

Distributed Systems

2. Networking Intro

Paul Krzyzanowski

pxk@cs.rutgers.edu

How do nodes share a network?

- Dedicated connection – no sharing: *physical circuit*
- Talk on different frequencies: *broadband*
- Take turns (*baseband*)
 - Short fixed time slots: *TDMA (Time Division Multiple Access)*
 - *Circuit switching*: performance equivalent to an isolated connection
 - Variable size time slots: *Packets*
 - *Statistical multiplexing* for network access
 - Permits many-to-many communication
- **Packet switching** is the dominant means of data communication

Modes of connection

Circuit-switched

- Dedicated path (route)
- Guaranteed (fixed) bandwidth
- Constant latency

Packet-switched

- Shared connection; competition for use with others
- Data is broken into chunks called packets
- Each packet contains a destination address
- available bandwidth \leq channel capacity
- variable latency

What's in the data?

For effective communication

- same language, same conventions

For computers:

- electrical encoding of data
- where is the start of the packet?
- which bits contain the length?
- is there a checksum? where is it?
how is it computed?
- what is the format of an address?
- byte ordering

Protocols

These instructions and conventions are known as **protocols**

Protocols

Exist at different levels

*understand format of address
and how to compute a checksum*

*humans vs. whales
different wavelengths*

versus

request web page

French vs. Hungarian

Layering

To ease software development and maximize flexibility:

- Network protocols are generally organized in **layers**
- Replace one layer without replacing surrounding layers
- Higher-level software does not have to know how to format an Ethernet packet

... or even know that Ethernet is being used

Layering

Most popular model of guiding
(not specifying) protocol layers is

OSI reference model

Adopted and created by ISO

7 layers of protocols

OSI Reference Model: Layer 1

Transmits and receives raw data to communication medium

Does not care about contents

voltage levels, speed, connectors

Physical

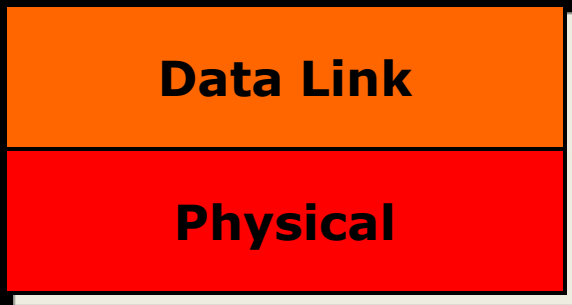
Examples: RS-232, 10BaseT

OSI Reference Model: Layer 2

Detects and corrects errors

Organizes data into packets before passing it down. Sequences packets (if necessary)

Accepts acknowledgements from receiver

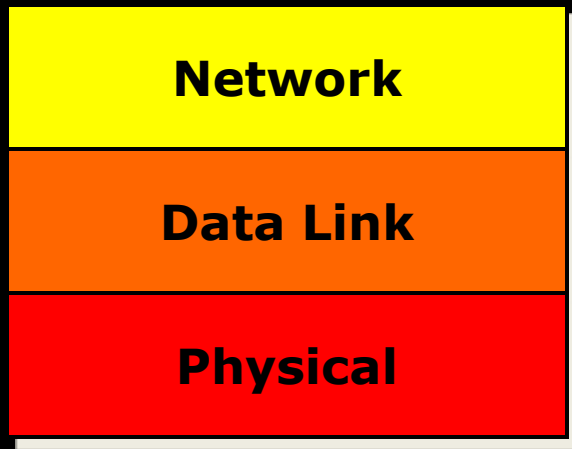


Examples: Ethernet MAC, PPP

OSI Reference Model: Layer 3

Relay and route information to destination

Manage journey of packets and figure out intermediate hops (if needed)

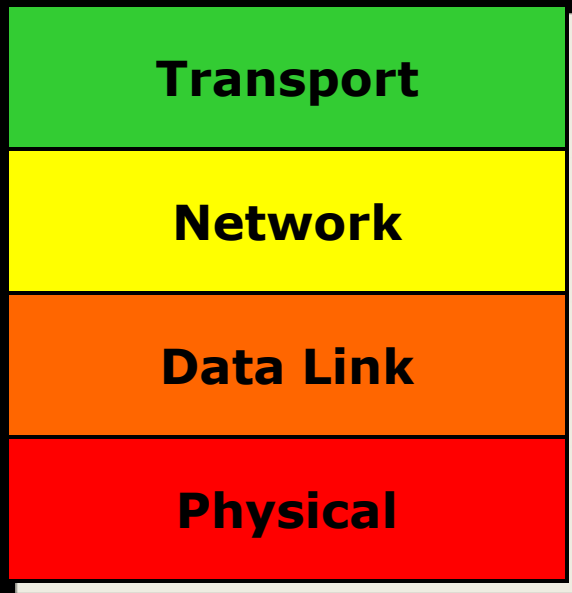


Examples: IP, X.25

OSI Reference Model: Layer 4

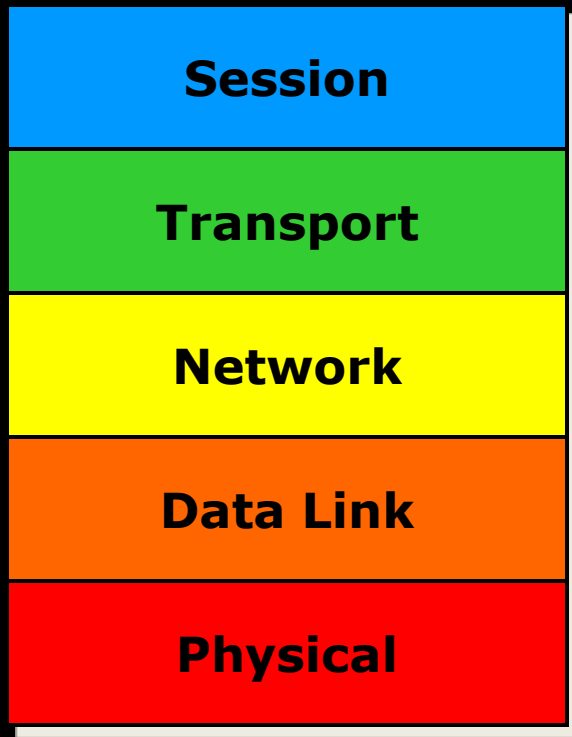
Provides a consistent interface for end-to-end (application-to-application) communication.
Manages flow control

Network interface is similar to a mailbox



Examples: TCP, UDP

OSI Reference Model: Layer 5



Services to coordinate dialogue and manage data exchange

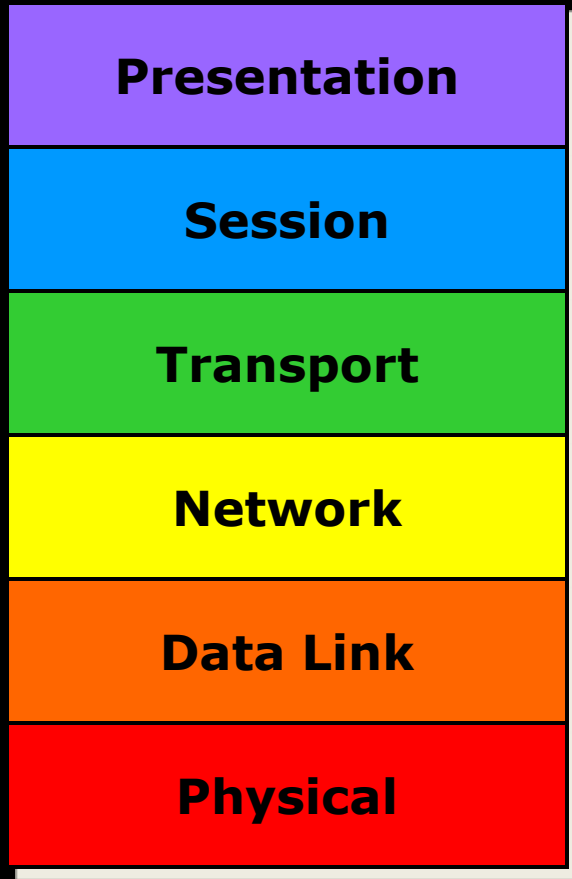
Software implemented switch

Manage multiple logical connections

Keep track of who is talking: establish & end communications

Examples: HTTP 1.1, SSL, NetBIOS

OSI Reference Model: Layer 6



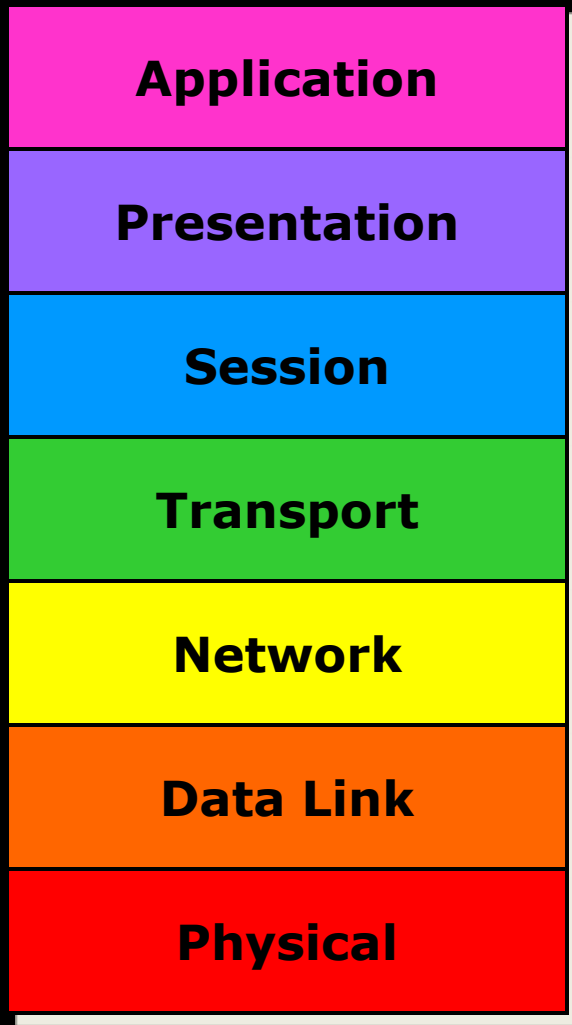
Data representation

Concerned with the meaning of data bits

Convert between machine representations

Examples: XDR, ASN.1, MIME

OSI Reference Model: Layer 7



Collection of application-specific protocols

Examples:

email (SMTP, POP, IMAP)
file transfer (FTP)
directory services (LDAP)

Baseband: Ethernet

Standardized by IEEE as 802.3 standard

Speeds: 100 Mbps - 1 Gbps typical today

- Ethernet: 10 Mbps
- Fast Ethernet: 100 Mbps
- Gigabit Ethernet: 1 Gbps
- 10 Gbps, 100 Gbps

Network access method is

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

- Node first listens to network to see if busy
- Send
- Sense if collision occurred
- Retransmit if collision

Client – Server Communication

Clients and Servers

- Send messages to *applications*
 - not just machines
- Client must get data to the desired *process*
 - server process must get data back to client process
- To offer a service, a server must get a **transport address** for a particular service
 - well-defined location

Machine address versus Transport address

Transport provider

Layer of software that accepts a network message and sends it to a remote machine

Two categories:

connection-oriented protocols

connectionless protocols

Connection-oriented Protocols

1. establish connection
2. [negotiate protocol]
3. exchange data
4. terminate connection

Connection-oriented Protocols

	<u>analogous to phone call</u>
1. establish connection	<i>dial phone number</i>
2. [negotiate protocol]	<i>[decide on a language]</i>
3. exchange data	<i>speak</i>
4. terminate connection	<i>hang up</i>

virtual circuit service

- provides illusion of having a dedicated circuit
- messages guaranteed to arrive in-order
- application does not have to address each message

vs. circuit-switched service

Connectionless Protocols

- no call setup
- send/receive data
 (each packet addressed)
- no termination

Connectionless Protocols

analogous to mailbox

- no call setup
- send/receive data
(each packet addressed)
- no termination

drop letter in mailbox
(each letter addressed)

datagram service

- client is not positive whether message arrived at destination
- no state has to be maintained at client or server
- cheaper but less reliable than virtual circuit service

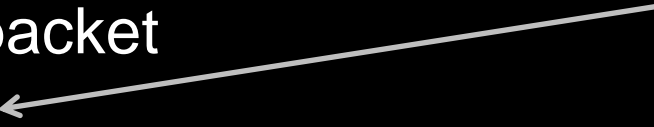
Ethernet

- Layers 1 & 2 of OSI model
 - Physical (1)
 - Cables: 10Base-T, 100Base-T, 1000Base-T, etc.
 - Data Link (2)
 - Ethernet bridging (via bridges)
 - Data frame parsing
 - Data frame transmission
 - Error detection
- **Unreliable, connectionless communication**

Ethernet

- 48-bit ethernet address
- Variable-length packet
 - 1518-byte **MTU**
 - 18-byte header, 1500 bytes data
- Jumbo packets for Gigabit ethernet
 - 9000-byte MTU

Maximum transmission unit



IP – Internet Protocol

Born in 1969 as a research network of 4 machines

Funded by DoD's ARPA

Goal:

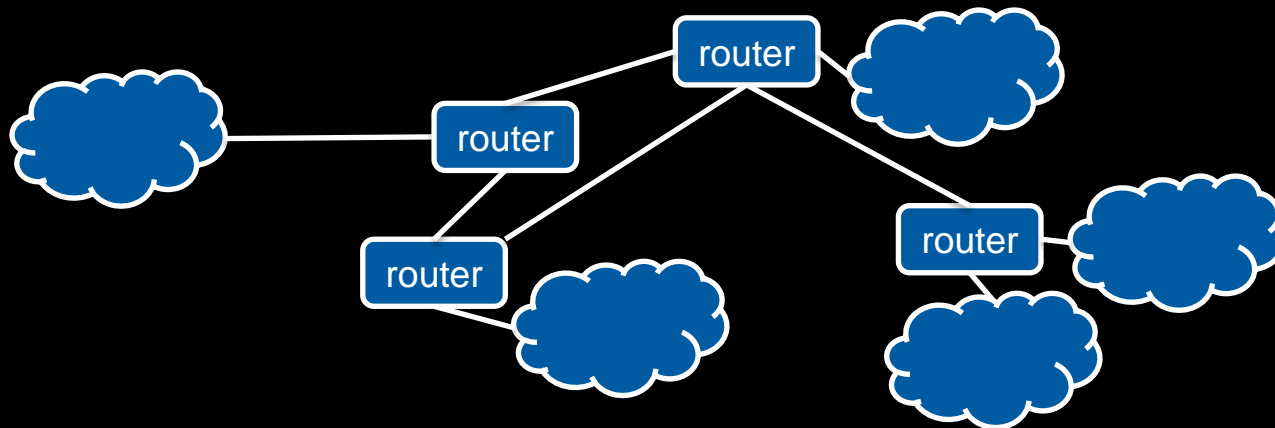
Build an efficient fault-tolerant network that could connect heterogeneous machines and link separately connected networks.

Internet Protocol

Connectionless protocol designed to handle the interconnection of a large number of local and wide-area networks that comprise the internet

IP can **route** from one physical network to another

Survivable design: support multiple paths for data



IP Addressing

Each machine on an IP network is assigned a unique 32-bit number for each network interface:

- **IP address**, *not* machine address

A machine connected to several physical networks will have several IP addresses

- One for each network

IP Address space

32-bit addresses → >4 billion addresses!

- Routers would need a table of 4 billion entries
- Design routing tables so one entry can match multiple addresses
 - **hierarchy**: addresses physically close will share a common prefix

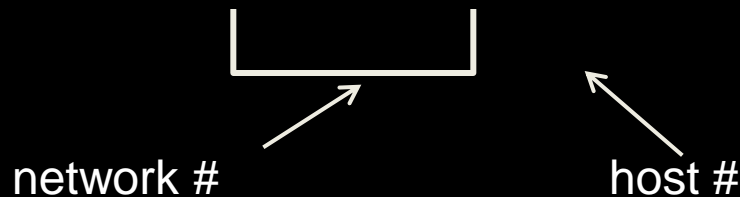
IP Addressing: networks & hosts

cs.rutgers.edu

128.6.4.2

remus.rutgers.edu

128.6.13.3



- first 16 bits identify Rutgers
- external routers need only one entry
 - route 128.6.*.* to Rutgers

IP Addressing: networks & hosts

- IP address
 - **network #**: identifies network machine belongs to
 - **host #**: identifies host on the network
- use network number to route packet to correct network
- use host number to identify specific machine

IP Addressing

Expectation:

- a few big networks and many small ones
- create different **classes** of networks
- use leading bits to identify network

class	leading bits	bits for net #	bits for host
A	0	7 (128)	24 (16M)
B	10	14 (16K)	16 (64K)
C	110	21 (2M)	8 (256)

IP Addressing: networks & subnets

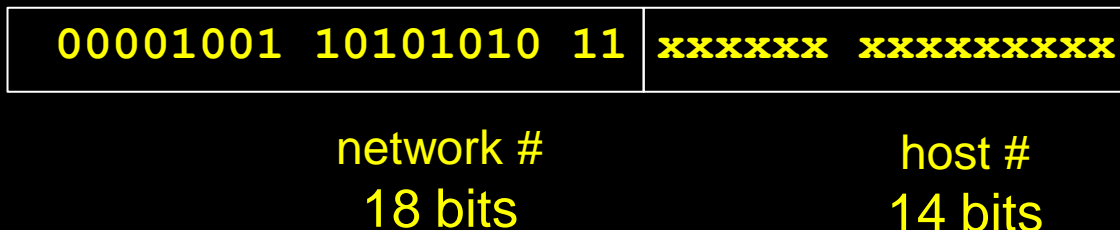
IBM: 9.0.0.0 – 9.255.255.255



To allow additional networks within an organization:

use high bits of host number for a “network within a network” – **subnet**

Subnet within IBM (internal routers only)



Running out of addresses

- Huge growth
- Wasteful allocation of networks
 - Lots of unused addresses: *Does IBM need 16.7M IP addresses?*
- Every machine connected to the internet needed a worldwide-unique IP address
- Solutions: **CIDR**, **NAT**, **IPv6**

IPv6 vs. IPv4

IPv4

- 4 byte (32 bit) addresses

IPv6:

- 16-byte (128 bit) addresses
3.6 x 10³⁸ possible addresses
8 x 10²⁸ times more addresses than IPv4
- 4-bit priority field
- Flow label (24-bits)

IP Transport Layer Protocols

Transport-layer protocols over IP

- IP sends packets to machine
 - No mechanism for identifying sending or receiving application
- Transport layer uses a **port number** to identify the application
- TCP – Transmission Control Protocol
- UDP – User Datagram Protocol

TCP – Transmission Control Protocol

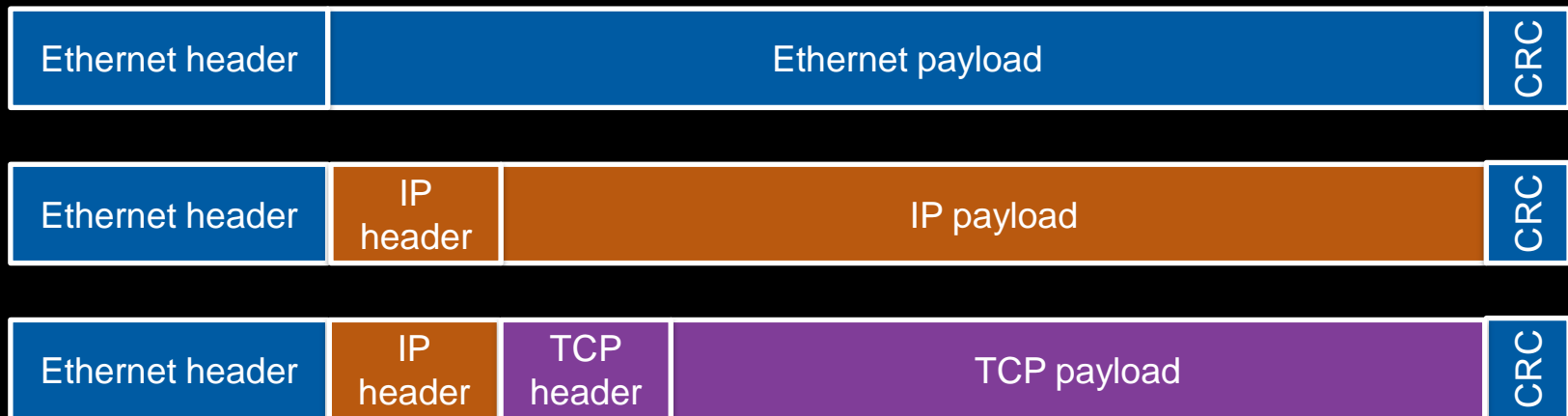
- Virtual circuit service
(connection-oriented)
- Send acknowledgement for each received packet
- Checksum to validate data
- Data may be transmitted simultaneously in both directions

UDP – User Datagram Protocol

- Datagram service (connectionless)
- Data may be lost
- Data may arrive out of sequence
- Checksum for data but no retransmission
 - Bad packets dropped

Protocol Encapsulation

- Layering protocols
- A higher level protocol is simply treated like data (payload)



The End