Networking Review (Protocols)

UT CS361S – Network Security and Privacy

Spring 2021

Lecture Notes

What is a Protocol?

- The set of rules that govern the interaction of two or more parties
- In networking, it defines how two nodes communicate
  - When
  - What *(including message structure)*
  - How
- ***Certain outcomes are guaranteed when the rules are followed***

Overloaded Term

- Actually, a protocol often refers to two separate things
- **FIRST**, the rules/specification referred to on the previous slide
- **SECOND**, the computer module that *implements* the rules

Common Contemporary Protocols

- HTTP – HyperText Transfer Protocol
- IP – Internet Protocol
- SMTP – Simple Mail Transport Protocol

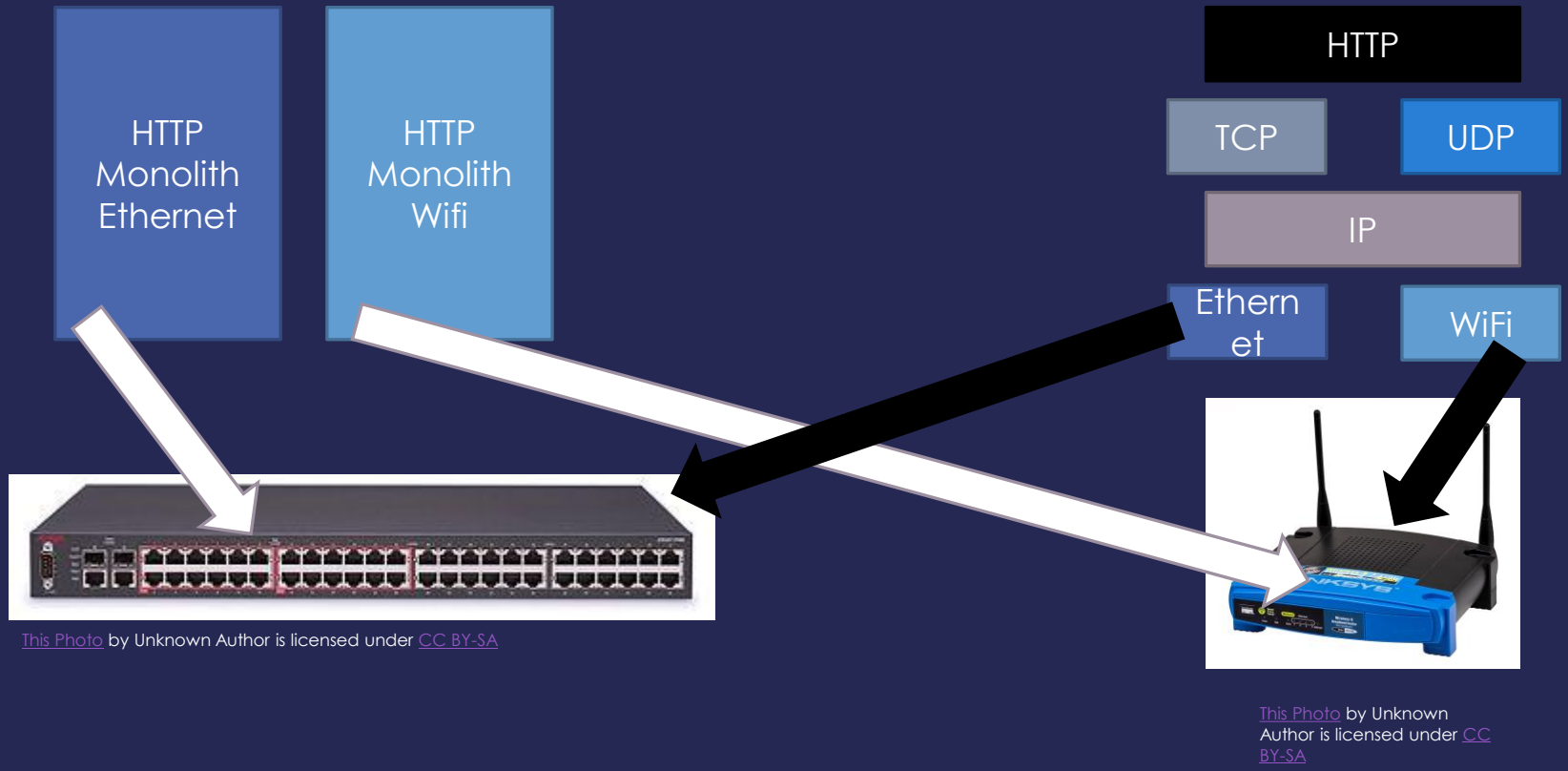
One Protocol is not Enough

- There are too many rules for any one protocol to handle
- Also, behavior/rules need to change for different hardware/goals
- For example, consider HTTP
  - HTTP protocol shouldn't need to worry about the IP protocol rules
  - HTTP definitely shouldn't need to worry about Ethernet rules
  - And HTTP should work even after a switch from Ethernet to Wifi

Protocol *Stacks*

- Object-oriented design!
  - Modularity
  - Abstraction
  - Information hiding
- Protocols are designed in an object-oriented fashion
  - Protocols are combined to solve more complex problems
  - Each protocol should focus on one purpose/goal (High Cohesion)
  - Different component protocols can be swapped (Low coupling)
- We call a group of protocols that work together a *protocol stack*
- In networking, a *network protocol stack* or a *network stack*

Monolithic vs Modular

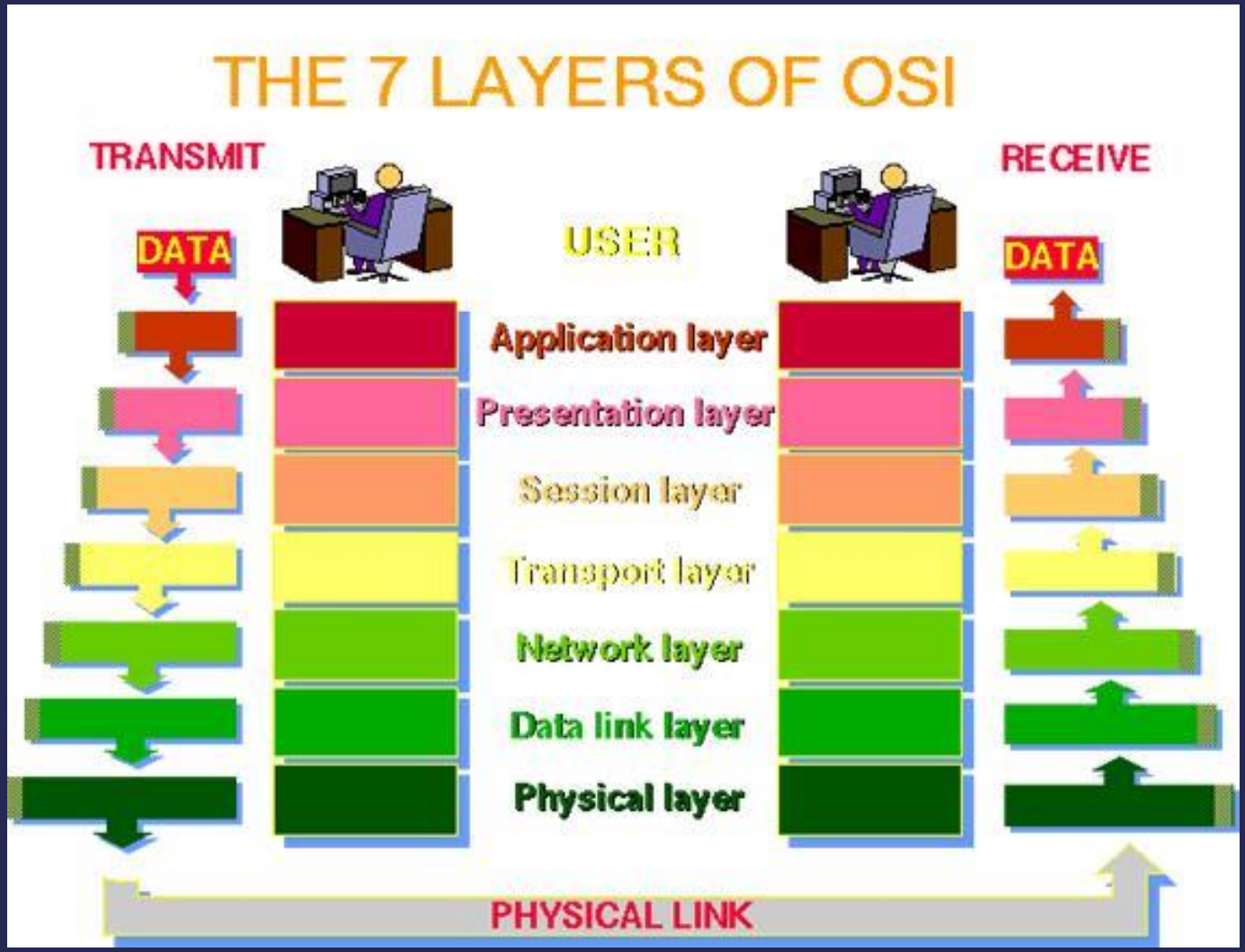


Other Problems with Monolithic

- No separation of user/kernel space components
- Code cannot be reused; code bloat
- NxM combinations
- Patching nightmare
- Testing limitations
- List goes on and on

OSI Model

- Good object-oriented design is implementation independent
- ***Conceptual guide*** for any given network stack ***implementation***
- It has seven layers:
  - 7: Application
  - 6: Presentation
  - 5: Session
  - 4: Transport
  - 3: Network
  - 2: Data Link
  - 1: Physical


THE 7 LAYERS OF OSI
TRANSMIT
RECEIVE
DATA
USER
DATA
Application layer
Presentation layer
Session layer
Transport layer
Network layer
Data link layer
Physical layer
PHYSICAL LINK

The OSI Model in Practice

- Like most OO-designs, the abstraction often breaks down
- The TCP/IP stack really only uses the following layers:
  - Application (Layer 7; example: HTTP)
  - Transport (Layer 4; TCP)
  - IP (Layer 3; IP)
  - Data Link (Layer 2; example: Ethernet)
- Some breakdown in information hiding, abstractions, etc
- NOTE: It's common to just refer to a layer by it's number (e.g., a layer-4 protocol)

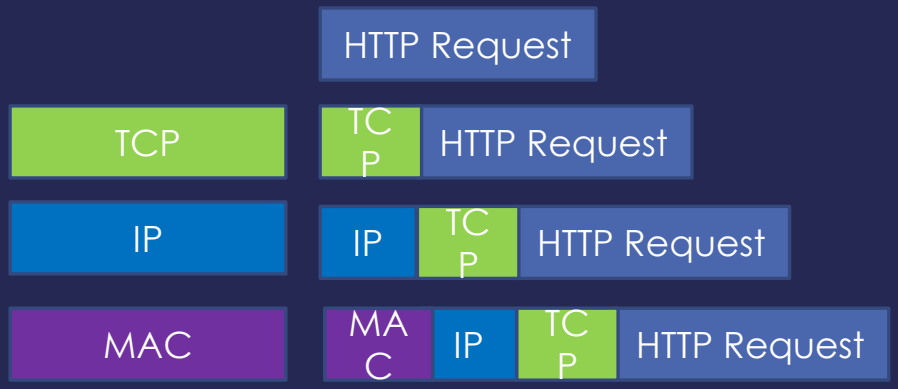
TCP/IP Stack

- For our purposes, we will focus on TCP/IP and TCP/IP-like stacks
- The TCP and IP layers are fixed for layers 3 and 4 (***hourglass***)
- But layers 7 and 2 vary widely
- Millions of networked applications work over TCP/IP at layer 7
- Many layer 2 protocols such as WiFi, Ethernet
  - Networked applications work over WiFi or Ethernet without any change
  - Sometimes called a MAC protocol (Media Access Protocol)
  - TCP/IP work over a walkie-talkie with an appropriate MAC protocol

How does Data Move in a Stack?

- To send, data is inserted (pushed) at the top-most protocol
- The receiving protocol
  - Processes the data, potentially splitting, recoding, etc
  - Derives one or more chunks of output data
  - Metadata added to each chunk (usually a header)
  - Each chunk, along with the meta-data is a "packet"
  - The packet is inserted (pushed) down to the next layer
- On the receiving side, the process is reversed, starting at bottom

TCP/IP Stack Send Example



Division of Labor in TCP/IP

- At the MAC layer, protocol connects 2 endpoints. Typically:
  - Has its own addressing scheme (MAC address)
  - Controls who talks when
  - Provides error detection and *error correction*
- IP (Internetwork Protocol)
  - Connects many different networks of different media types
  - Global addressing scheme
- TCP
  - Reliable, in-order delivery (Session)
  - Multiplexing

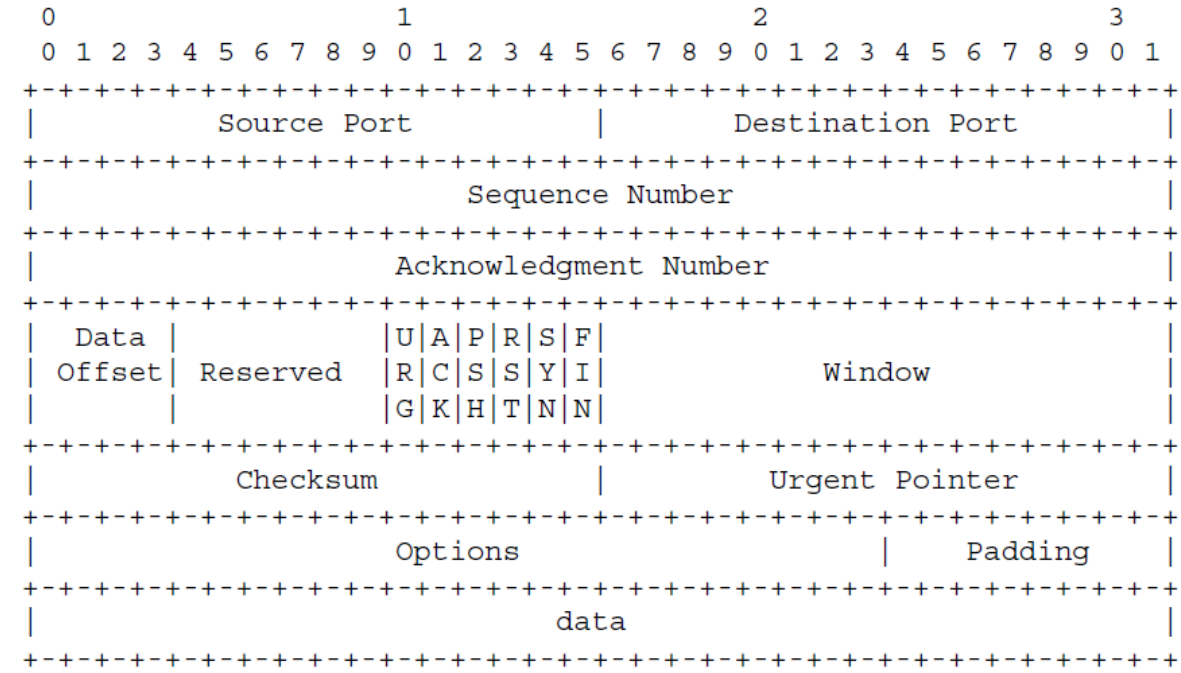
Interoperability

- No one company writes all TCP modules
- Protocol specifications are approved by the IETF
  - You can find the specifications in RFC's (Request For Comments)
  - RFC 793 was the first specification of TCP (1981)
- If an implementation follows the spec, it will be interoperable

RFC 793 (TCP) Overview

- Data broken into "segments" in section 2.2
- Network layers in section 2.5 (a little different from our usage)
- Section 2.6 lays out critical goal: Reliability
  - Data is delivered reliably (i.e., delivery is assured)
  - Data is delivered in-order
  - How? Sequence numbers and acknowledgements on segments
- Section 2.7 identifies another goal: Multiplexing
  - Different flows get different ports
- Section 2.8 indicates that this is a *stream* based protocol

TCP Header Format



TCP Header Format

Note that one tick mark represents one bit position.

Figure 3.