Network Security I - Computer Network Concepts

CSE 565 - Fall 2025
Computer Security

Hongxin Hu (hongxinh@buffalo.edu)

Updates

- **Project 2 SQL Injection Attack**
  - Deadline: Thursday, October 7
- **Assignment 2**
  - Deadline: Tuesday, October 9
- **Midterm Exam**
  - Thursday, October 16

Midterm Exam

- **Question 1: True/False questions. (10 points)**
- **Question 2: Multiple-choice questions. (20 points)**
- **Question 3: Short Answer questions. (50 points)**
- **Question 4: Lab 1 question. (10 points)**
- **Question 5: Lab 2 questions. (10 points)**

Updates

- Survey 1 Large Language Model Security
  - Deadline:
    - Thursday, October 23, 2024

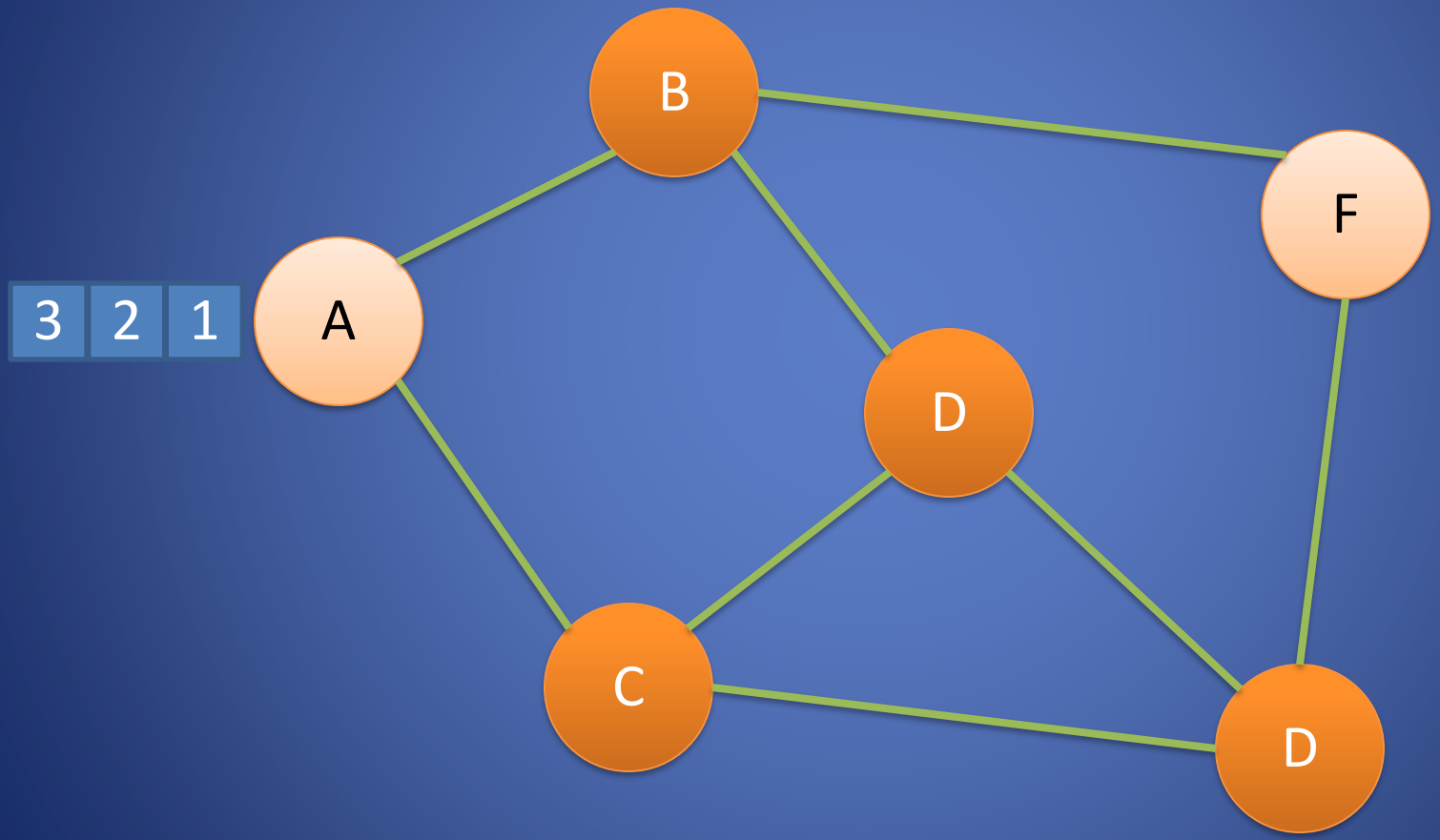
Updates

- Survey Papers
  - Structure
    - Title
    - Abstract
    - Introduction
    - Main Techniques
    - Issues and Problems
    - Future Trends
    - Reference (less than 10)
  - Page limit
    - 3-4 pages excluding references
  - Please use the following IEEE paper template to prepare your survey paper:
    - https://www.ieee.org/conferences/publishing/templates.html

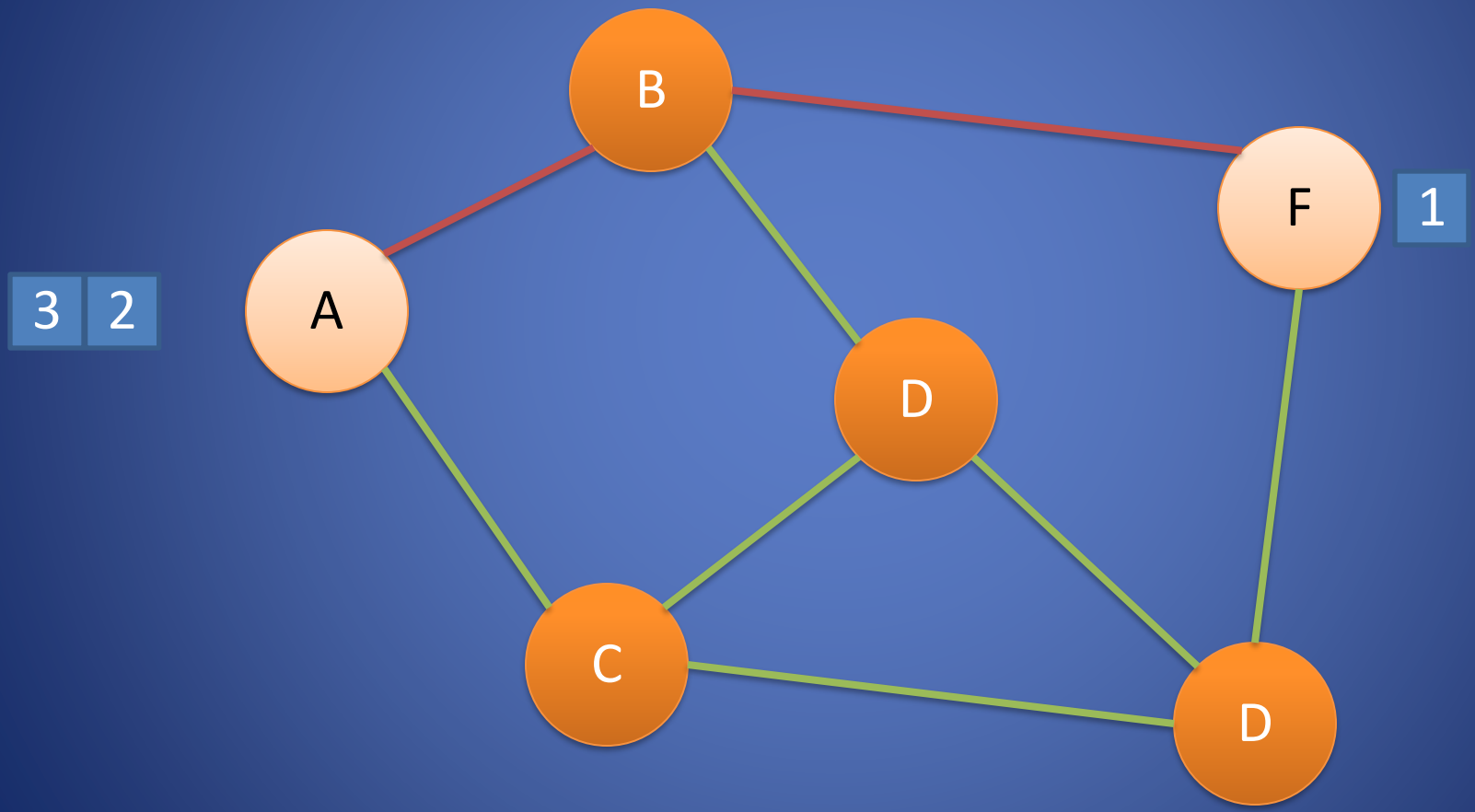
Circuit and Packet Switching

- Circuit switching
  - Legacy phone network
  - Single route through sequence of hardware devices established when two nodes start communication
  - Data sent along route
  - Route maintained until communication ends
- Packet switching
  - Internet
  - Data split into packets
  - Packets transported independently through network
  - Each packet handled on a best efforts basis
  - Packets may follow different routes

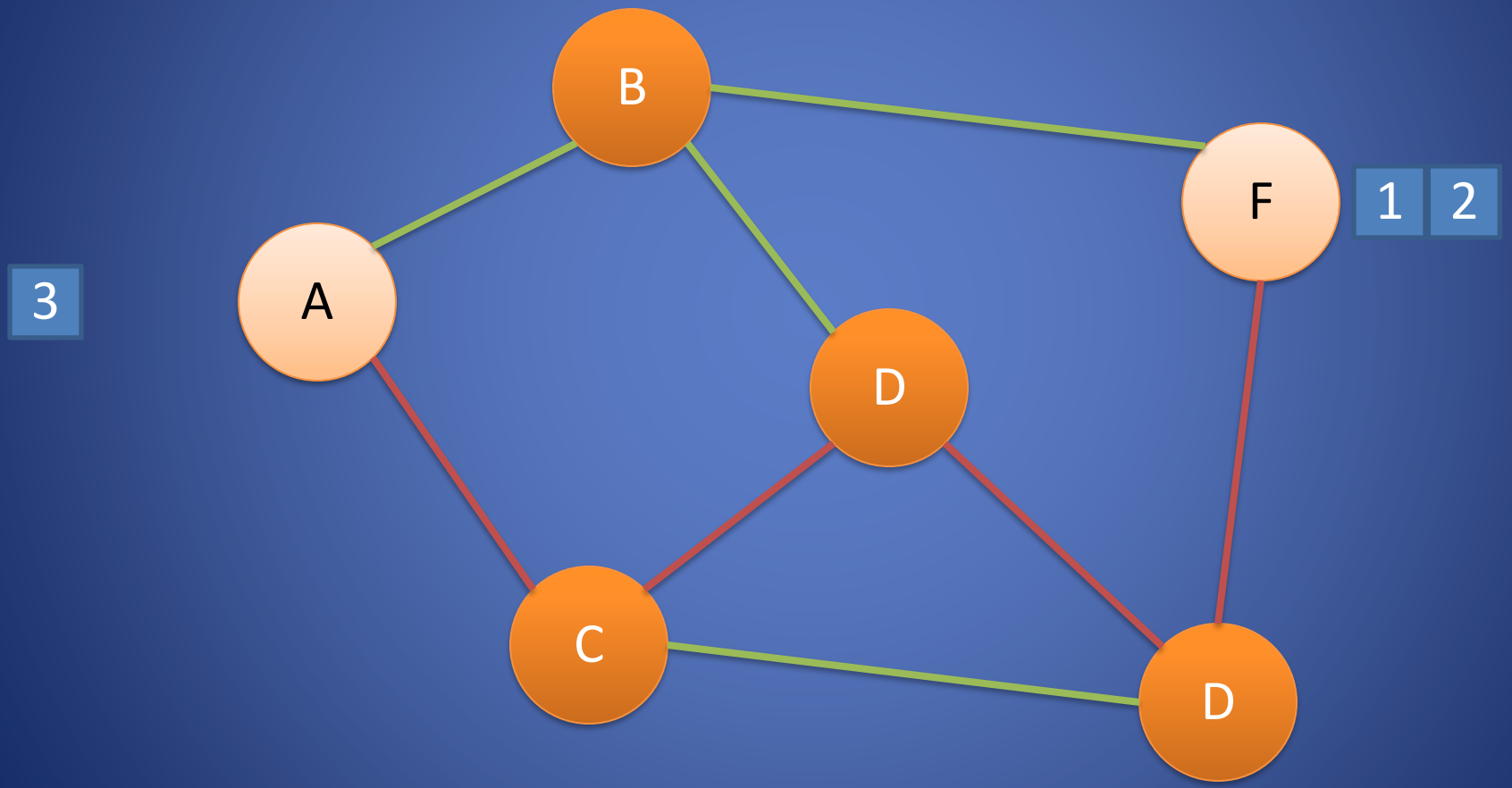
Packet Switching



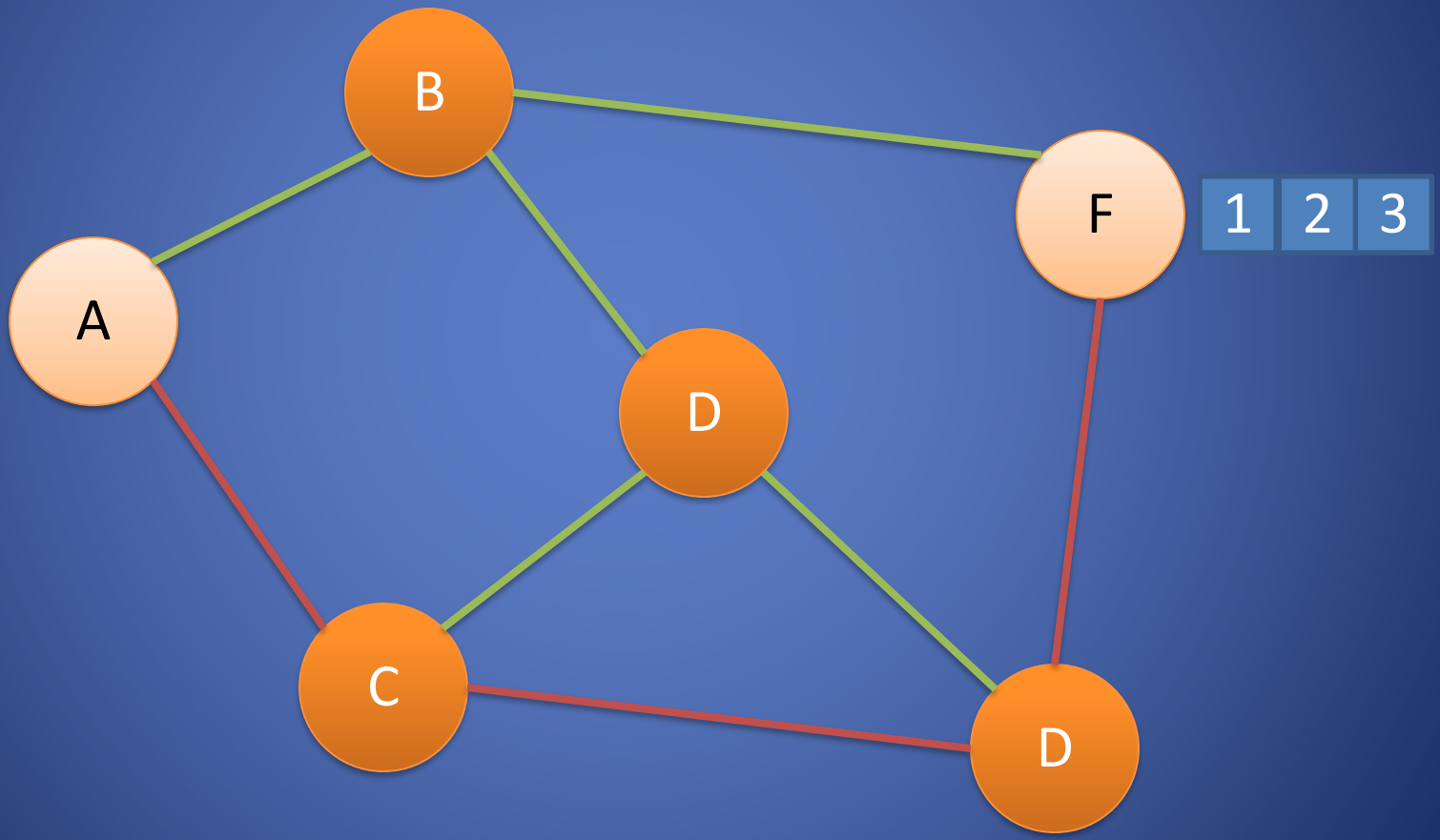
Packet Switching



Packet Switching



Packet Switching



Protocols

- A protocol defines the rules for communication between computers
- Protocols are broadly classified as connectionless and connection oriented
  - Connectionless protocol
    - Sends data out as soon as there is enough data to be transmitted
    - E.g., user datagram protocol (UDP)
  - Connection-oriented protocol
    - Provides a reliable connection stream between two nodes
    - Consists of set up, transmission, and tear down phases
    - Creates virtual circuit-switched network
    - E.g., transmission control protocol (TCP)

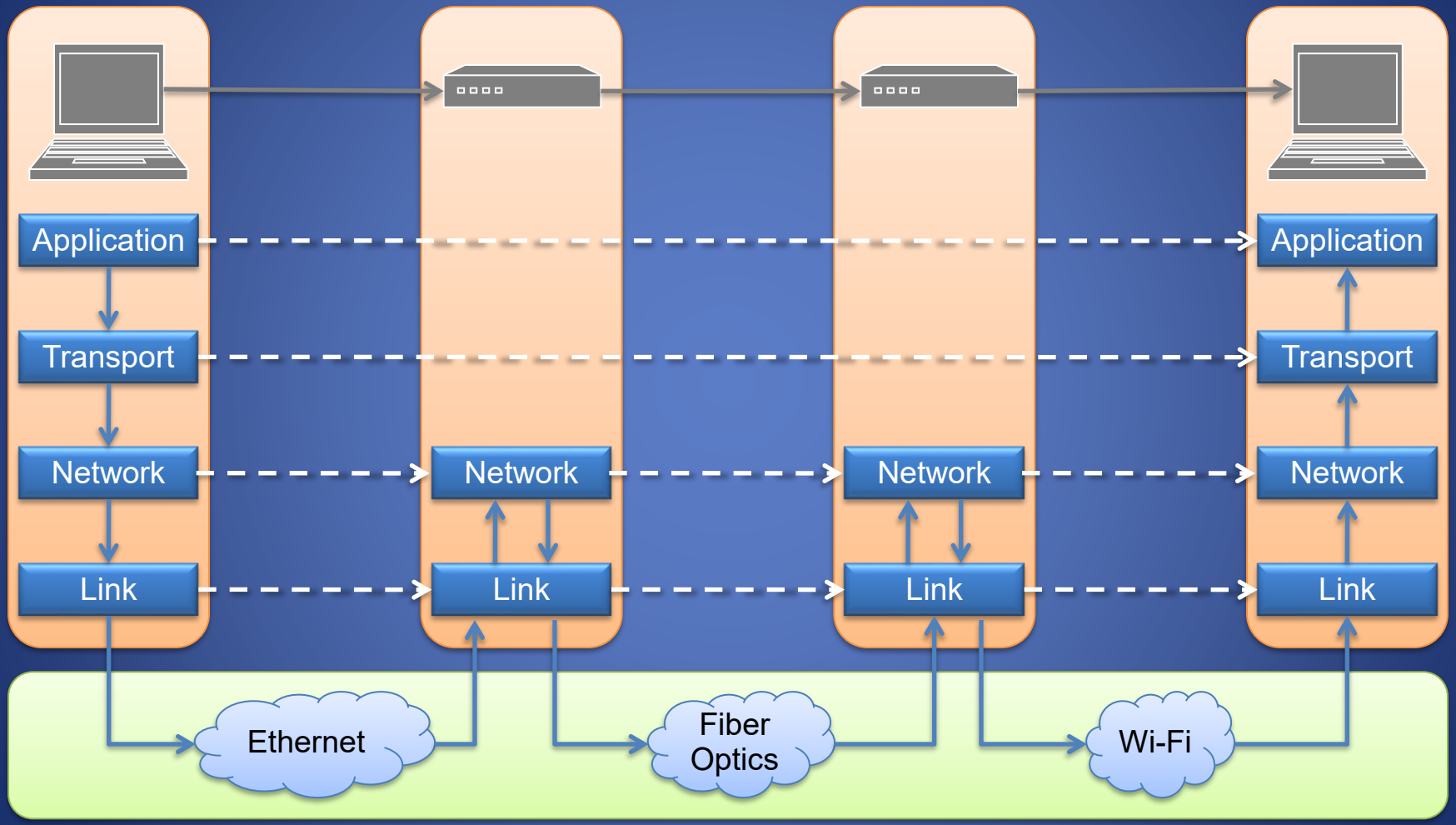
Encapsulation

- A packet typically consists of
  - Control information for addressing the packet: header and footer
  - Data: payload
- A network protocol N1 can use the services of another network protocol N2
  - A packet p1 of N1 is encapsulated into a packet p2 of N2
  - The payload of p2 is p1
  - The control information of p2 is derived from that of p1

Network Layers

- Network models typically use a stack of layers
  - Higher layers use the services of lower layers via encapsulation
  - A layer can be implemented in hardware or software
  - The bottommost layer must be in hardware
- A communication channel between two nodes is established for each layer
  - Actual channel at the bottom layer
  - Virtual channel at higher layers

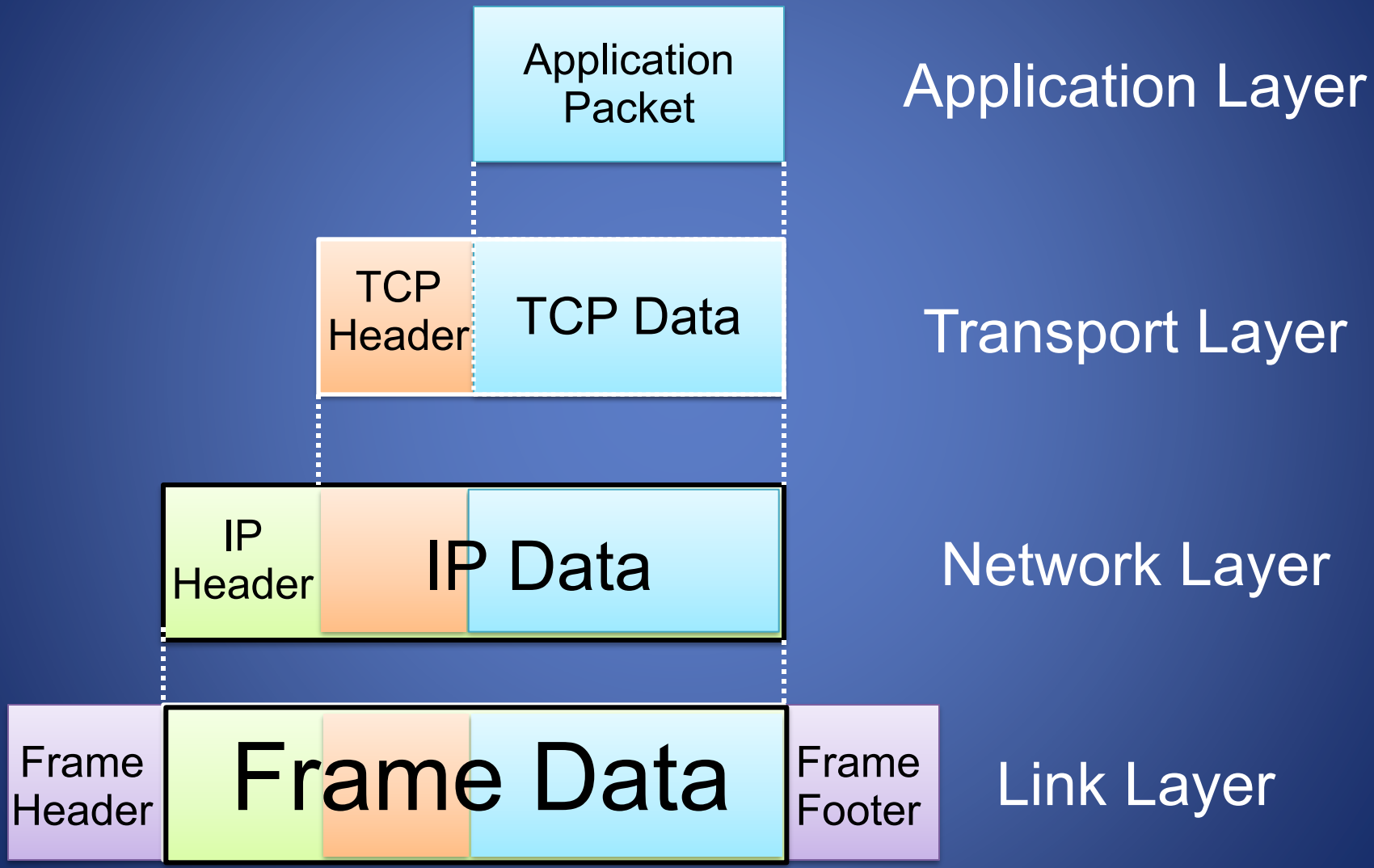
Internet Layers



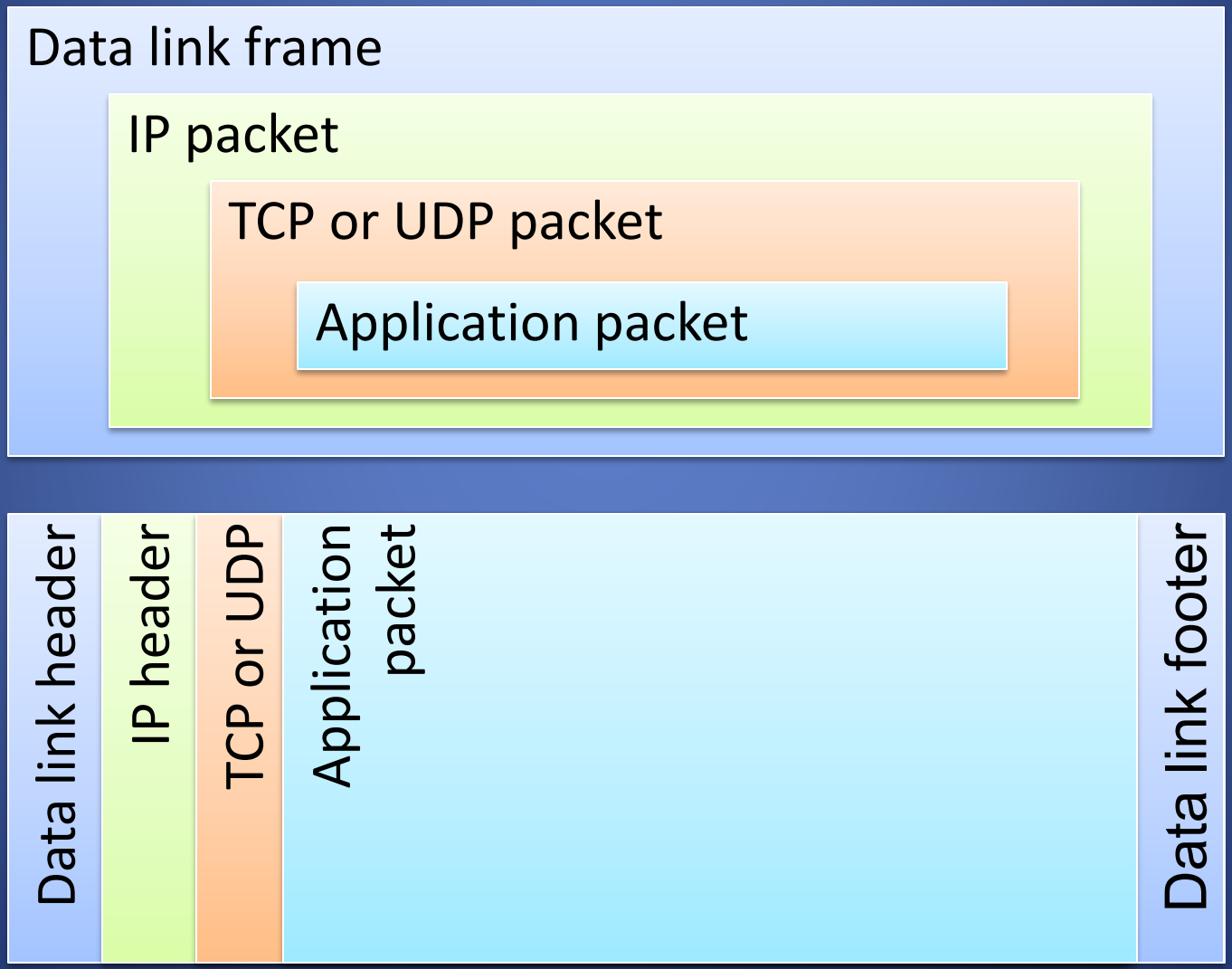
Intermediate Layers

- Link layer
  - Local area network: Ethernet, WiFi, optical fiber
  - 48-bit media access control (MAC) addresses
  - Packets called frames
- Network layer
  - Internet-wide communication
  - Best efforts
  - 32-bit internet protocol (IP) addresses in IPv4
  - 128-bit IP addresses in IPv6
- Transport layer
  - 16-bit addresses (ports) for classes of applications
  - Connection-oriented transmission layer protocol (TCP)
  - Connectionless user datagram protocol (UDP)
- Application layer
  - HTTP, DNS, SMTP, VoIP..

Internet Packet Encapsulation

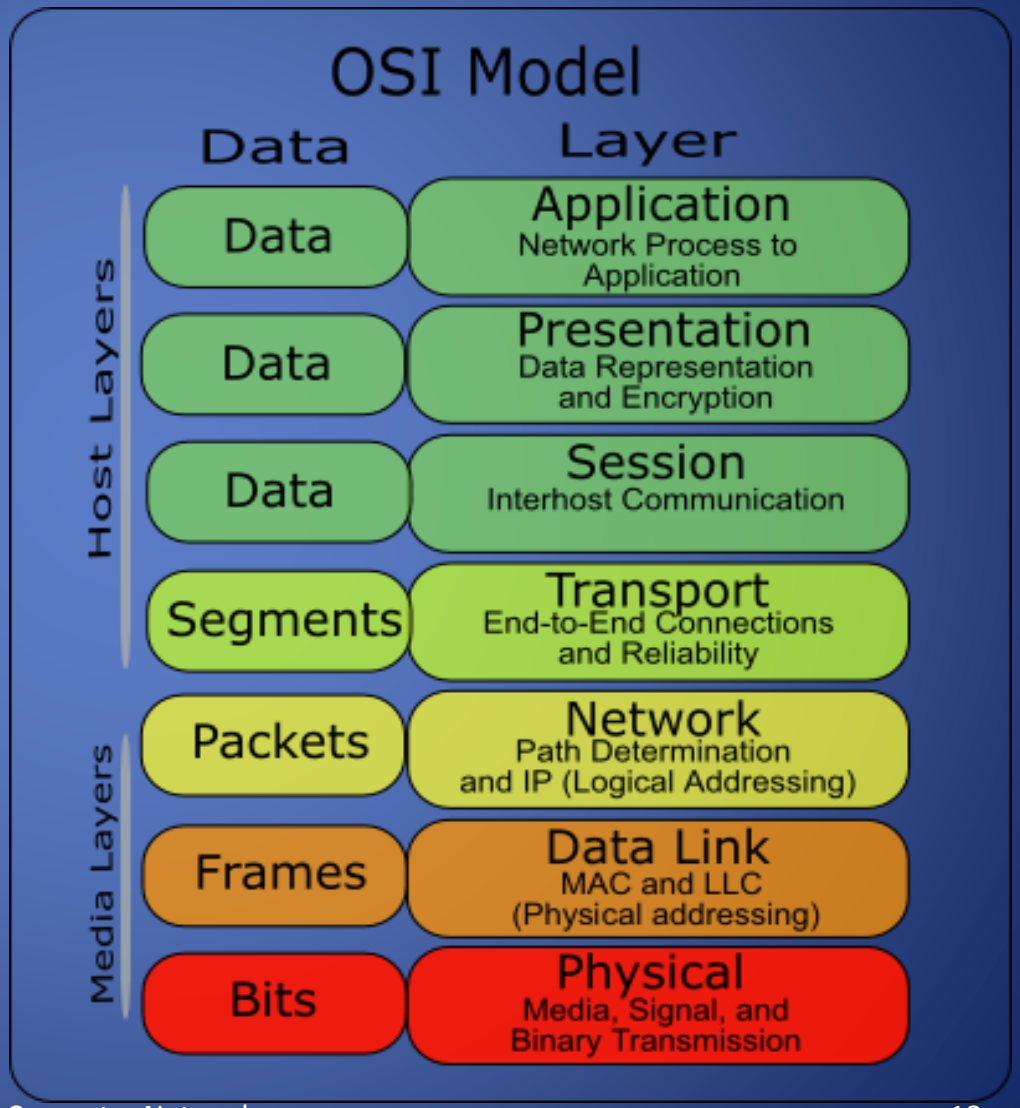


Internet Packet Encapsulation



The OSI Model

- The OSI (Open System Interconnect) Reference Model is a network model consisting of seven layers
- Created in 1983, OSI is promoted by the International Standard Organization (ISO)



Network Interfaces

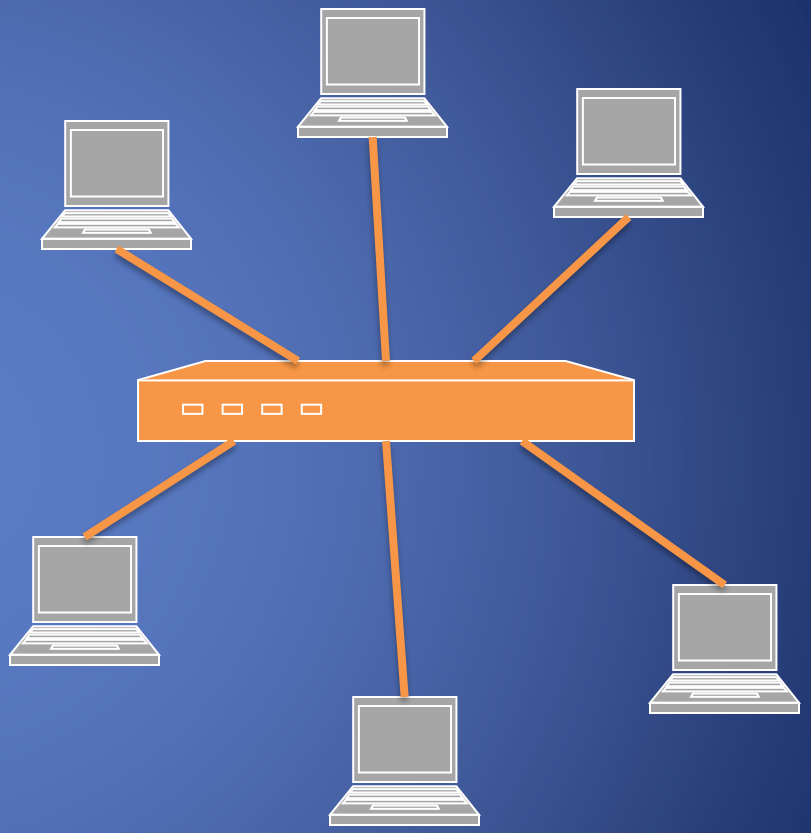
- Network interface: device connecting a computer to a network
  - Ethernet card
  - WiFi adapter
- A computer may have multiple network interfaces
- Packets transmitted between network interfaces
- Most local area networks, (including Ethernet and WiFi) broadcast frames
- In regular mode, each network interface gets the frames intended for it
- Traffic sniffing can be accomplished by configuring the network interface to read all frames

MAC Addresses

- Most network interfaces come with a predefined MAC (Media Access Control) address
- A MAC address is a 48-bit number usually represented in hex
  - E.g., 00-1A-92-D4-BF-86
- The first three octets of any MAC address are IEEE-assigned Organizationally Unique Identifiers
  - E.g., Cisco 00-1A-A1, D-Link 00-1B-11, ASUSTek 00-1A-92
- The next three can be assigned by organizations as they please, with uniqueness being the only constraint
  - Organizations can utilize MAC addresses to identify computers on their network
  - MAC address can be reconfigured by network interface driver software

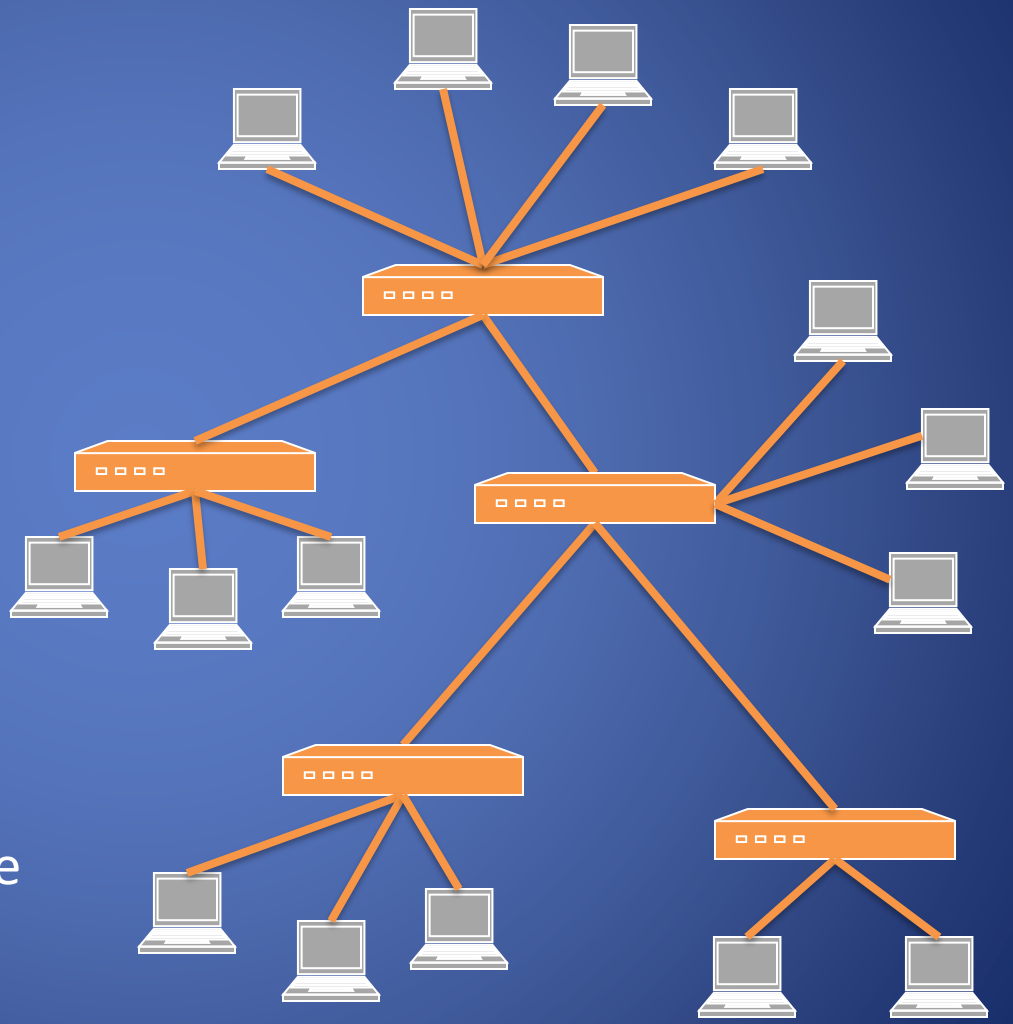
Switch

- Hub vs. Swith
- A switch is a common network device
  - Operates at the link layer
  - Has multiple ports, each connected to a computer
- Operation of a switch
  - Learn the MAC address of each computer connected to it
  - Forward frames only to the destination computer



Combining Switches

- Switches can be arranged into a tree
- Each port learns the MAC addresses of the machines in the segment (subtree) connected to it
- Fragments to unknown MAC addresses are broadcast
- Frames to MAC addresses in the same segment as the sender are ignored



MAC Address Filtering

- A switch can be configured to provide service only to machines with specific MAC addresses
- Allowed MAC addresses need to be registered with a network administrator
- A MAC spoofing attack impersonates another machine
  - Find out MAC address of target machine
  - Reconfigure MAC address of rogue machine
  - Turn off or unplug target machine
- Countermeasures
  - Block port of switch when machine is turned off or unplugged
  - Disable duplicate MAC addresses

Viewing and Changing MAC Addresses

- Viewing the MAC addresses of the interfaces of a machine
  - Linux: ifconfig
  - Windows: ipconfig /all
- Changing a MAC address in Linux
  - Stop the networking service: /etc/init.d/network stop
  - Change the MAC address: ifconfig eth0 hw ether <MAC-address>
  - Start the networking service: /etc/init.d/network start
- Changing a MAC address in Windows
  - Open the Network Connections applet
  - Access the properties for the network interface
  - Click “Configure …”
  - In the advanced tab, change the network address to the desired value
- Changing a MAC address requires administrator privileges

ARP

- The address resolution protocol (ARP) is a link-layer protocol that connects the network layer to the link layer by converting IP addresses to MAC addresses
- ARP works by broadcasting requests and caching responses for future use
- The protocol begins with a computer broadcasting a message of the form

  who has <IP address1> tell <IP address2>

- Then the machine with <IP address1> responds the requestor with an ARP reply as

  <IP address1> is <MAC address>

- The Linux and Windows command arp - a displays the ARP table

```
  Internet Address      Physical Address      Type
  128.148.31.1          00-00-0c-07-ac-00     dynamic
  128.148.31.15         00-0c-76-b2-d7-1d     dynamic
  128.148.31.71         00-0c-76-b2-d0-d2     dynamic
  128.148.31.75         00-0c-76-b2-d7-1d     dynamic
  128.148.31.102        00-22-0c-a3-e4-00     dynamic
  128.148.31.137        00-1d-92-b6-f1-a9     dynamic
```

ARP Spoofing

- The ARP table is updated whenever an ARP response is received
- Requests are not tracked
- ARP announcements are not authenticated
- Machines trust each other

- A rogue machine can spoof other machines

ARP Spoofing (ARP Poisoning)

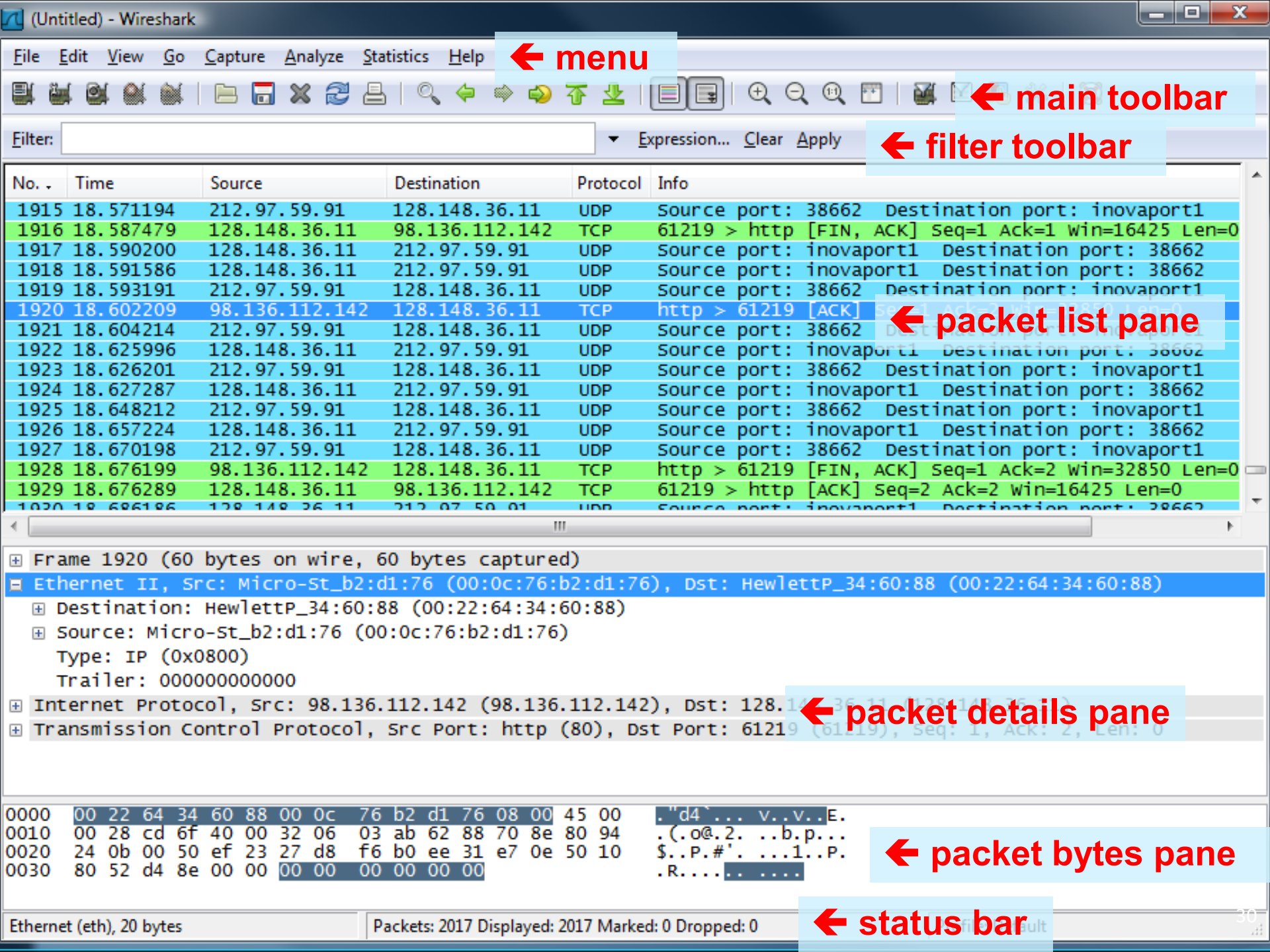
- According to the standard, almost all ARP implementations are stateless
  - An ARP cache updates every time that it receives an ARP reply… even if it did not send any ARP request!
  - It is possible to “poison” an ARP cache by sending gratuitous ARP replies
  - Using static entries solves the problem but it is almost impossible to manage!

Telnet Protocol (RFC 854)

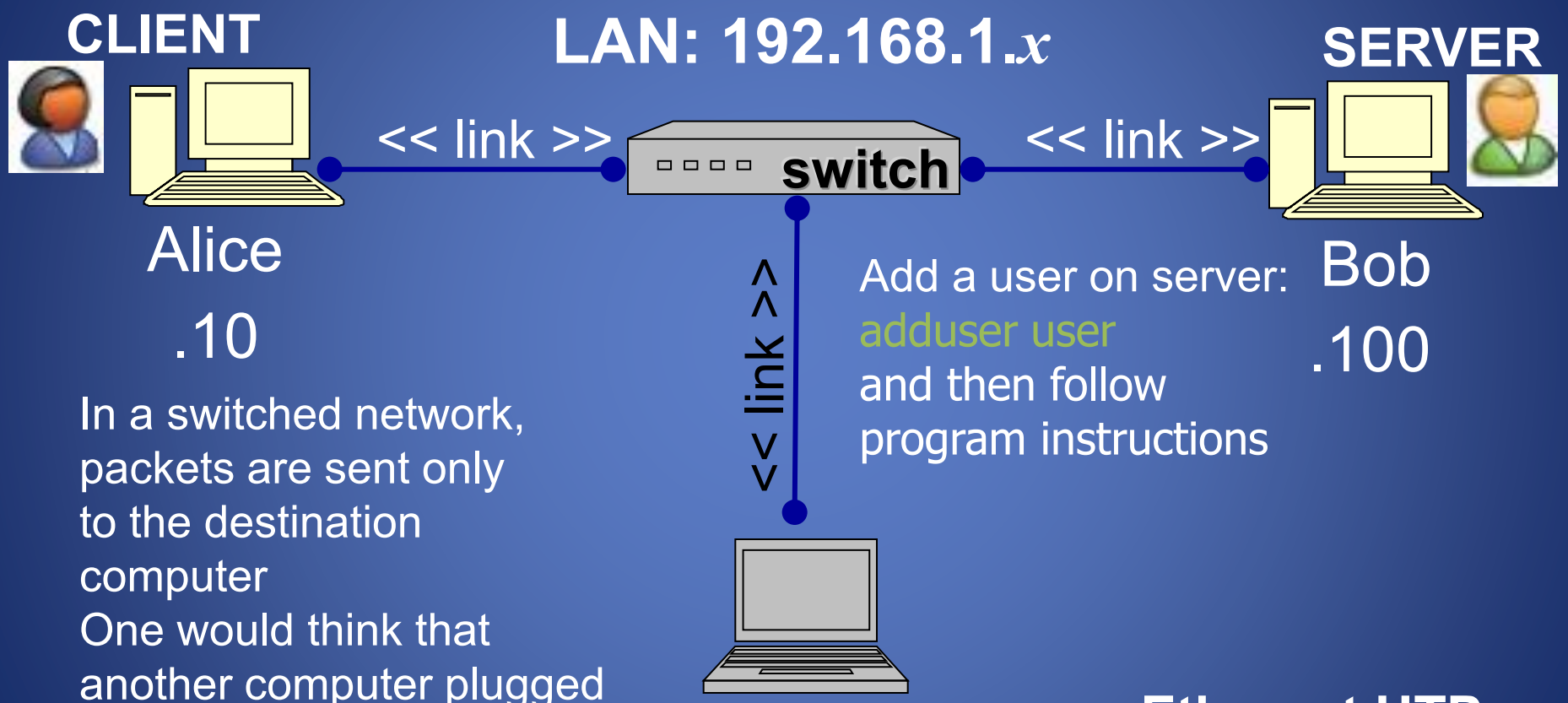
- Telnet is a protocol that provides a general, bi-directional, not encrypted communication
- `telnet` is a generic TCP client
  - Allows a computer to connect to another one
  - Provides remote login capabilities to computers on the Internet
  - Sends whatever you type
  - Prints whatever comes back
  - Useful for testing TCP servers (ASCII based protocols)

Wireshark

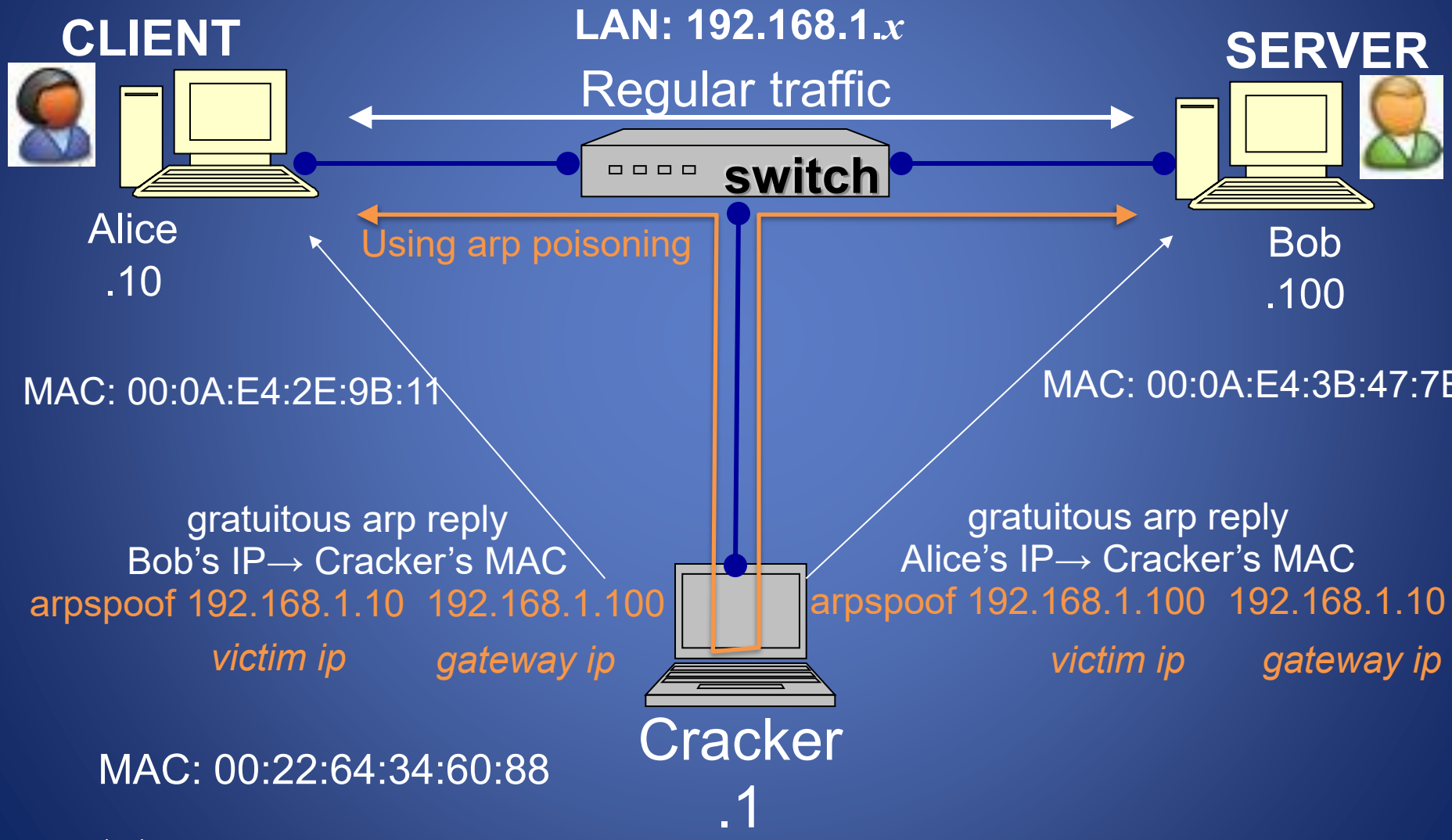
- Wireshark is a packet sniffer and protocol analyzer
  - Captures and analyzes frames
  - Supports plugins
- Usually required to run with administrator privileges
  - Setting the network interface captures traffic across the entire LAN segment and not just frames addressed to the machine
- Freely available 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
1920 18.602209 98.136.112.142 128.148.36.11 TCP http > 61219 [ACK]
1921 18.604214 212.97.59.91 128.148.36.11 UDP Source port: 38662
1922 18.625996 128.148.36.11 212.97.59.91 UDP
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.
Transmission Control Protocol, Src Port: http (80), Dst Port: 6121
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

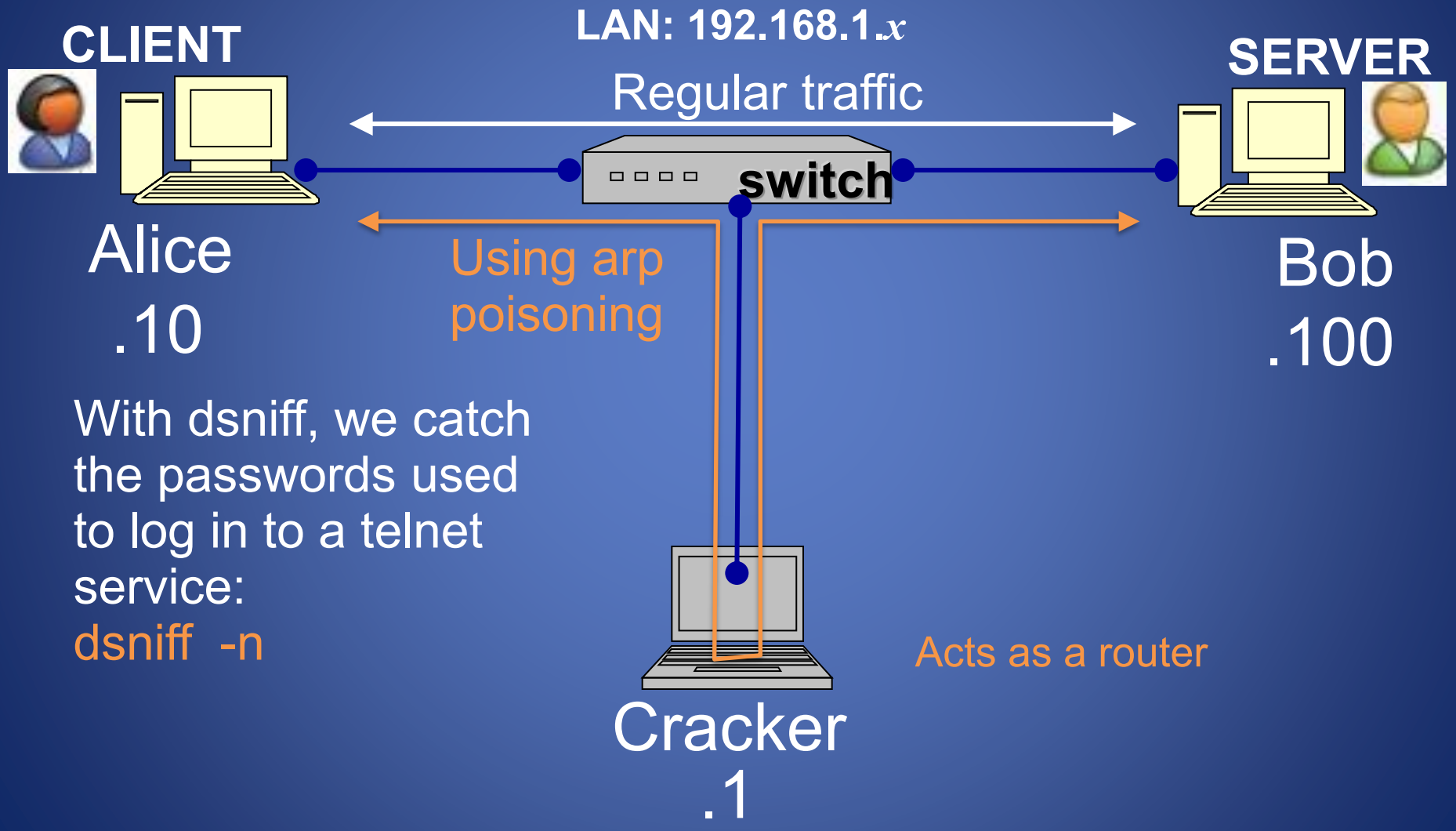
DEMO 1: Configuration using Telnet



Example 1: ARP Spoofing



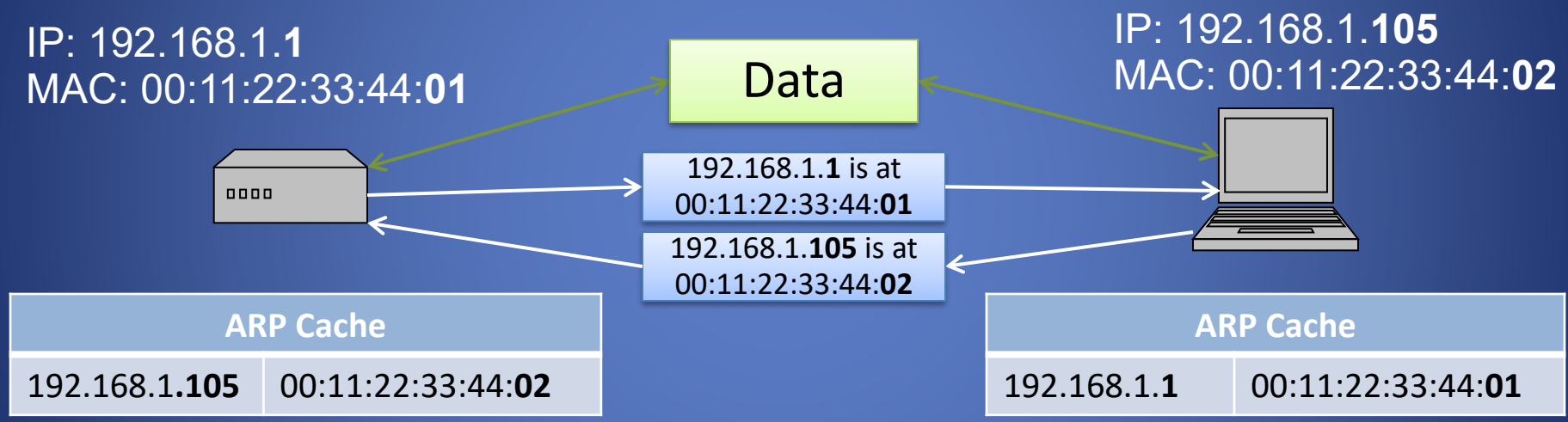
Example 1: catch telnet password



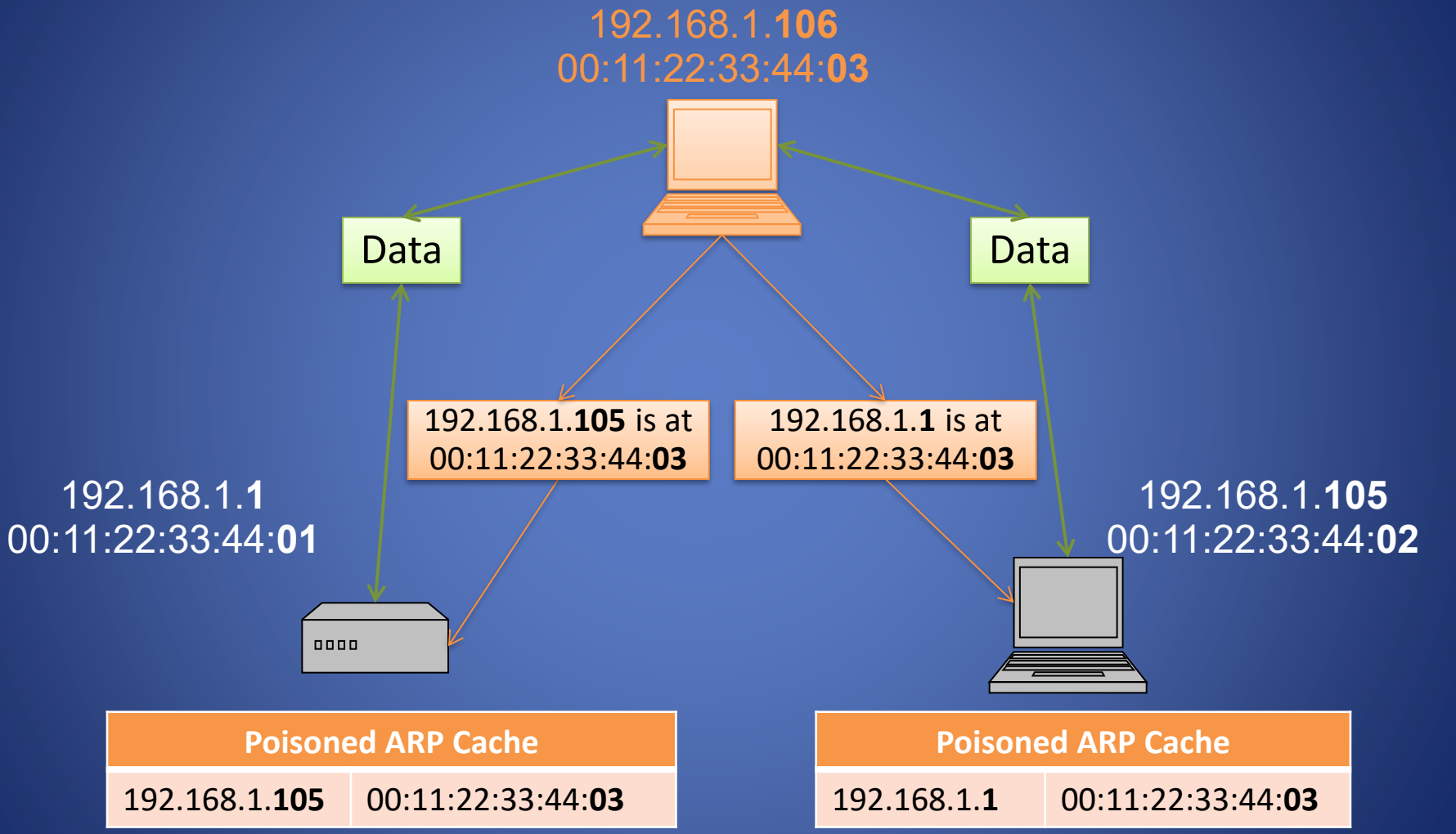
With dsniff, we catch the passwords used to log in to a telnet service:

dsniff -n

ARP Caches


IP: 192.168.1.1
MAC: 00:11:22:33:44:01
Data
IP: 192.168.1.105
MAC: 00:11:22:33:44:02
192.168.1.1 is at 00:11:22:33:44:01
192.168.1.105 is at 00:11:22:33:44:02
ARP Cache
192.168.1.105
00:11:22:33:44:02
ARP Cache
192.168.1.1
00:11:22:33:44:01

Poisoned ARP Caches



Poisoned ARP Cache	
192.168.1.**105**	00:11:22:33:44:**03**

Poisoned ARP Cache	
192.168.1.**1**	00:11:22:33:44:**03**