CS 250: FOUNDATIONS OF COMPUTER SYSTEMS [NETWORKING]

Encapsulation and Layering
Packets grow with headers and trailers
As they trickle down successive layers

A small price to pay
To keep complexity at bay

SHRIDEEP PALLICKARA
Computer Science
Colorado State University

Frequently asked questions from the previous class survey

- The `ServerSocket` establishes the connection (involving two sockets) and then gets out of the way?
- Different ways in which signals are transmitted?
- `DataInputStream`: Can it send multi-byte data?
- Is packet loss what makes calls (VoIP) bad? But downloaded content clear?
- How are "servers" and "clients" different? Can a machine have be either one?

Coding and why it's important

- Primary skill as computer science professionals
  - Gateway to understanding algorithms and software engineering
- Analogy with musicians
  - What kind of music would someone produce if they worked on their craft (and honing their skills) only for 2-3 hours per week?
  - As budding computer scientists, you should be coding at least 20 hours/week on assignments, research (if you are doing this), or personal projects
- You should have coded for 20-30 hours per-week for 6 months, before you say, "I don't like coding". **Things are hard before they get easy!**
- For courses
  - Get started early on assignments
  - Procrastination adversely impacts retention
  - Don't (please just don't!) use GitHub Co-Pilot

Topics covered in this lecture

- Delay x Bandwidth product
- Support for Common Services
- Encapsulation
- Internet Architecture

5

Network performance is measured in two fundamental ways

- **Bandwidth**
  - Number of bits transmitted over the network in a given time (e.g., 10 **m**illion **b**its **p**er seconds 10 Mbps)
  - Also called *throughput*
- **Latency**
  - How long it takes for message to go from one end of the network to another?

COLORADO STATE UNIVERSITY Professor: SHRIDEEP PALLICKARA COMPUTER SCIENCE DEPARTMENT NETWORKING L18.6

6

Components of latency

- Speed-of-light **propagation** delay
  - 3 x 10^8 m/sec in vacuum
  - 2.3 x 10^8 m/sec in cable
- Amount of time to **transmit** a unit of data
- **Queuing** delays

The Delay x Bandwidth product

Viewing the Network as a pipe

The Delay x Bandwidth product

- The product gives us information about *how many bits* **fit** in the pipe
- Transcontinental channel
  - 50 ms one-way latency
  - Bandwidth: 45 Mbps
  - Can hold: 50×10^{-3} seconds x 45×10^{6} bits/second
    - 2.25×10^{6} bits = 280 KB

The Delay x Bandwidth product

- Corresponds to how many bits the sender must send
  - Before first bit arrives at the receiver
- Bits in the pipe are said to be **in flight**

Bandwidth and latency improvements are not in lockstep

- Over past 45 years approximately
  - Bandwidth improvements: 220-1200 times
  - Latency improvements: 4-20 times
- Ethernet 802.3 (1978)
  - 10 Mbps
  - Latency 3 millisecond
- Ethernet 802.3ae (2003)
  - 10,000 Mbps (1000 times)
  - Latency 0.19 millisecond (15 times)
- Ethernet 802.3ck and 802.3db standards [100 Gbps] approved 9/21/23

11

What does not change as the bandwidth increases?

- Speed of light
- High-speed **does not** mean that latency improves at the same rate as bandwidth
  - Transcontinental delays of 100 ms for
    - 1-Mbps/1-Gbps link

12

Sending 1 MB data over a cross country link. Delay 100 ms

- 1 Mbps link
  - Pipe: $100 \times 10^{-3} \times 10^{6}$ = 100 Kb = 0.1 Mb
  - So, you need 80 pipes to transmit 1 MB (8 Mb)
    - 8 Mb/0.1 Mb = 80
- 1 Gbps
  - Pipe: $100 \times 10^{-3} \times 10^{9}$ = 100 Mb
    - So, you need 8 Mb/100 Mb = approx 1/12th of the pipe is utilized

A recent underwater cable [Marea]

- Transatlantic cable (6km under the ocean's surface)
  - Reefs, earthquake zones, etc.
- About the width of a garden hose (1.5x)
- 4000 miles long
  - Virginia Beach, US to Bilboa, Spain
- Transfer rate: 160 Tbps
  - 6000 HiDef Movies
  - The entire catalog of the US Library of Congress
    - ~38 million books and a medium novel's about 1 MB

SUPPORT FOR COMMON SERVICES

COMPUTER SCIENCE DEPARTMENT COLORADO STATE UNIVERSITY

15

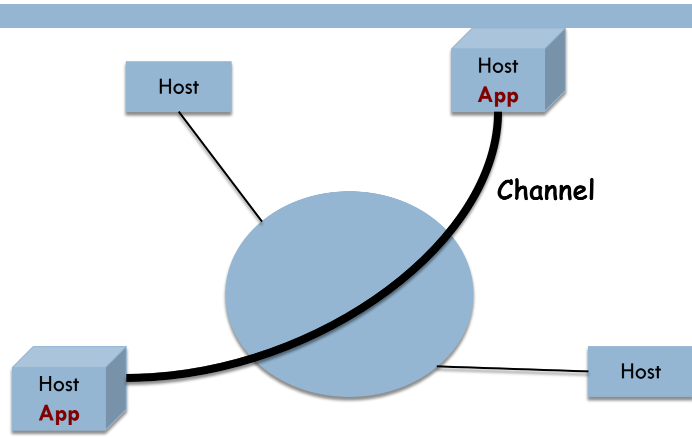
More accurate to think of network as allowing applications to communicate

- When 2 applications need to communicate
  - Lot of things need to happen
  - Beyond just sending messages between the hosts
- Build all functionality into each app?
- Identify and build right set of **common services**
  - Hide complexity without constraining functionality

COLORADO STATE UNIVERSITY Professor: SHRIDEEP PALLICKARA COMPUTER SCIENCE DEPARTMENT NETWORKING L18.16

16

Processes communicating over an abstract channel



17

Guarantees provisioned in the channel

- Guaranteed delivery?
- Ordered delivery?
- Thwart eavesdropping?

18

Not just *which* functionality, but *where* they will be provided

- View network as a **bit pipe**
  - High-level communication semantics provided by end hosts
  - Keeps switches in the middle *very simple*
- Alternative: Push functionality **onto** switches
  - End hosts are dumb devices
    - Telephones

Mask failures so that the network appears more reliable than it really is

- Bit errors
- Burst errors: Consecutive bits are corrupted
- Packet failures
  - Discarded because the switch buffer is full
    - **Congested**
  - Routing mistakes
- Node and link failures
  - Route around failed nodes and links

21

All communications in networked systems are based on sending and receiving messages

- **No shared memory**
- Sending message from **A** to **B**
  - *Build message* in **A**'s address space
  - *Send message* over the network
  - *Reconstruct* message at **B**

COLORADO STATE UNIVERSITY Professor: SHRIDEEP PALLICKARA COMPUTER SCIENCE DEPARTMENT NETWORKING L18.22

22

But A and B must agree on the *meaning* of the bits

- Signaling **1**'s and **0**'s
- What is the *last bit* of the message?
- Detect if the message is lost or damaged
  - Respond to problems
- Representation of data types

Layering and Protocols

- Start with services provided by hardware
- Add a **sequence** of layers
  - Each providing higher level of service
- Services at higher layers implemented **in terms of** lower layers

Advantages of layering

- Decomposes problem into **manageable** components
- Provides **modular** design
  - Adding functionality may result only in minor modifications

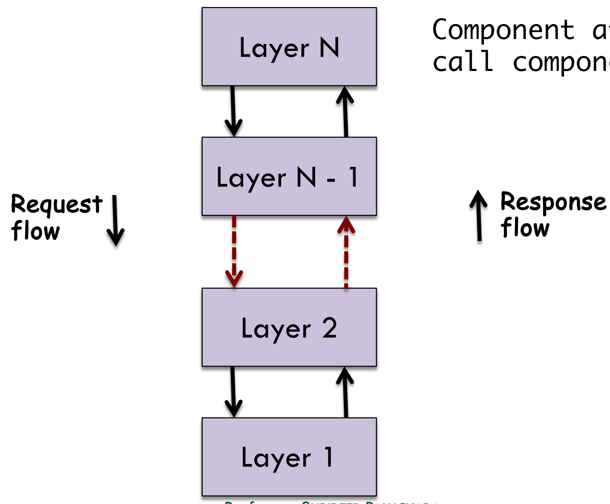
25

Advantages of layering

- Each layer can be changed **independently** of the other
  - Change as technology improves

26

Layered Architectures: Requests go down the hierarchy; results flow upward



Component at layer L_i can call components at layer L_{i-1}

Example of a layered network system

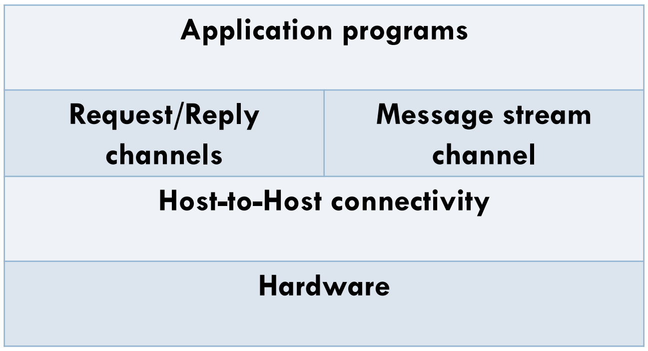
Application programs

Process-to-process channels

Host-to-Host connectivity

Hardware

Layered system with alternative abstractions at a given layer



Abstract objects that comprise layers of a network system are called **protocols**

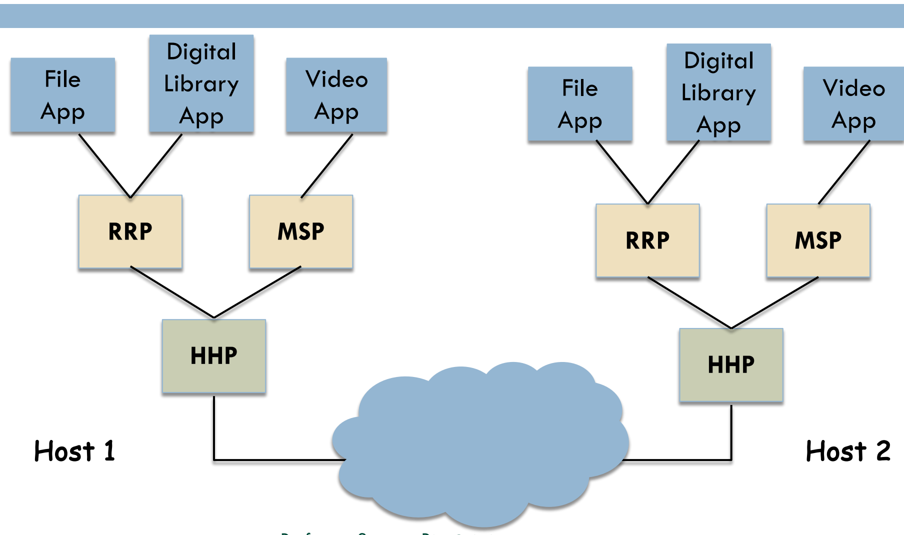
- Provides a **service interface** to other objects on the same computer
  - Wishing to use its communication services
- Defines the **form** and **meaning of messages** exchanged by protocol peers
- Protocol also refers to modules that implement a specification

ENCAPSULATION

COMPUTER SCIENCE DEPARTMENT
COLORADO STATE UNIVERSITY

31

Example of a protocol graph



COLORADO STATE UNIVERSITY
Professor: SHRIDEEP PALLICKARA
COMPUTER SCIENCE DEPARTMENT
NETWORKING
L18.32

32

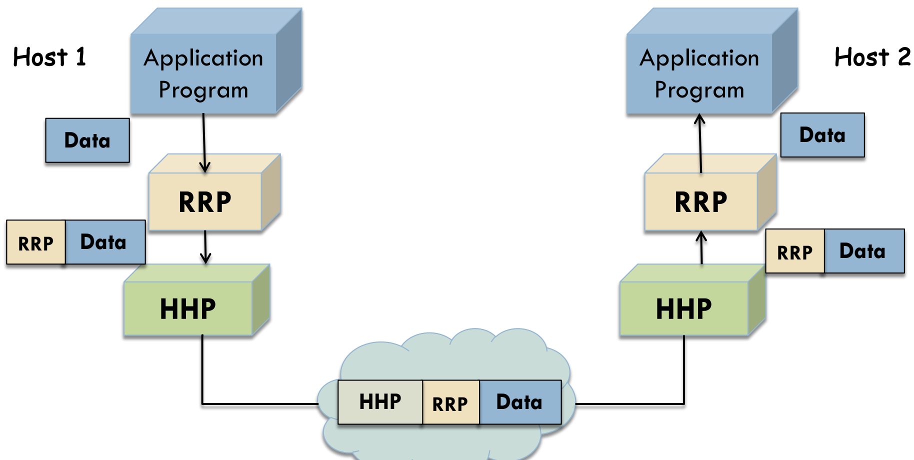
Encapsulation

- RRP receives a set of bytes to transmit from the application
  - E-mail, integers, images etc
- RRP is responsible for sending this data to its peer at the other end
  - Must communicate **control information** to its peer
  - Instruct how to **handle** the message

When asked to transmit info, lower level layers add information to the message

- Attach a **header** to the message
  - Small data structure
  - Few bytes to several dozen bytes
- Control info at the end of message: **trailer**
- Format is specific to the protocol
- Data being transmitted: body or **payload**
- Application data is said to be **encapsulated**

Encapsulating high-level messages inside low-level messages



Encapsulation: Some more info

- Low-level protocol does not **interpret** message given to it by high-level protocol
  - *Cannot extract* meaning
- Low-level protocol may apply simple **transformations** to the data it is given
  - Compress
  - Encrypt

Multiplexing is applicable up-and-down the protocol graph too

- RRP attaches header to every message that goes through it
  - Header includes information to identify the application
    - Called demultiplexing key or **demux key**
- At the destination host, RRP strips its header
  - Examines demux key
  - Demultiplexes message to correct application

Demux key is used at all levels of the protocol stack

- Some use an 8-bit field {**TCP** (6), **UDP** (17)}
  - Can support only 2^8 (256) high level protocols
  - Can also be 16/32-bits
- There could be a **single** demultiplexing field
  - Same demux key used at both ends
- There could be a **pair** of demultiplexing fields
  - Each side uses different key to identify high-level protocol

A HIGH-LEVEL VIEW OF THE INTERNET

39

A computer network is a system that allows computing devices to **communicate** with each other

- Networks can be connected wirelessly, using technologies like Wi-Fi
  - Transmit data using radio waves
- Networks can also be connected with cables
  - Such as copper wiring or fiber optics

COLORADO STATE UNIVERSITY — Professor: SHRIDEEP PALLICKARA, COMPUTER SCIENCE DEPARTMENT — NETWORKING — L18.40

40

A 30,000 foot view of the internet

- The internet is a globally connected set of computer networks that all use a **suite of common protocols**
- The internet is a **network of networks**
  - Connecting networks from various organizations all around the world

Does physical connectivity equate with communications? [1/2]

- Physically connecting the networks of the world *isn't enough* to allow the devices on those networks to communicate with each other
- All participating computers need to **communicate in the same way**

Does physical connectivity equate with communications? [2/2]

- The internet protocol suite **standardizes** the method of communication on the internet
  - Ensuring that all devices on the network speak the same language
- The two foundational protocols in the internet protocol suite are
  - Transmission Control Protocol (TCP) and Internet Protocol (IP)
  - Collectively known as **TCP/IP**

Network protocols operate in a layered model [1/2]

- An implementation of such a model is referred to as a **network stack**
- The protocols at the lowest layer
  - Interact with the underlying networking hardware
- Applications interact with protocols in the upper layers

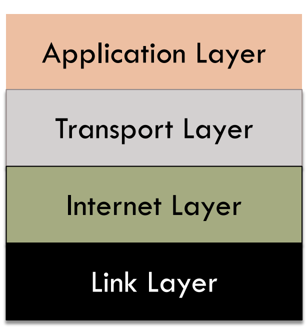
Network protocols operate in a layered model [2/2]

- Protocols in the intermediate layers provide services such as addressing and reliable delivery of data
- A protocol at a certain layer doesn't have to concern itself with the entire networking stack
  - **Only the layers with which it interfaces**, simplifying the overall design
  - This is another example of encapsulation

The internet protocol suite is designed around a four-layer model

- This is sometimes called the **TCP/IP model**
- The four layers of the protocol suite:
  - The link layer
  - The internet layer
  - The transport layer
  - The application layer

The Internet Layered Protocol Stack



Networking layers

- Networking layers represent an **abstraction**
  - A model for us to use when discussing the operation of the internet
- In practice, each layer is realized with specific networking protocols
- Each network layer **represents a scope of responsibilities**
  - Protocols must fulfill the responsibilities of their assigned layer

Application Layer

- Protocols that operate at the application layer provide **application-specific functionality**
  - Sending an email, retrieving a web page, transferring a file, etc.
  - E.g.: SMTP, HTTP, SSH
- These protocols accomplish tasks that end users (or backend services) wish to complete
- Application layer protocols structure the data used in process-to-process communication across a network
- All lower layer protocols exist as “plumbing” to support the application layer

Transport Layer

- Transport layer protocols provide a **communications channel** for applications to send and receive data between hosts
  - E.g., TCP and UDP
- An application
  - **Structures data** according to an application layer protocol and
  - **Hands off that data** to a transport layer protocol for delivery to a remote host

Internet Layer

- Internet layer protocols provide a mechanism for **communicating across networks**
- Responsible for identifying hosts with addresses and enabling the routing of data from network to network across the internet
- The transport layer relies on the internet layer for **addressing** and **routing**

Link Layer

- Link layer protocols provide a way to **communicate on a local network**
- Protocols at this layer are closely associated with the type of networking hardware on a local network
  - E.g., Wi-Fi, Ethernet, FDDI, etc.
- Protocols at the internet layer rely on link layer protocols to communicate on a local network

Not every device is interested in using all 4 layers

- Network hosts (such as a client or server) make use of protocols from all four layers
- Other types of networking hardware (such as switches and routers) only use protocols associated with lower layers
  - Such devices can perform their jobs without bothering to examine the higher layer protocol data contained in a network transmission

The contents of this slide-set are based on the following references

- *Computer Networks: A Systems Approach. Larry Peterson and Bruce Davie. 4th edition. Morgan Kaufmann. ISBN: 978-0-12-370548-8. [Chapter 1, 2]*
- Matthew Justice. *How Computers Really Work: A Hands-On Guide to the Inner Workings of the Machine*. ISBN-10/ISBN-13 : 1718500661/ 978-1718500662. No Starch Press. 2020. [Chapter 11]