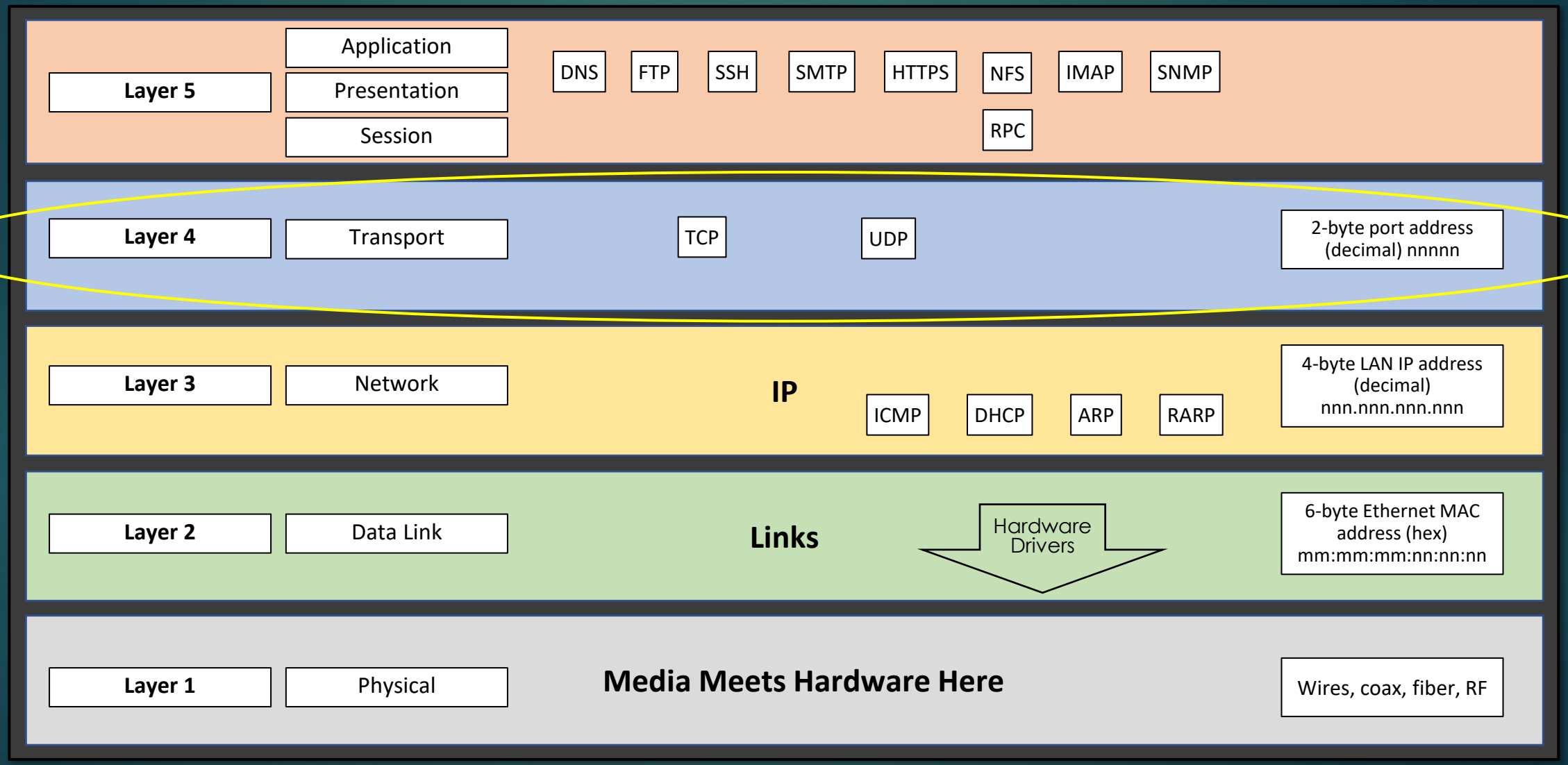
IT/CYBR 4323, Chapter 16:

UDP Transport

- Mr. Donald Privitera, Mr. Gennadiy Kemelmakher
- Kennesaw State University
- Marietta Campus
- College of Computing and Software Engineering
- IT Department
- dprivit2@kennesaw.edu, gkemelma@kennesaw.edu

5-Layer TCP/IPv4 Network Stack



Port Numbers Video

Peter L Dordal, An Introduction to Computer Networks

https://youtu.be/RDotMcs0Erg

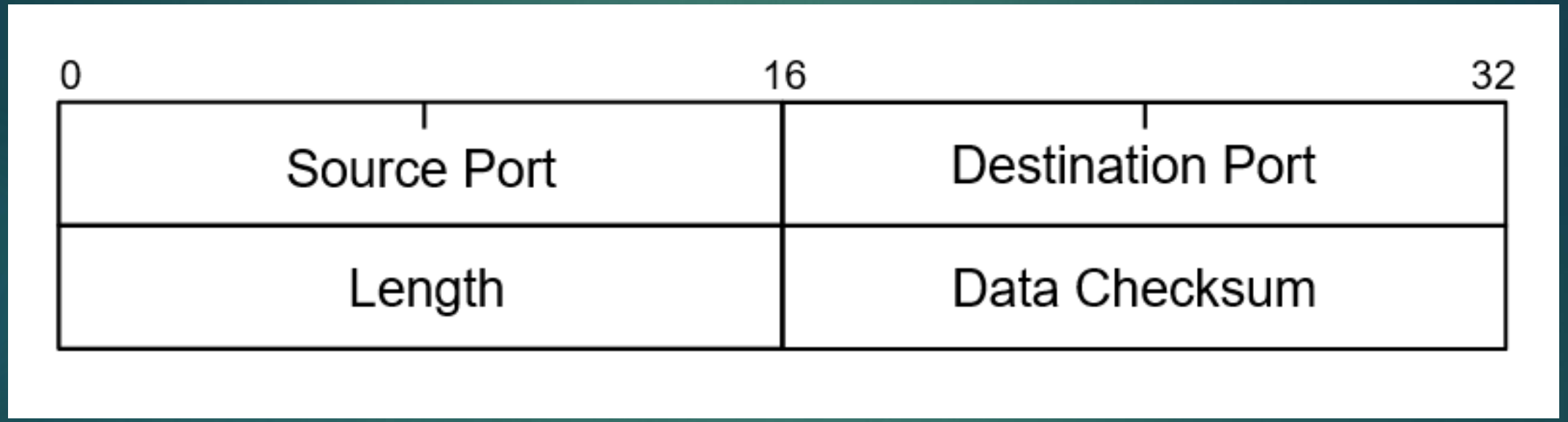
User Datagram Protocol (UDP)

- UDP is an unreliable transport protocol. Why does it even exist? What is the difference between plain-old IP and UDP?
- IP = 151.101.1.67
- UDP = 151.101.1.67:80
- :80 is the port address or **port number**. When you enter a URL into a browser, the browser automatically adds :80 if you do not specify a port number. In UDP/IP, a port number is ALWAYS required.
- If port numbers did not exist, a server could not practically communicate since there would be only 1 communication channel.
- Port numbers are 16-bits which allow up to a theoretical 65535 communication channels to exist.

So, why UDP?

- No persistent connection
- No packet counting and re-transmission
- = very little overhead making it FAST!
- Great for time critical packets like video and audio
- Used as the underpinning for “streaming”
- Lightweight

The UDP header – a model of simplicity



Peter L Dordal, An Introduction to Computer Networks

UDP

- <host,port> pair is called a **socket**
- UDP is **unreliable**
- UDP is **unconnected** or **stateless**
- UDP works well for video and voice which is **loss tolerant** where a few packet losses are insignificant and **delay-intolerant** where late packets are useless
- UDP forms the underpinning for more sophisticated protocols like, **Real-Time Transport Protocol (RTP)**

Problems

- UDP allows flooding attacks
- UDP allows **traffic amplification**

UPD metalayers

- UDP works by itself in the scope of a local LAN.
- Beyond local LAN scope, many other transport protocols are built on top of UDP.
- Therefore, UDP can be thought of as a sub-layer for some protocols.
- The following are some protocols built upon UDP:
  - DNS, RTP, SNMP, RIP, DHCP, QUIC, UDT, DCCP, RPC, TFTP, …

How many protocols are there?

https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Python example of UDP in action

```
import socket

print ("Launching UDP server")

# get the IP address of the server so we can open a port to UDP data

UDP_IP = input("Enter server address: ")
UDP_PORT = 5005
BUFFER_SIZE = 10

sock = socket.socket(socket.AF_INET, # Internet
                     socket.SOCK_DGRAM) # UDP
sock.bind((UDP_IP, UDP_PORT))

print("Listening on",UDP_IP,":",UDP_PORT)

a = ""

# keep doing this until the client sends a ~ character
while a != "~":
    try:
        data, addr = sock.recvfrom(BUFFER_SIZE)

        # convert bytearray into a string
        a = str(data)

        # remove the first 2 characters and last character
        a = a[2:-1]

        # print the character immediately and don't do a newline
        print(a,end="",flush=True)

        # print a newline and client communication data on if a . is received
        if ( a == "." ):
            print(" < from",addr,">")
    except:
        print("socket receive error")
```

```
import socket

print("Launching UDP client")

# this opens a port on the specified server

UDP_IP = input("Enter server IP address: ")
UDP_PORT = 5005
MESSAGE = ""

print("UDP target IP: %s" % UDP_IP)
print("UDP target port: %s" % UDP_PORT)


try:
    sock = socket.socket(socket.AF_INET,    # Internet
                         socket.SOCK_DGRAM)  # UDP

    # keep doing this until the user enters a ~ character
    while MESSAGE != "~":
        # get a string to send from the user
        MESSAGE = input("Message? ")
        msglen = len(MESSAGE)
        if (msglen > 0): # send the string one character at a time to test UDP reliability
            for x in range(msglen):
                sock.sendto(bytearray(MESSAGE[x],encoding="ascii"),
                            (UDP_IP, UDP_PORT))

            # make a note of what we sent and where
            print("Message <",MESSAGE,"> sent to ",UDP_IP,":",UDP_PORT)
except:
    print("socket error")
```

UDP Compared to TCP

User Datagram Protocol (UDP)	Transmission Control Protocol (TCP)
Connectionless	Connection Oriented
Unreliable	Reliable
Not ordered	Ordered
Lightweight	Heavy
Smaller headers & packets	Larger headers & packets
Datagrams	Streaming
No congestion control	Congestion control (rate & errors)
Unicast, multicast, broadcast	Unicast

Peter L Dordal, An Introduction to Computer Networks

UDP vs TCP Video

Peter L Dordal, An Introduction to Computer Networks

https://youtu.be/Vdc8TCESIg8