Networking with Wireshark

Cybersecurity

Networks 101 Day 2

Today’s Objectives

By the end of class, you will be able to:

- □ Describe the flow of typical HTTP conversations at the application layer.
- □ Describe the flow of DNS conversations at the application layer.
- □ Describe the flow of typical TCP conversations.

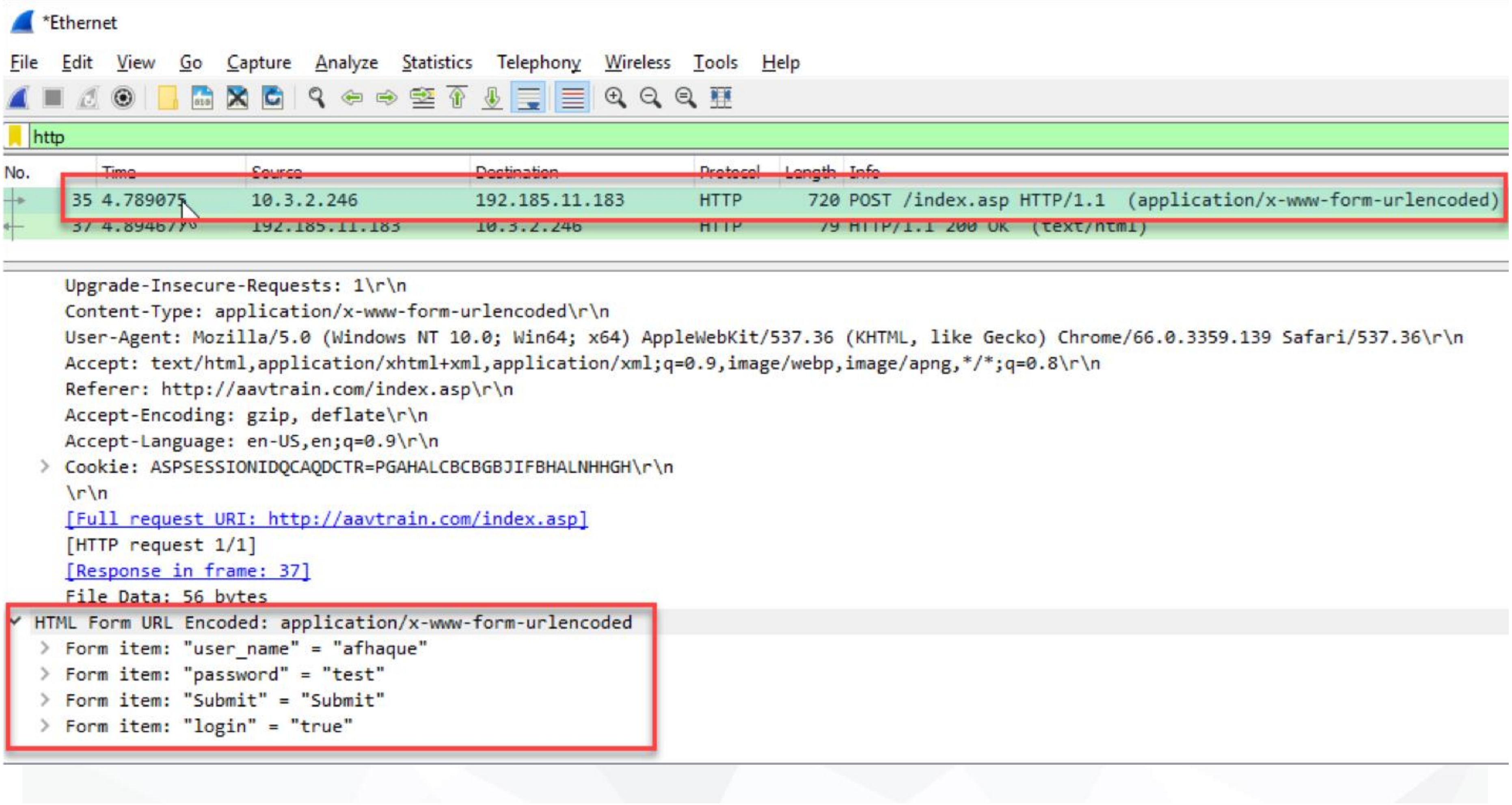
Activity: Leaky HTTP Traffic

In this partner activity, you’ll use Wireshark to retrieve a user’s username and password being communicated through an insecure website.

Suggested Time:
8 Minutes

Times Up! Let’s Review.

Leaky HTTP Traffic



```
http
```

```
No.    Time          Source           Destination      Protocol  Length Info
    35 4.789075      10.3.2.246       192.185.11.183   HTTP         720 POST /index.asp HTTP/1.1  (application/x-www-form-urlencoded)
    37 4.894677      192.185.11.183   10.3.2.246       HTTP          79 HTTP/1.1 200 OK  (text/html)
```

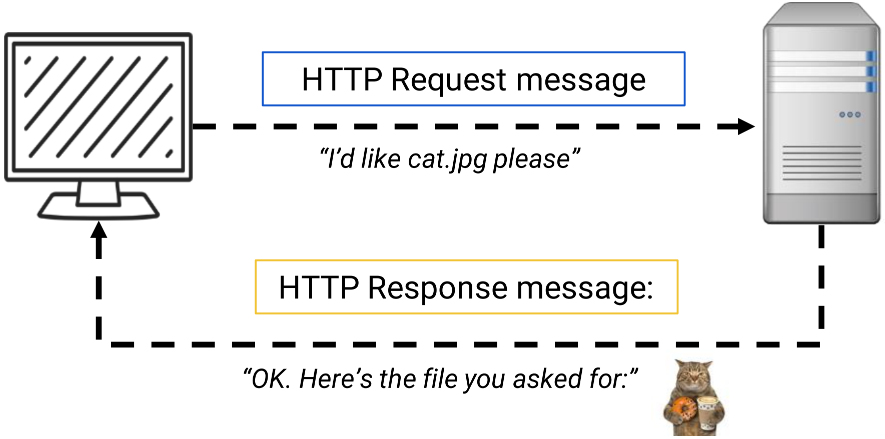
```
    Upgrade-Insecure-Requests: 1\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.139 Safari/537.36\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n
    Referer: http://aavtrain.com/index.asp\r\n
    Accept-Encoding: gzip, deflate\r\n
    Accept-Language: en-US,en;q=0.9\r\n
  > Cookie: ASPSESSIONIDQCAQDCTR=PGAHALCBCBGBJIFBHALNHHGH\r\n
    \r\n
    [Full request URI: http://aavtrain.com/index.asp]
    [HTTP request 1/1]
    [Response in frame: 37]
    File Data: 56 bytes
v HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "user_name" = "afhaque"
  > Form item: "password" = "test"
  > Form item: "Submit" = "Submit"
  > Form item: "login" = "true"
```

HTTP

HTTP: Hypertext Transfer Protocol

HTTP is an application-layer protocol designed primarily for communication between web browsers and web servers, and uses typical client/server architecture.

```
GET /cat.jpg HTTP/1.1
```



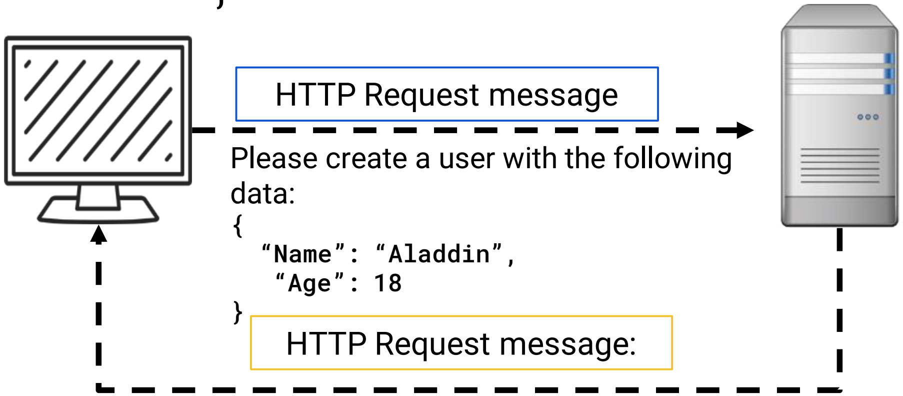
HTTP Request message

"I'd like cat.jpg please"

HTTP Response message:

"OK. Here's the file you asked for:"

```
POST /userHTTP/1.1
...
{
  "Name": "Aladdin",
   "Age": 18
}
```



HTTP Request message

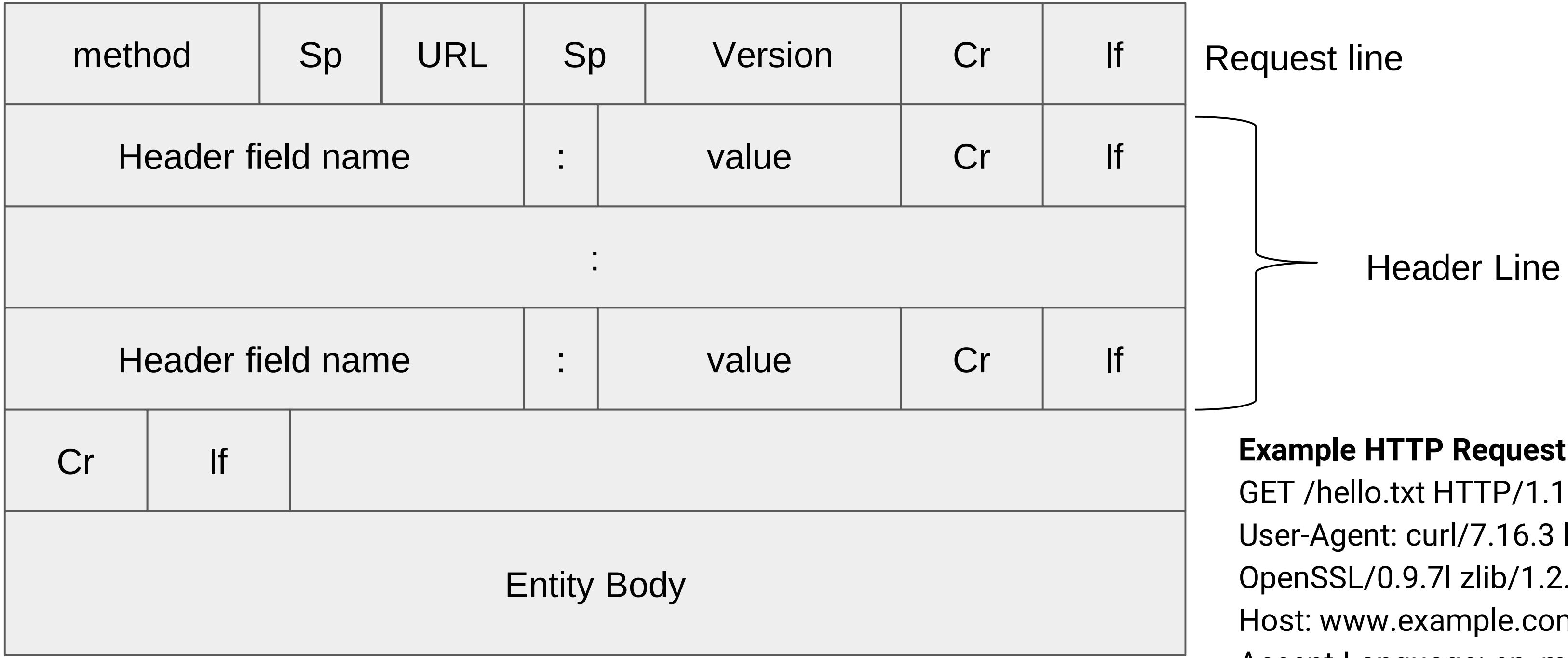
Please create a user with the following data:

```
{
  "Name": "Aladdin",
   "Age": 18
}
```

HTTP Request message:

"We created it"

HTTP Request



Example HTTP Request:
GET /hello.txt HTTP/1.1
User-Agent: curl/7.16.3 libcurl/7.16.3
OpenSSL/0.9.7l zlib/1.2.3
Host: www.example.com
Accept-Language: en, mi

HTTP Status Codes

Type	Status Codes	Examples
Informational	1xx	100: Continue, 101: Switching Protocol
Success	2xx	200: OK, 201: Created, 202: Accepted
Redirection	3xx	300: Multiple Choices, 301: Moved Permanently, 302: Found
Client Error	4xx	400: Bad Request, 403: Forbidden, 404: Not Found, 422: Unprocessable Entity
Server Error	5xx	500: Internal Server Error, 503: Service Unavailable

HTTP vs HTTPS

HTTPs (HTTP Secure) uses an SSL certificate (TLS) to encrypt data before sending, and decrypt upon arrival.

Activity: Analyzing HTTP

In this activity, you will look at HTTP conversations to reverse-engineer the HTTP protocol.

Suggested Time:
20 Minutes

Times Up! Let’s Review.

Analyzing HTTP

Activity: The Search for Something Cool

In this activity, you will open a previously captured file and then tasked with importing a pcap file and using display filters to retrace a user's browsing history.

Instructions sent via Slack.

Suggested Time:
20 Minutes

Times Up! Let’s Review.

The Search for Something Cool

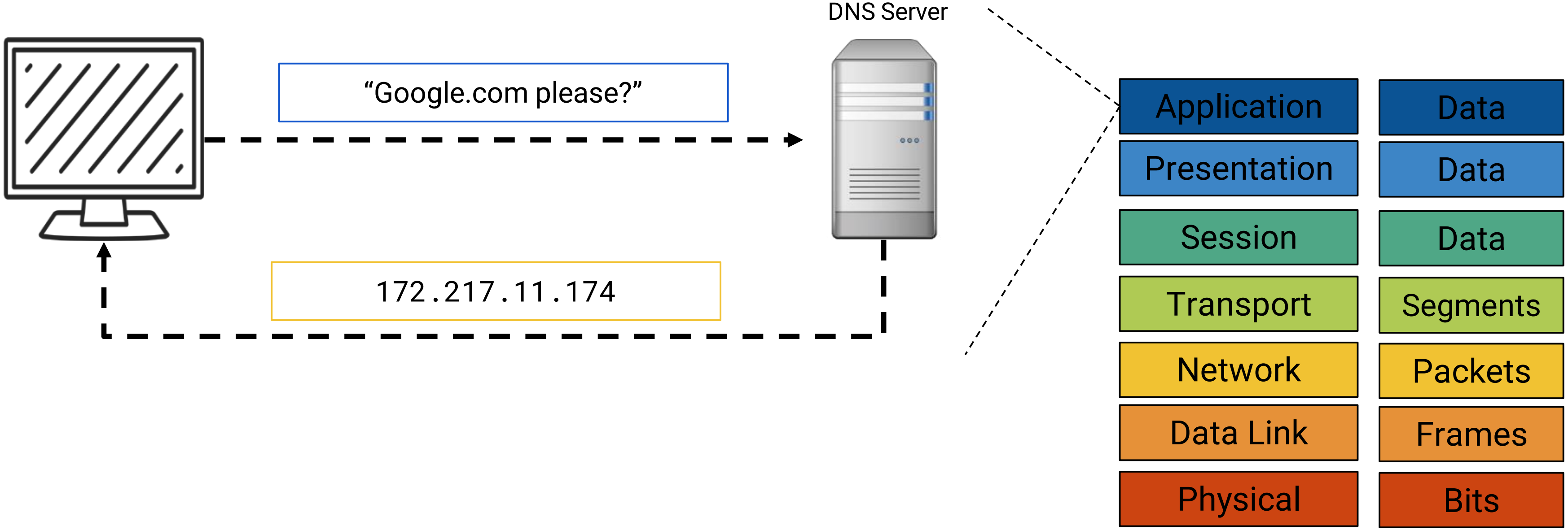
Today’s Objectives Checkout

By the end of class, you will be able to:

- ☑ Describe the flow of typical HTTP conversations at the application layer.
- □ Describe the flow of DNS conversations at the application layer.
- □ Describe the flow of typical TCP conversations.

DNS

Domain Name System (DNS) is an application-layer protocol designed to translate domain names into IP addresses.



Application	Data
Presentation	Data
Session	Data
Transport	Segments
Network	Packets
Data Link	Frames
Physical	Bits

Domain Names

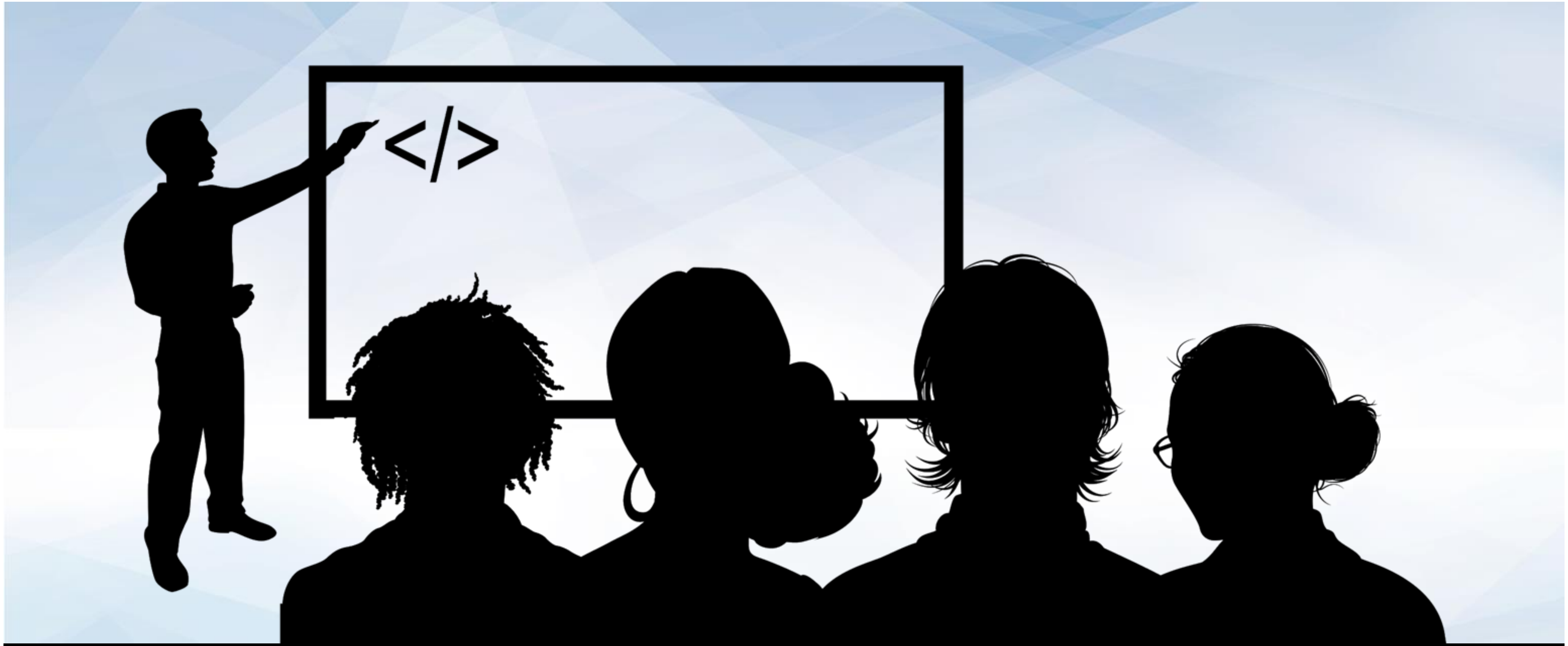
nslookup is a command line tool used for manual DNS resolution.

```
$ nslookip google.com
Server: RAC2V1S
Address: 192.168.1.1

Non-authoritative answer:
Name: google.com
Addresses:
2607:f8b0:4004:800:200e

172.217.15.110
```

DNS Record Types

DNs allows you to query for more than just domain → IP Address. Record types:

- A record: IPv4 address from a hostname query
- AAAA record: IPv6 address from a hostname query
- MX record: mail server for the domain
- CNAME: alias to the domain name
- NS record: nameserver of the domain
- PTR record: hostname from an IP address



Instructor Demonstration

DNS in Wireshark

Activity: Wireshark DNS Analysis

In this activity, you will look at pcap files and identify DNS traffic.

Suggested Time:
10 Minutes

Your Turn: Analyzing DNS in Wireshark

Instructions:

Open the dns-1.pcap file.

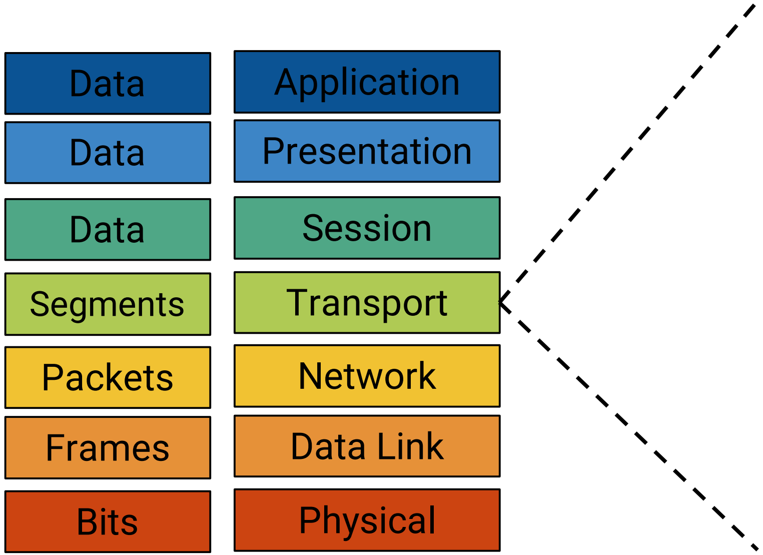
- □ This file only contains DNS replies. How many DNS requests were there?
- □ When the user asked for assets.espn.go.com, what happened?
- □ What is/are the IP address(es) for a1.espncdn.com?

Open the dns-2.pcap file.

- □ This capture contains an attempted query, but something went wrong.
- □ What happened?
- □ Which flag in the packet reveals what went wrong?
- □ The request went to 8.8.8.8. Did the response come directly from 8.8.8.8?

Transport Layer Protocol

Transport Layer Protocol



Data	Application
Data	Presentation
Data	Session
Segments	Transport
Packets	Network
Frames	Data Link
Bits	Physical

Transport Layer is responsible for end-to-end communication over a network and sending data to the appropriate application.

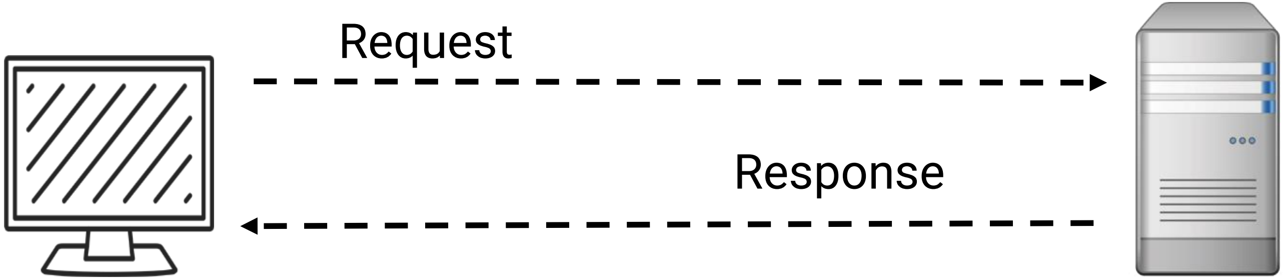
Works with session layer to manage connection between application on different machines.

Two most common protocols of transport is UDP and TCP

UDP

UDP is stateless or "connectionless," and it's used when we don't care whether we get all of our data. It is also typically faster

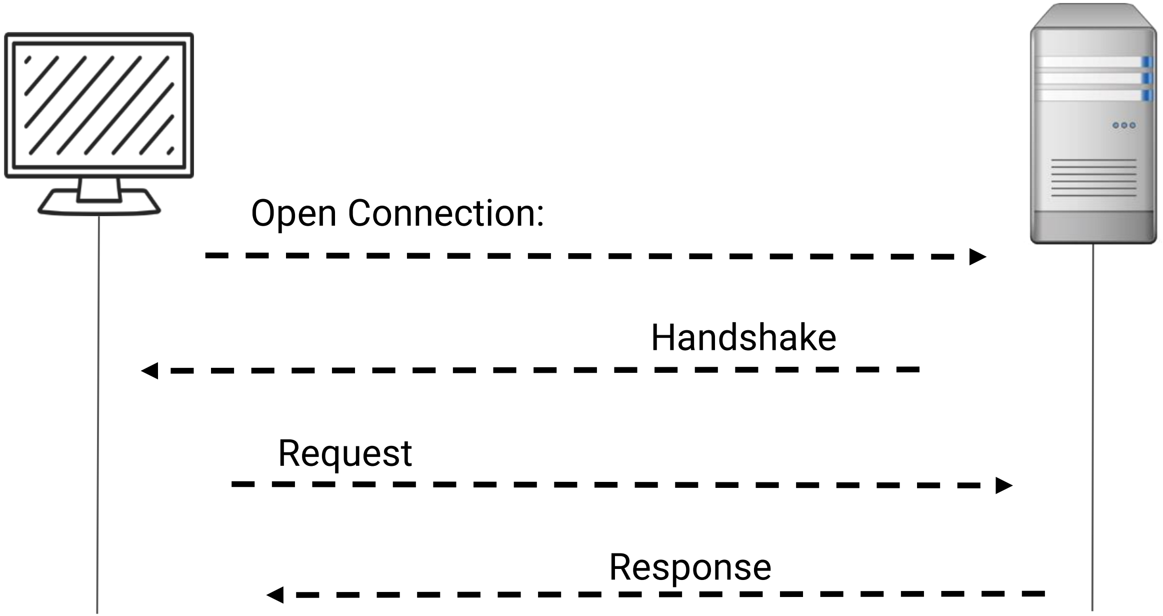
UDP Request / Response Paradigm:



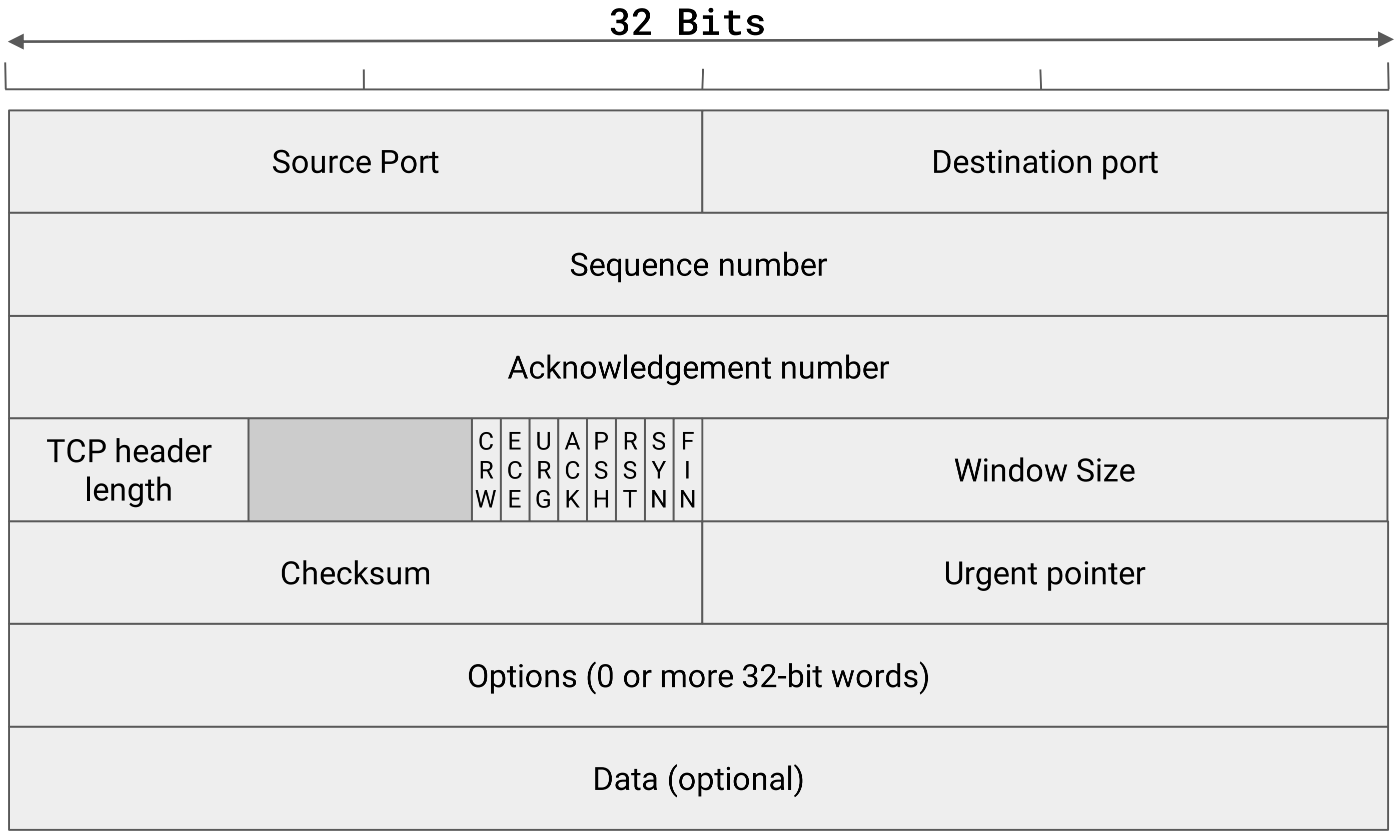
TCP

TCP is used when all transmitted data must be received. Used with familiar protocols such as HTTP, FTP, SSH, and SMTP.

TCP Handshake Paradigm:



TCP Headers and Flags



32 Bits

Source Port

Destination port

Sequence number

Acknowledgement number

TCP header length

CRW

ECE

URG

ACK

PSH

RST

SYN

FIN

Window Size

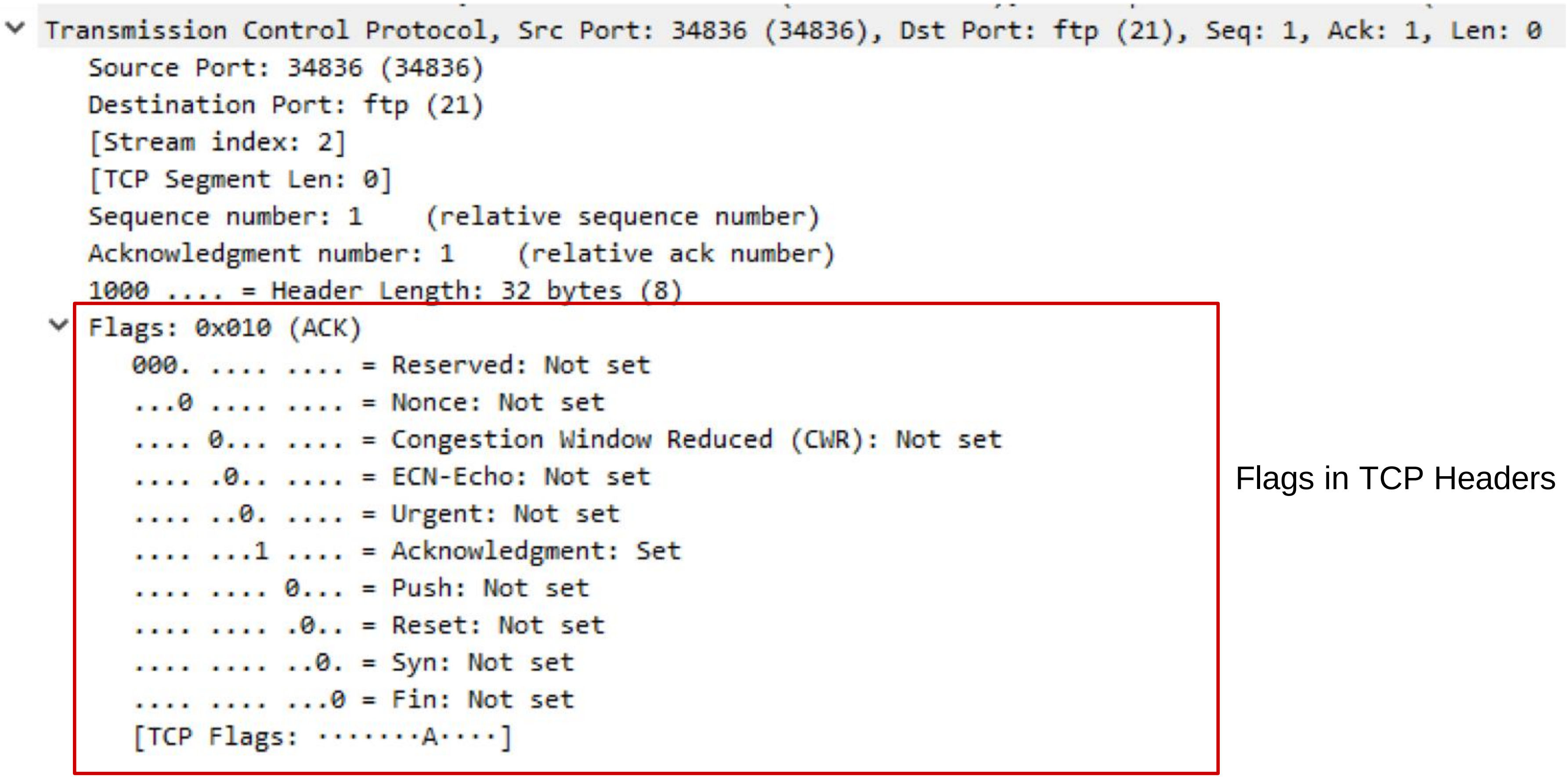
Checksum

Urgent pointer

Options (0 or more 32-bit words)

Data (optional)

TCP Headers and Flags



```
Transmission Control Protocol, Src Port: 34836 (34836), Dst Port: ftp (21), Seq: 1, Ack: 1, Len: 0
  Source Port: 34836 (34836)
  Destination Port: ftp (21)
  [Stream index: 2]
  [TCP Segment Len: 0]
  Sequence number: 1    (relative sequence number)
  Acknowledgment number: 1    (relative ack number)
  1000 .... = Header Length: 32 bytes (8)
  Flags: 0x010 (ACK)
     000. .... .... = Reserved: Not set
     ...0 .... .... = Nonce: Not set
     .... 0... .... = Congestion Window Reduced (CWR): Not set
     .... .0.. .... = ECN-Echo: Not set
     .... ..0. .... = Urgent: Not set
     .... ...1 .... = Acknowledgment: Set
     .... .... 0... = Push: Not set
     .... .... .0.. = Reset: Not set
     .... .... ..0. = Syn: Not set
     .... .... ...0 = Fin: Not set
     [TCP Flags: ·······A····]
```

Flags in TCP Headers

TCP Headers and Flags

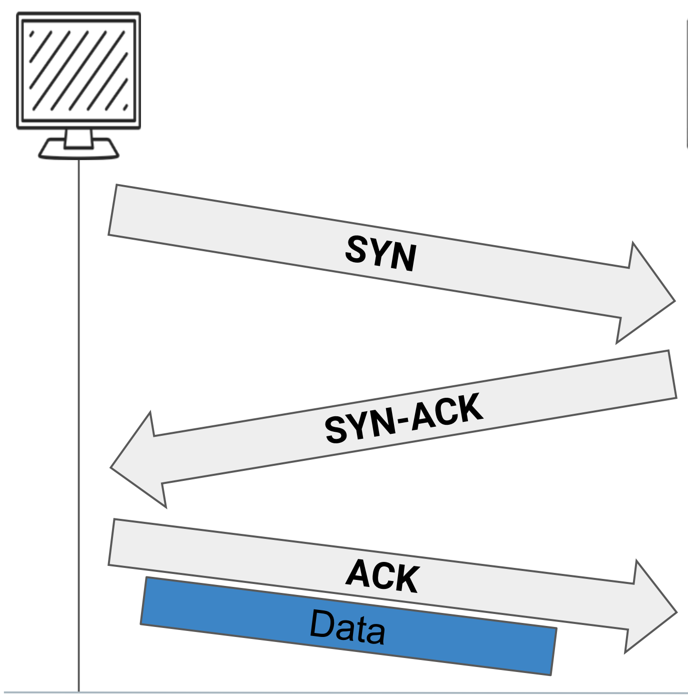
Flags to indicate what kind of TCP message is contained within:

- SYN – Synchronization: first step in establishing handshake between hosts
- ACK – Acknowledgement: used to acknowledge successful receipt of a packet
- PSH – Push: tell recipient to process this packet as it is received, don’t buffer it
- RST – Reset: sent when a packet was sent that was unexpected
- FIN – Synchronization: first step in establishing handshake between hosts

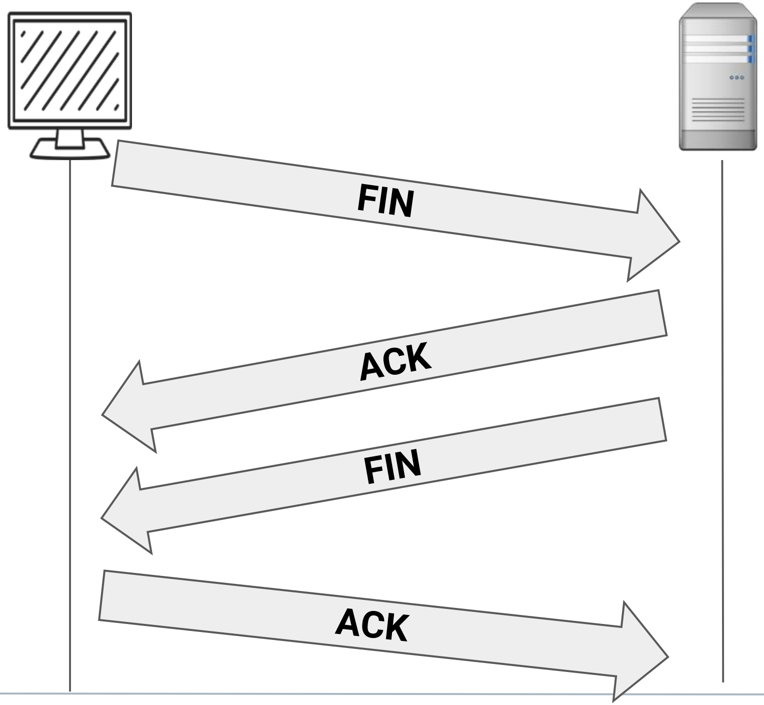
TCP Handshake

Connection-Oriented, 3-way handshake, 4-way close

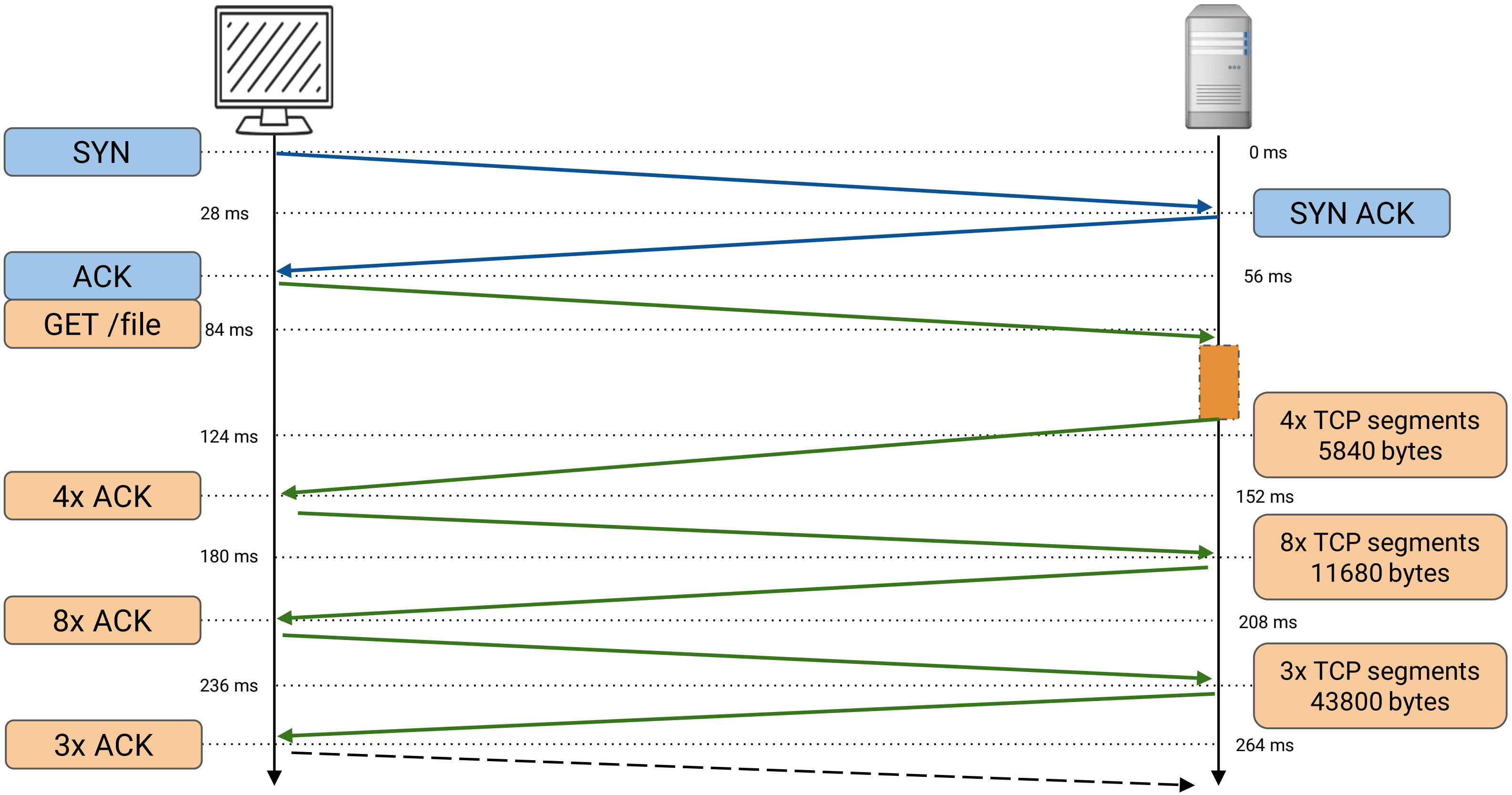
Setup: SYN, SYN-ACK, ACK



Teardown: FIN, ACK, FIN, ACK



TCP Handshake



TCP vs UDP Comparison Example

Scenario: you want your friend’s toy.

TCP:

1. You call your friend’s phone number.
2. He picks up and says, “Howdy, Buzz!”
3. You say “Hey, Woody!”

3 - Way Handshake

4. You ask for the toy.
5. He sends you the toy

UDP:

1. You call and leave a voicemail saying you want his toy.

1. The toy may arrive in the mail.

Activity: Explain / Draw the Process

In this activity, you will practice explaining the basics of what happens in layers 4-7 of the OSI Model and then draw the process of request/response for an HTTP page through the network.

Instructions sent via Slack.

Suggested Time:
10 Minutes

Your Turn: Analyzing DNS in Wireshark

Instructions:

Draw the process of request/response for an HTTP page through the network in the context of the OSI model, labelling:

- ☐ The protocol(s) used at each layer
- ☐ How the protocol works at that stage of the process
- ☐ The format of the data at that point in the layer

When confident in your drawing, compare with your partner’s drawing.

Try to explain your diagram, and note any differences between the two.

Act out the TCP communication and handshake process with your partner.

Bonus: Do the same for DNS.

Times Up! Let’s Review.

Explaining and Drawing the Process

Activity: Digging Into TCP Communication

In this activity, students will observe a pcap with TCP communication and answer a few questions about the file.

Instructions sent via Slack.

Suggested Time:
15 Minutes

Your Turn: Digging Into TCP Communications

Instructions:

- □ Open the tcp.pcapng file with Wireshark
- □ Filter for TCP packets only.
- □ Find all TCP SYN packets.
- □ Find all TCP FIN packets.
- □ Filter for a single TCP stream using the "FTP" protocol.
- □ Find the 3-way handshake sequence.
- □ Find the TCP teardown sequence.
- □ What are the source/destination IP addresses and ports?

Times Up! Let’s Review.

Digging into TCP Communications

Today’s Objectives

By the end of class, you will be able to:

- ☑ Describe the flow of typical HTTP conversations at the application layer.
- ☑ Describe the flow of DNS conversations at the application layer.
- ☑ Describe the flow of typical TCP conversations.