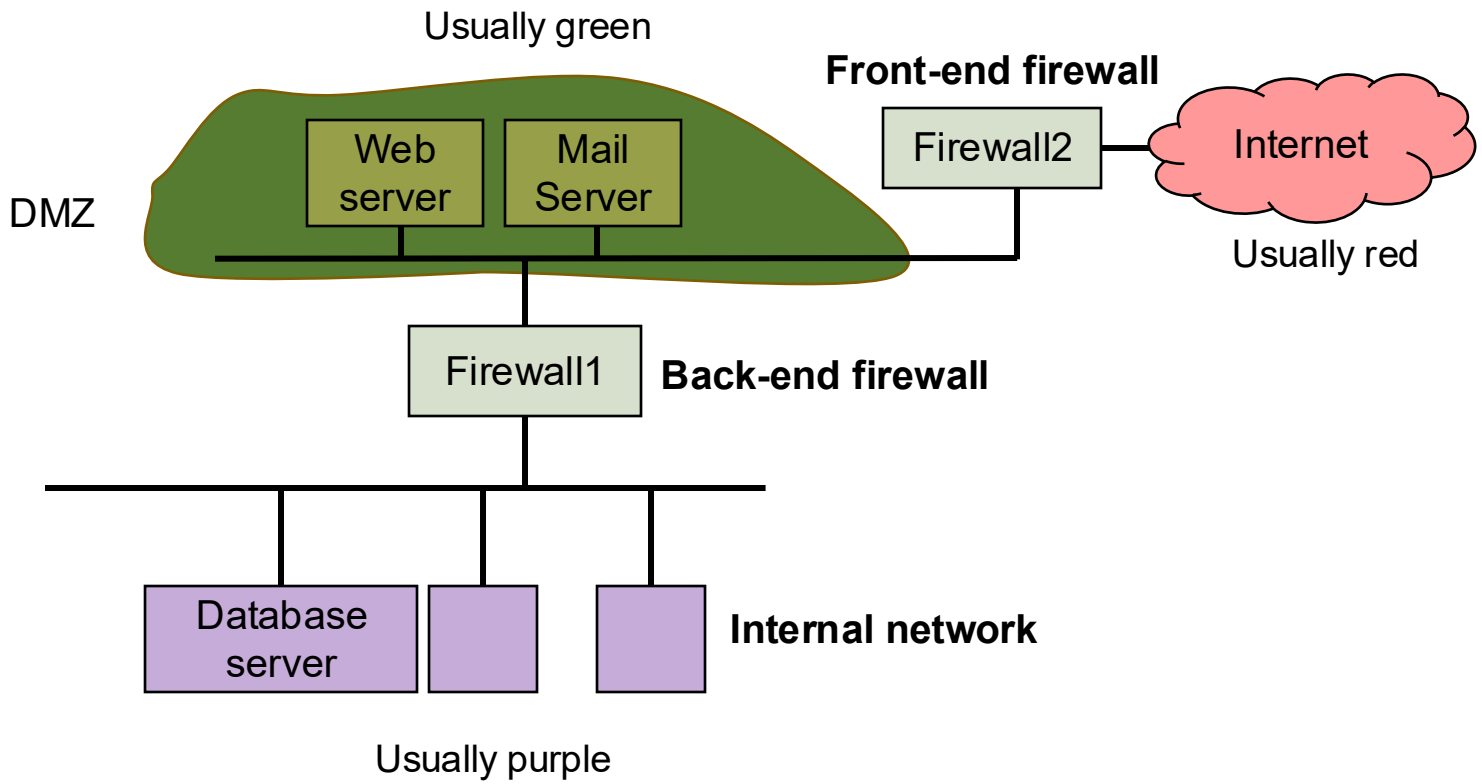
- Tools to control access to the network.
  - **Firewall**: is a gatekeeper, prevents unauthorized access
  - **Intrusion detection system (IDS):** is a watcher that raises alerts by monitoring and analysing

Firewall and DMZ

- A Firewall controls what traffic is allowed to enter the network (***ingress*** filtering) or leave the network (***egress*** filtering).
  - Firewall are devices or programs that control the flow of network traffic between networks or hosts that employ differing security postures. (from Guidelines)
- **DMZ: Demilitarized zone**
  - A sub-network that exposes the organization's external service to the (untrusted) Internet.
  - Separates an internal local area network (LAN) from untrusted external networks, usually the Internet.
  - Its purpose is to add an extra layer of security to an organization's internal network.
- DMZs are created using firewalls

A typical 2-firewall setting

- **Internet**: untrusted
- **Front-end-firewall**: control incoming/outgoing traffic to DMZ
- **DMZ**: public facing servers accessible from the Internet
- **Back-end firewall**: Only specific traffic is allowed from the DMZ to the internal zone
- **Internal network**: Highly trusted area where critical server are present.



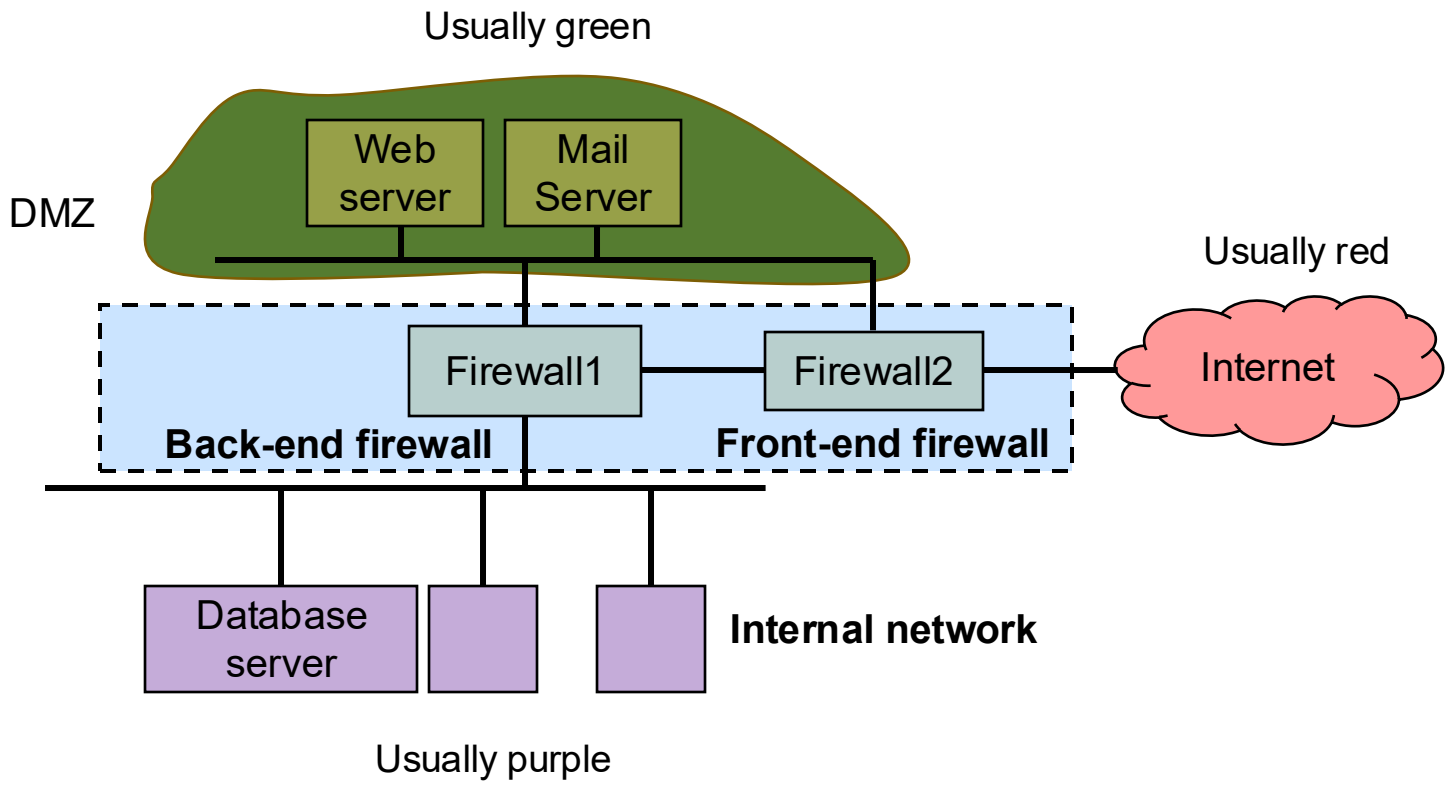
A typical 2-firewall setting

- If a web/mail server in the DMZ is compromised, the attacker still has to bypass the **back-end firewall** to reach sensitive data.

- The **two-firewall model** ensures **defense in depth**:
  - One firewall (front-end) restricts external access.
  - Another firewall (back-end) restricts access to the internal network.

A typical 2-firewall setting.

Very often, the two firewalls residue in a single hardware/router



Packet filtering / Screening (link)

- Firewall's controls are achieved by ***"packets filtering"*** (aka screening).
  - Filtering may occur in router, gateway/bridge, host, etc.
- Packet filtering inspects every packet, typically only on the TCP/IP packet's header information (network & transport layer).
- If the ***payload is inspected***, we call it ***deep packet inspection (DPI)***.
- Action taken after inspection could be
  - Allow the packet to pass
  - Just drop the packet
  - Reject the packet (i.e., drop and inform the sender)
  - Log info
  - Notify system admin
  - Modify the packet (for more advanced device).

Example of firewall rules

- Drop packets with “source ip-address” not within the organization’s network. This *can stop attacks originated within the network*
- **Whitelist**
  - Drop all packets except those specified in the white-list. (e.g., drop all except http, email protocol, and DNS)
- **Blacklist**
  - Accept all packets except those specified in the black-list. (e.g., allow https except ip-address in the blacklist).

Example Rule

Rule	Type	Direction	Source Address	Destination Address	Designation Port	Action
1	TCP	in	*	192.168.1.*	25	Permit
2	TCP	in	*	192.168.1.*	69	Permit
3	TCP	out	192.168.1.*	*	80	Permit
4	TCP	in	*	192.168.1.18	80	Permit
5	TCP	in	*	192.168.1.*	*	Deny
6	UDP	in	*	192.168.1.*	*	Deny

Matching condition ***action***

- The rules are processed sequentially starting from rule 1, 2, …. The first matching rule determines the action.
- The symbol “*” matches any value. This is a symbol in “regular expression”.

Types of firewall

- NIST's document (NIST 800-41) groups the firewalls into 3 types:

1. **Packet filters.**
   - Inspect only the header. (mainly IP packet's header).
   - **Use**: Blocking traffic from certain IP or port
2. **Stateful Inspection.**
   - Keep a state on previously received packets.
   - E.g., counting number of connection a particular IP address has made in the past one hour.
   - **Use**: Blocking abnormal connection pattern or unauthorized session attempts.
3. **Proxy.**
   - Act as intermediaries that fully receive, inspect, and forward (possibly modify) packets between client and server.
   - **Use**: block certain URLs or scan for malware in HTTP traffic

Intrusion Detection System (IDS)

- An IDS is a security tool or software that ***monitors computer systems*** and networks for signs of
  - malicious activity, policy violations, or security breaches.

- An IDS system consists of a ***set of "sensors"*** that gather data such as logs, network packets, etc.
  - Sensors can be deployed on hosts, or network router.

- The ***data are analyzed*** for intrusion either in real-time or after collection to detect suspicious patterns, attacks, or abnormal behaviour.

Three types of IDS

- **Attack Signature Detection**
  - Looks for specific, well-defined patterns or “signatures” of known attacks in the data collected by sensors.
  - For e.g., using certain port number, certain source ip address.
- **Anomaly Detection**
  - The IDS attempt to detect abnormal pattern that deviate from the established “normal” behaviour of the network
  - For e.g., a sudden surge of packets with certain port number.
- **Behavior-based IDS**
  - Can be viewed as a type of anomaly detection that focuses on human behavior.
  - For e.g. The system might keep the profile of each user and detect any user who deviates from the profile (e.g., start to download large files).

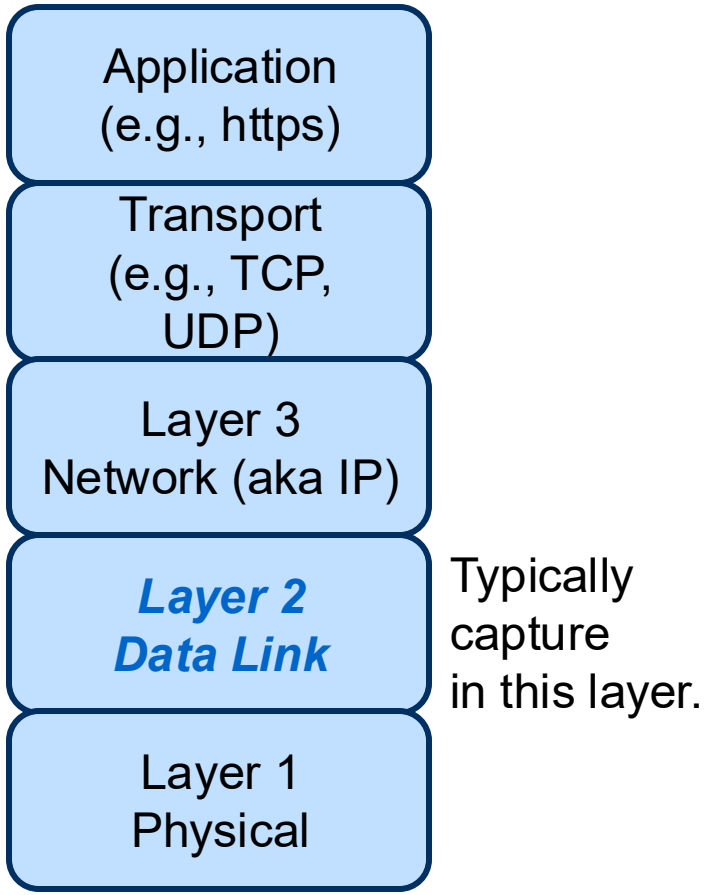
Summary

- Crypto + PKI can provide confidentiality & authenticity even in the presence of MITM.
- **Routing**.
  - **Layering**. Intermediate nodes need to see and modify routing info at different “layers”.
  - Protection at different layers. MITM in different layers.
- Some specific attacks:
  - DNS spoofing, DoS, ARP table poisoning.
- Popular protocols: SSL/TLS (application), IPSEC (network), WPA2 (link)
- Firewalls and IDS
- **Tools**: Wireshark, nmap, nslookup.

Explore On Your OWN: Useful Tools

1. Wireshark (packets analyzer) - Demo

- Wireshark: a free open-sourced packets analyzer.
- Exactly what does wireshark capture?
- Wireshark listens to “interactions” between the OS and the network card driver. (In other words, it is a MITM between OS and network card).
- Hence, header added by the network card, or modification made by the network card, may not be captured by Wireshark. This depends on the OS and the hardware.



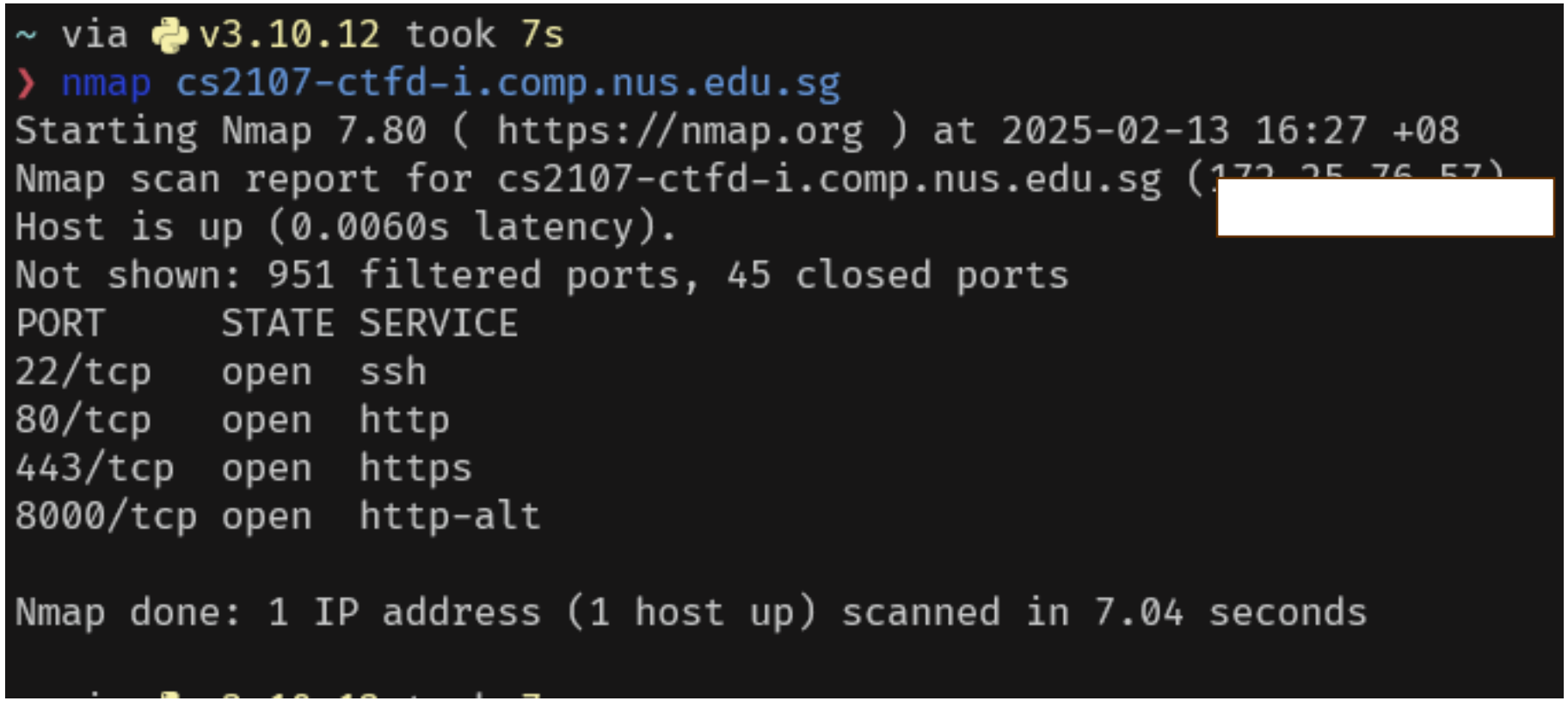
Try Out

- Wireshark – capture, interface, mac address
- Display filter
- Expand each packet – various part
  - TCP
  - TLS handshake
  - DNS
  - ARP
  - HTTPs – application data

2. Nmap (port scanning)

- Port scanning: The process of determining which ports are open in a network.
- Port scanner: A tool for port scanning. E.g., Nmap.
- Port scanner is a useful tool for attacker, and network administrator to scan for vulnerabilities.
- Is port scanning illegal?
- (Read if you want to use it) https://nmap.org/book/legal-issues.html

2. Nmap (port scanning) Example



3. Netwox tool

- Netwox provides a wide range of network utilities and tools for testing, monitoring, and manipulating networks.
- You can run a command like following (the parameters depend on which tool you are using) : `$ sudo netwox number [parameters ... ]`
- For some of the tool, you have to run it with the root privilege.
- If you are not sure how to set the parameters, you can look at the manual by issuing "netwox number --help".