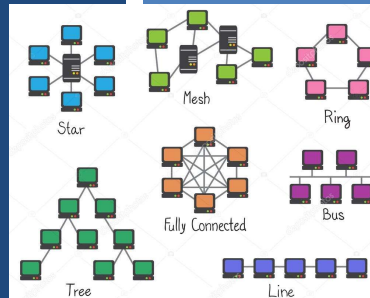


# TCSS 558: APPLIED DISTRIBUTED COMPUTING

## Communication

Wes J. Lloyd

Institute of Technology  
University of Washington - Tacoma



## OBJECTIVES

- Assignment 1 questions
- Midterm Review
- Active Reading Quiz
- Feedback from 10/31
- Ch. 4 - Communications
  - Protocols
  - Remote procedure calls / RMI
  - Message-oriented communication: sockets

November 7, 2017

TCSS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.2

## AWS EDUCATE CREDITS

- Try this website:
- <https://www.awseducate.com/Registration?apptype=student&courseview=true>
- Register for University of Washington, TCSS 558
- Please report success obtaining credits this way

November 7, 2017

TCSS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.3

## FEEDBACK FROM 10/31

- Differences between application agnostic & application specific:
  - Refers to whether a protocol or method (e.g. load balancing) is able to support ***any application*** without modification
  - ***Application specific***: approaches may provide better performance as they use custom methods to address a problem using information about how the application is leveraged
  - Approach often provided by application developer(s)
  - ***Application agnostic***: approaches are generic in nature and do not require any special awareness of the applications involved (e.g. for load balancing)

November 7, 2017

TCSS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.4

## FEEDBACK - 2

- **What other cross-platform languages exist besides Pascal?**
- Python, Perl, Ruby, Java, Lua, C/C++
- Portability effort of code will vary from high (C/C++) to none (Java).
- Portability may depend on the number of specialized features or platform specific libraries are required.

November 7, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.5

## CHAPTER 4

- **4.1 Foundations**
  - Protocols
  - Types of communication
- **4.2 Remote procedure call**
- **4.3 Message-oriented communication**
  - Socket communication
  - Messaging libraries
  - Message-Passing Interface (MPI)
  - Message-queueing systems
  - Examples
- **4.4 Multicast communication**
  - Flooding-based multicasting
  - Gossip-based data dissemination

November 7, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.6



# CH. 4.1: FOUNDATIONS

L11.7

## LAYERED PROTOCOLS

- Distributed systems lack shared memory
- All communication is based on sending and receiving low-level messages
  - $P \rightarrow Q$
- Open Systems Interconnection Reference Model (OSI Model)
  - Open systems communicate with any other open system
  - Standards govern format, contents, meaning of messages
  - Formalization of rules forms a **communication protocol**

November 7, 2017	TCSS558: Applied Distributed Computing [Fall 2017] Institute of Technology, University of Washington - Tacoma	L11.8
------------------	--	-------

## LAYERED PROTOCOLS - 2

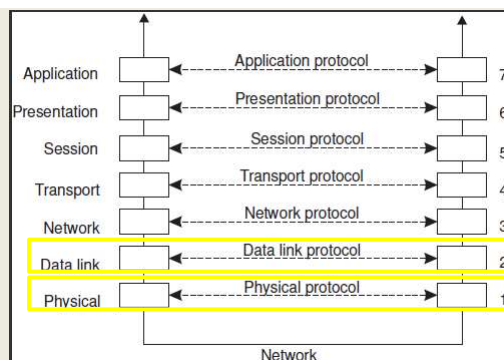
- Protocols provide a **communication service**
  - **Connection-oriented:** sender/receiver establish connection, negotiate parameters of the protocol, close connection when done
    - Physical example: telephone
  - **Connectionless services:** No setup. Sender sends. Receiver receives.
    - Physical example: Mailing a letter

November 7, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.9

## OSI MODEL REVISITED



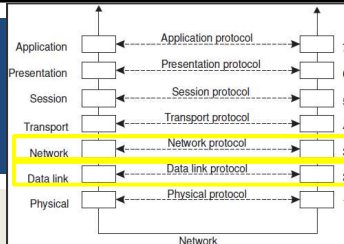
- **Physical layer:** just sends bits
- **Data link layer:** Groups bits into frames
  - Provides error correction via **checksum**
  - Special bit pattern at start/end of frame

November 7, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.10

## OSI MODEL - 2



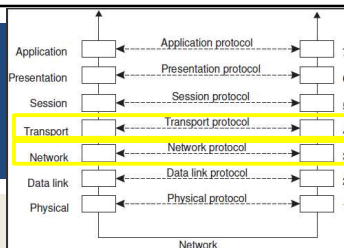
- **Data link layer:**
  - Checksum: computed by adding all bytes in frame in particular way
  - Added to message
  - Receiver removes checksum, recomputes checksum, and compares
  - If receiver and sender agree, frame is considered correct
  - Receiver can request failed frames to be resent
  - Frames assigned sequence numbers *in the header*
  
- **Network layer:**
  - Sometimes referred to as the *Internet layer*
  - On WANs sending msgs between client/server requires routing
  - Provides addressing using IPV4 (32-bit), IPV6 (64-bit)

November 7, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.11

## OSI MODEL - 3



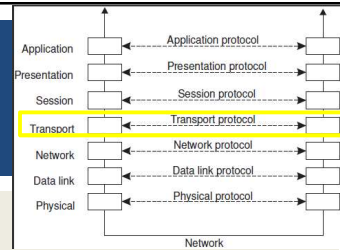
- **Network layer:**
  - Helps with routing network traffic
  - Shortest route (# of hops) may not be the best route
  - Minimizing delay (latency) is paramount
  - Routing algorithms: use long-term average network conditions, or try to adapt to changing conditions
  - ICMP Protocol: Internet Control Message Protocol
  - Not typically for sending data, used for diagnostic/control purposes
  - ICMP Examples: (*ping, traceroute*)
  
- **Transport layer:**
  - Provides reliable connections
  - Reorganizes packets arriving out of sequence
  - Request delivery of missing packets

November 7, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.12

## OSI MODEL - 4



### ■ Transport layer:

1. Breaks application layer protocol messages into pieces to transmit
2. Assigns messages sequence numbers
3. Sends all messages

### ■ Transport layer provides an infallible “message pipe”

- Put messages in
- Always come out undamaged, in correct order

### ■ Transport layer protocols:

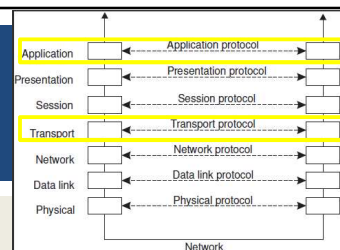
- TCP: Transmission Control Protocol (connection-oriented)
- UDP: Universal Datagram Protocol (connectionless)

November 7, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.13

## OSI MODEL - 5



### ■ Other transport protocols

- Real-time transport protocol (RTP): real-time data, no data delivery guarantee
- Streaming Control Transmission Protocol (SCTP): alternative to TCP

### ■ Higher-level protocols

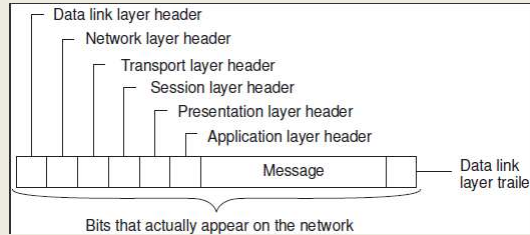
- Session layer: rarely used
- Presentation layer: meaning of the bits;
- Application layer: protocols that don't fit into other layers
  - Many protocols: FTP, SFTP, HTTP, etc. etc.

November 7, 2017

TCCS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.14

# OSI MODEL - 2

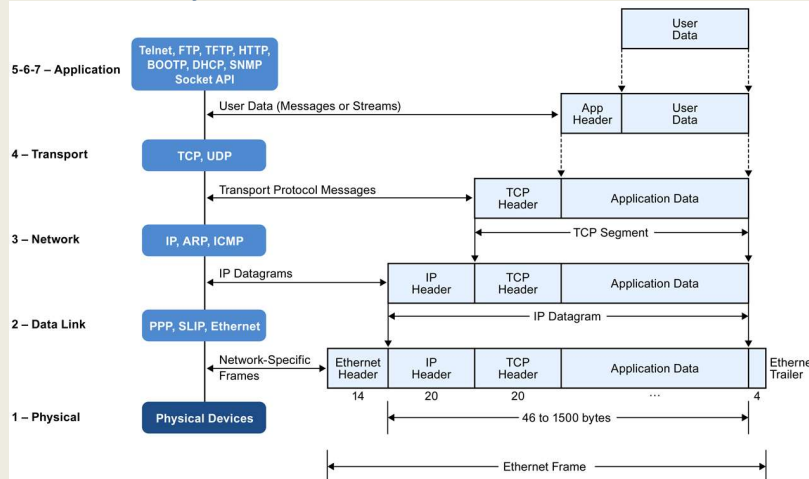


- OSI layers contribute overhead bits to the message
- Layers append data to front (and maybe end) of the message
- Receiving end strips off layers as the message goes up the OSI model stack:  
*physical* → *data-link* → *network* → *transport* → *application*

November 7, 2017	TCCS558: Applied Distributed Computing [Fall 2017] Institute of Technology, University of Washington - Tacoma	L11.15
------------------	--	--------

# PROTOCOL STACK

- Collection of layers used for communication from OSI model



November 7, 2017	TCCS558: Applied Distributed Computing [Fall 2017] Institute of Technology, University of Washington - Tacoma	L11.16
------------------	--	--------



## MIDDLEWARE PROTOCOLS

- Communication frameworks/libraries
- Reused by multiple applications
- Provided needed functions apps build and depend on
- Example:
  - **Authentication protocols:** supports granting users and processes access to authorized resources
  - General, application-independent in nature
  - Doesn't fit as an "application specific" protocol
  - Considered as a "Middleware protocol"

November 7, 2017

TCSS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.17

## QUESTIONS



November 7, 2017

TCSS558: Applied Distributed Computing [Fall 2017]  
Institute of Technology, University of Washington - Tacoma

L11.18

EXTRA SLIDES

19