Operating Systems Design
15. Client-Server Networking

Paul Krzyzanowski
pxk@cs.rutgers.edu

Some networking terminology

Local Area Network (LAN)

Communications network

- small area (building, set of buildings)
- same, sometimes shared, transmission medium
- high data rate (often): 1 Mbps – 1 Gbps
- Low latency
- devices are peers
  - any device can initiate a data transfer with any other device

Most elements on a LAN are workstations

- endpoints on a LAN are called **nodes**

Connecting nodes to LANs

Connecting nodes to LANs

Adapter

- expansion slot (PCI, PC Card, USB dongle)
- usually integrated onto main board

Network adapters are referred to as
Network Interface Cards (NICs) or adapters

or Network Interface Component
(since they're often not cards anymore)

Hubs, routers, bridges

Hub

- Device that acts as a central point for LAN cables
- Take incoming data from one port & send to all other ports

Switch

- Moves data from input to output port.
- Analyzes packet to determine destination port and makes a virtual connection between the ports.

Concentrator or **repeater**

- Regenerates data passing through it

Bridge

- Connects two LANs or two segments of a LAN: extends a LAN
- Connection at <u>data link layer</u> (layer 2)

Router

- Determines the next network point to which a packet should be forwarded
- Connects different types of local and wide area networks at <u>network layer</u> (layer 3)

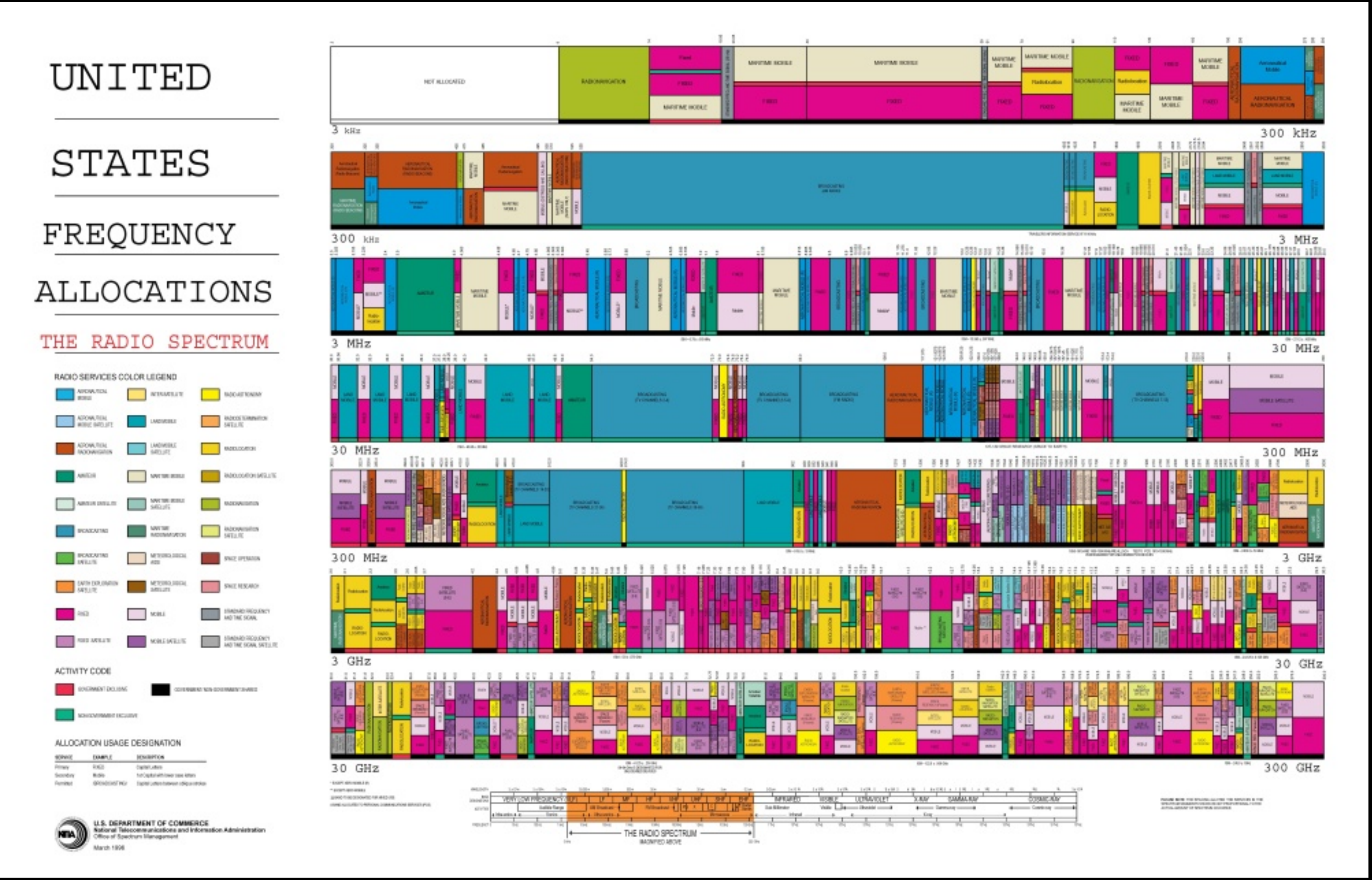
How do nodes share a network?

- Dedicated connection – no sharing: *physical circuit*
- Talk on different frequencies: *broadband*
  - Data uses segment of medium (channels, frequency bands)
- Take turns (*baseband)*
  - A node can use the entire bandwidth of medium

  1. Short fixed time slots: *TDMA (Time Division Multiple Access)*
     - *Circuit switching: performance equivalent to an isolated connection*
  2. Variable size time slots: *Packets*
     - *Statistical multiplexing* for network access
     - Permits many-to-many communication

- Packet switching is the dominant means of data communication

Broadband: RF broadcasts

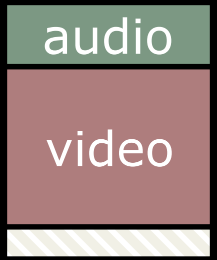


http://www.ntia.doc.gov/osmhome/allochrt.pdf

Broadband/Baseband: Cable TV

Broadband

55-552 MHz: analog channels 2-78
553-865 MHz: digital channels 79-136



Baseband within Broadband

DOCSIS: Data Over Cable Service Interface Specification
(approved by ITU in 1998; DOCSIS 2.0 in 2001; DOCSIS 3.0 in 2006)

Downstream: 50-750 MHz range, 6 MHz bandwidth

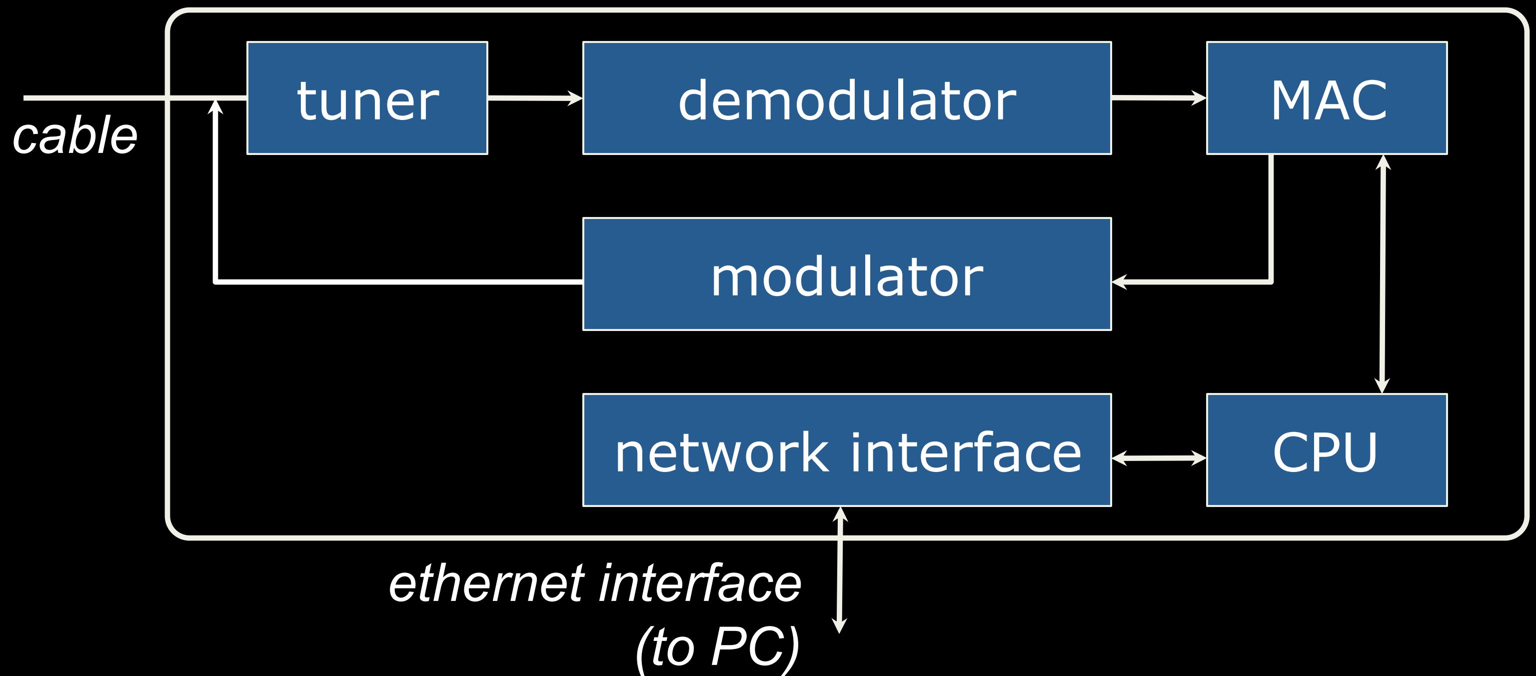
- up to 38 Mbps
- received by all modems

Upstream: 5-42 MHz range

- 30.72 Mbps (10 Mbps in DOCSIS 1.0, 1.1)
- data delivered in timeslots (TDM)

DOCSIS 3.0 features channel bonding for greater bandwidth

DOCSIS Modem



Restrictions on upload/download rates set by transferring a configuration file to the modem via TFTP when it connects to the provider.

Baseband: Ethernet

- Packet-based network
  - Data sent as a series of frames (packets)
- Speeds: 1 Gbps most common interface today
  - Ethernet: 10 Mbps • Fast Ethernet: 100 Mbps • Gigabit Ethernet: 1 Gbps
  - Also 10 Gbps and 100 Gbps
- Ethernet network access method on a shared channel 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
- Ethernet switches don’t use shared channels – no need for CSMA/CD

802.11 Family (Wi-Fi)

- Network access via
  Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)
  - Cannot use CSMA/CD on wireless networks – can't listen while sending
  - Node first listens on the desired channel to see if idle
  - Send a packet if idle
  - If busy, wait until transmission stops + random contention period
  - Transmit if still idle
  - Receive ACK from receiver

Client-Server Networking

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 ≤ 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

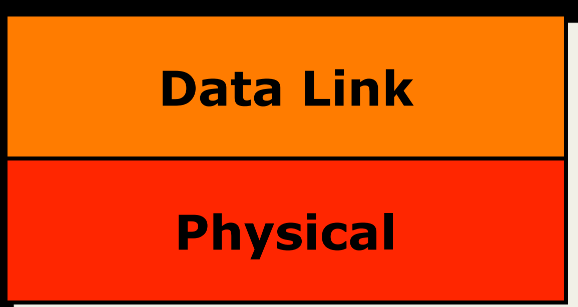
Examples: USB, RS-232, 1000BaseT

OSI Reference Model: Layer 2

Detects and corrects errors

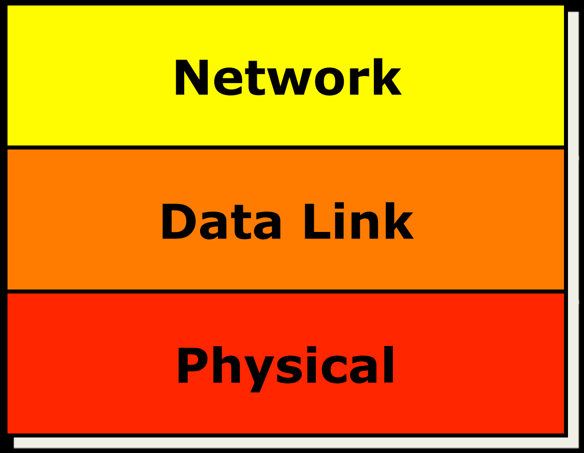
Organizes data into packets before passing it down. Sequences packets (if necessary)

Accepts acknowledgements from receiver



Examples: Ethernet MAC, PPP

OSI Reference Model: Layer 3

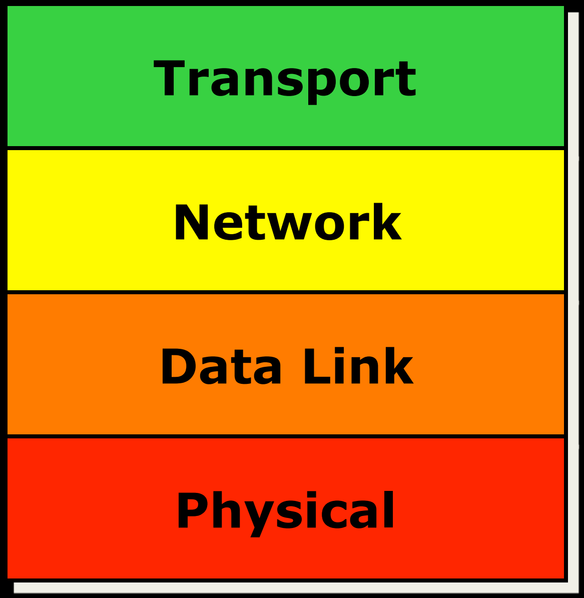


Relay and route information to destination

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

Examples: IP, X.25

OSI Reference Model: Layer 4

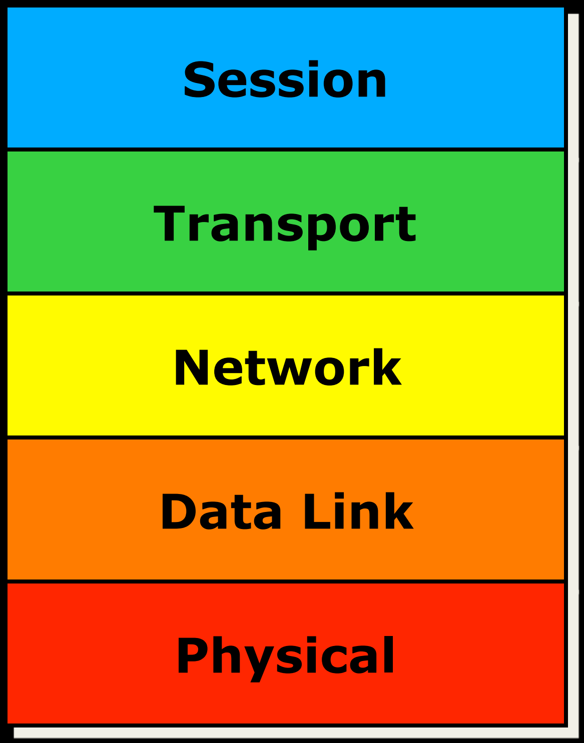


Provides a consistent interface for end-to-end (application-to-application) communication. Manages flow control

Network interface is similar to a mailbox

Examples: TCP, UDP

OSI Reference Model: Layer 5



Services to coordinate dialogue and manage data exchange

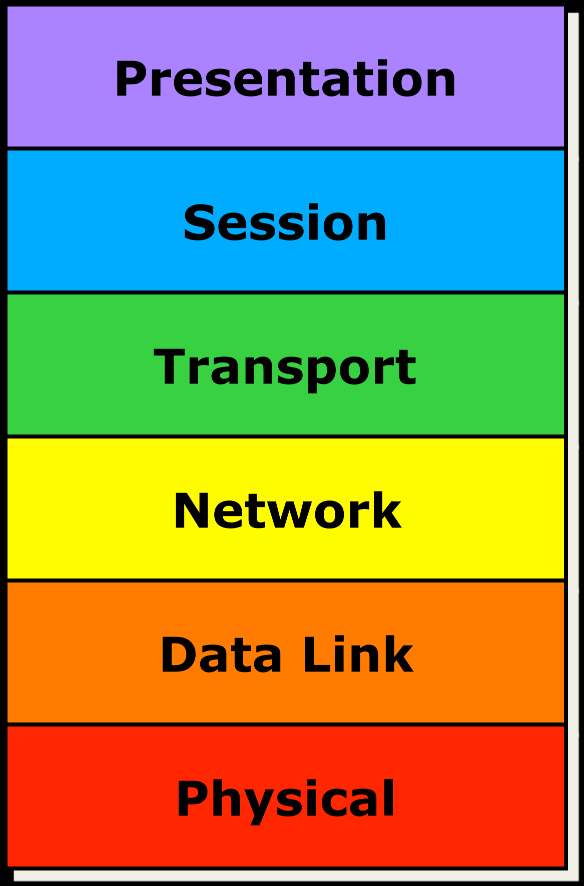
Software implemented switch

Manage multiple logical connections

Keep track of who is talking: establish & end communications

Examples: HTTP 1.1, SSL, NetBIOS

OSI Reference Model: Layer 6



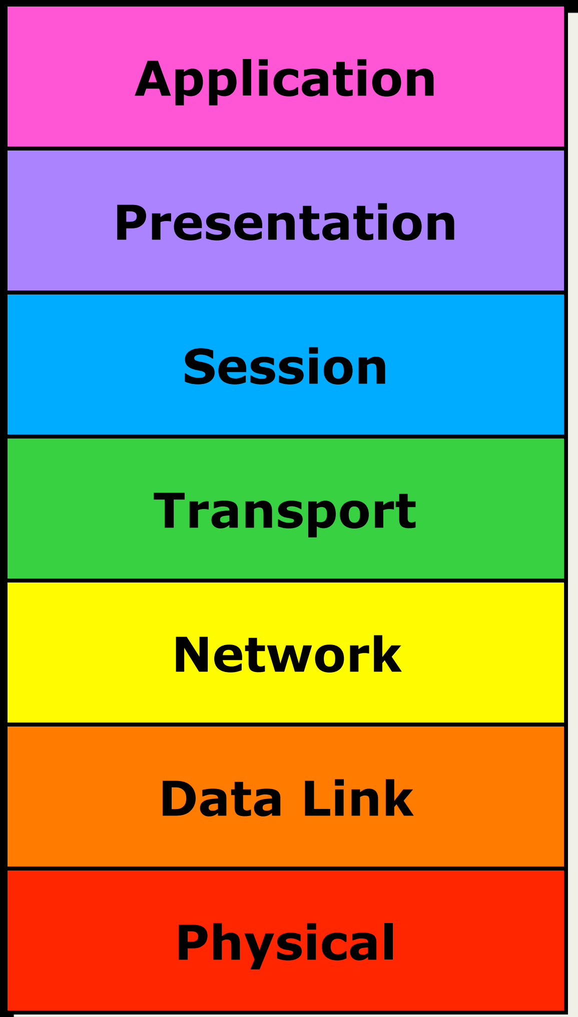
Data representation

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)

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

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

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

<u>analogous to mailbox</u>

- 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

Maximum transmission unit

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

dest addr	src addr	frame type	data (payload)	CRC

IP – Internet Protocol

Born in 1969 as a research network of 4 machines

Funded by DoD's ARPA

<ins>Goal</ins>:

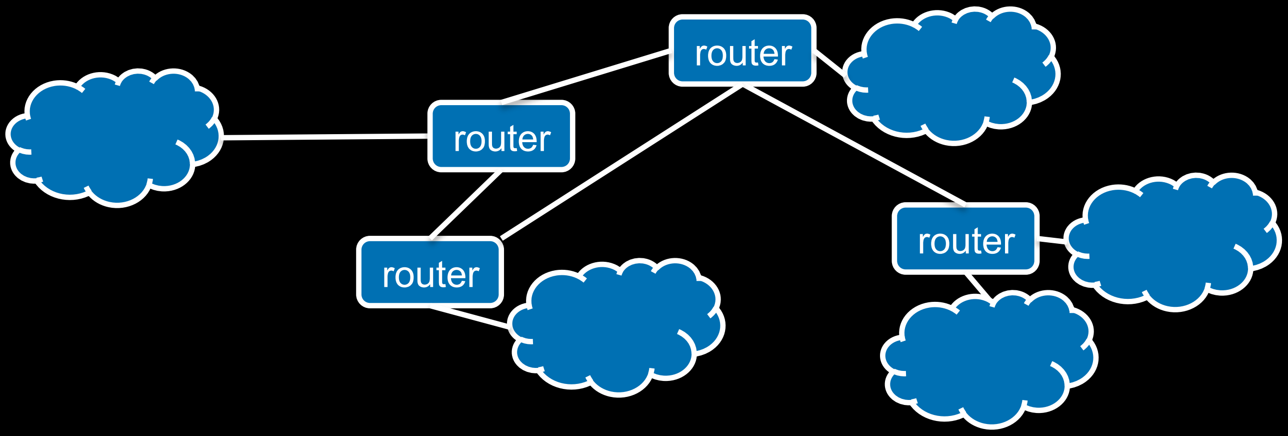
Build an efficient fault-tolerant network that could connect heterogeneous machines and link separately connected networks.

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

sakai.rutgers.edu

128.6.31.88

network #

host #

- 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

00001001	xxxxxxxx xxxxxxxx xxxxxxxxx
network # 8 bits	host # 24 bits

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)

00001001 10101010 11	xxxxxx xxxxxxxxx
network # 18 bits	host # 14 bits

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:

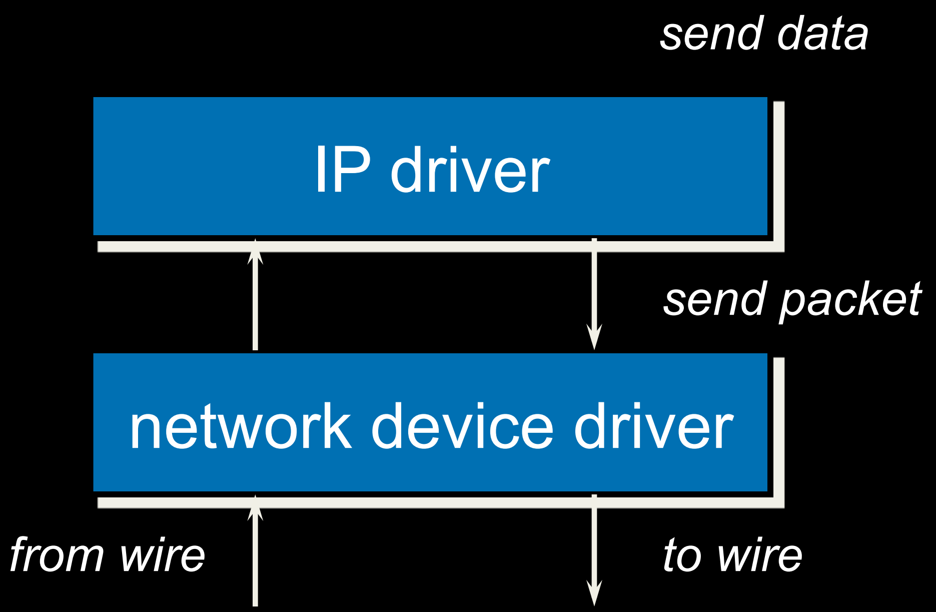
- 16-byte (128 bit) addresses
  3.6×10^{38} possible addresses
  8×10^{28} times more addresses than IPv4
- 4-bit priority field
- Flow label (24-bits)

Networking: Back to the OS

Getting to the machine

IP is a logical network on top of multiple physical networks

OS support for IP: **IP driver**



IP driver responsibilities

- Get operating parameters from device driver
  - Maximum packet size (MTU)
  - Functions to initialize HW headers
  - Length of HW header
- Routing packets
  - From one physical network to another
- Fragmenting packets
- Send operations from higher-layers
- Receiving data from device driver
- Dropping bad/expired data

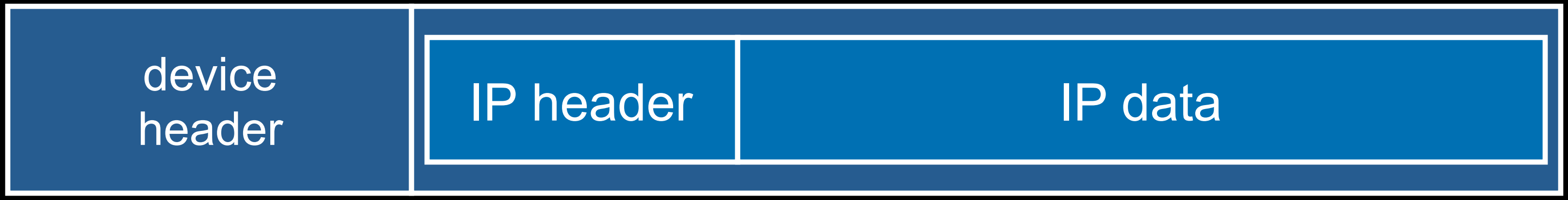
Network device driver responsibilities

- Controls network interface card
  - Comparable to character driver
- Processes interrupts from network interface
  - Receive packets
  - Send them to IP driver
- Get packets from IP driver
  - Send them to hardware
  - Ensure packet goes out without collision

Packet encapsulation

- Network card examines packets on wire
  - Compares destination addresses

- Before a packet is sent, it must be **enveloped** for the physical network



Addressing The Device

To send an IP packet, we need to know the *ethernet address* that corresponds to the *IP address* (or the ethernet address of the router that should get the packet)

Address Resolution Protocol (ARP)

Find the ethernet address for a given IP address:

1. Check local ARP cache
2. Send broadcast message requesting ethernet address of machine with certain IP address
3. Wait for response (with timeout)

Routing

Router

- Switching element that connects two or more transmission lines (e.g., Ethernet)
- Routes packets from one network to another
(OSI layer 3 – Network Layer)
- Special-purpose hardware or a general-purpose computer with two or more network interfaces

Routing

- Packets take a series of **hops** to get to their destination
  - Figure out the path
- Generate/receive packet at machine
  - check destination
    - If destination = local address, deliver locally
  - else
    - Increment hop count (discard if hop # = TTL)
    - Use destination address to search **routing table**
    - Each entry has address and netmask. Match returns interface
    - Transmit to destination interface
- **Static routing**

Dynamic Routing

- Class of protocols by which machines can adjust routing tables to benefit from load changes and failures

- Route cost:
  - Hop count (# routers in the path)
  - Time: Tic count – time in 1/18 second intervals

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

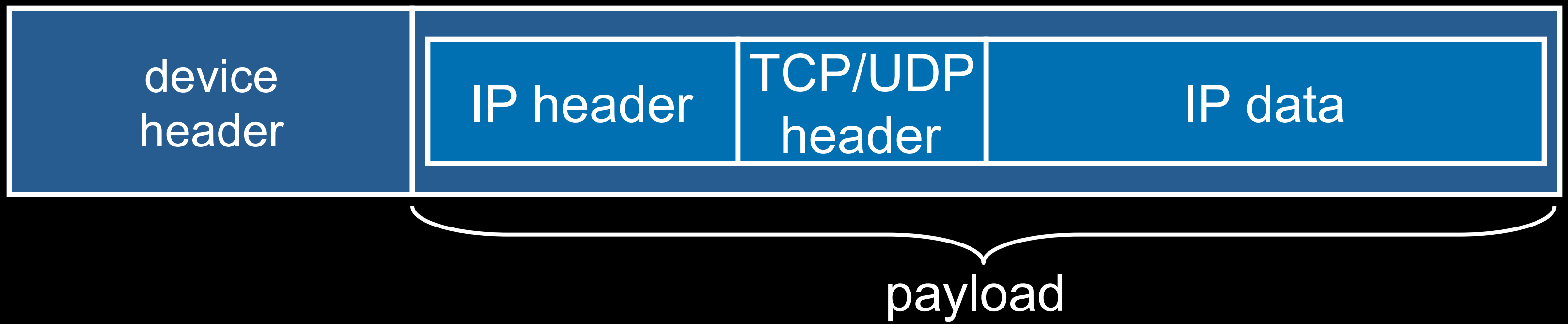
TCP – Transmission Control Protocol

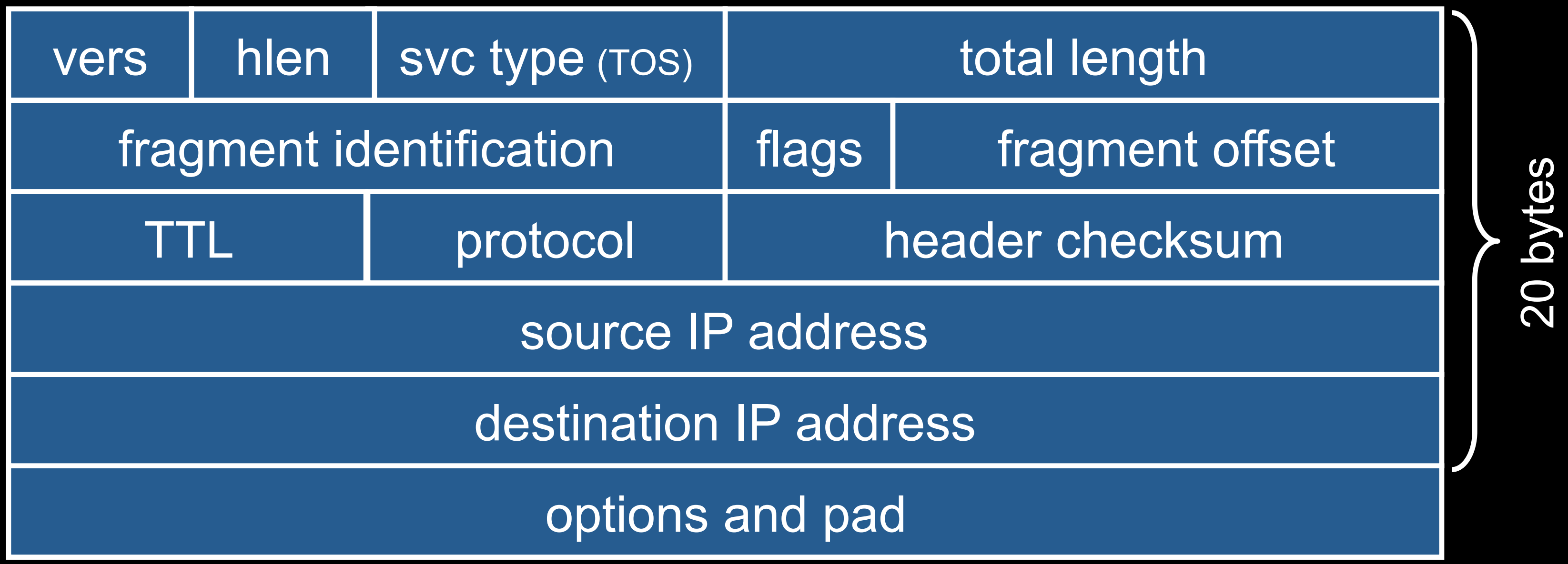
- Virtual circuit service
(connection-oriented)
- Send acknowledgement for each received packet
- Checksum to validate data
- Data may be transmitted simultaneously in both directions

UDP – User Datagram Protocol

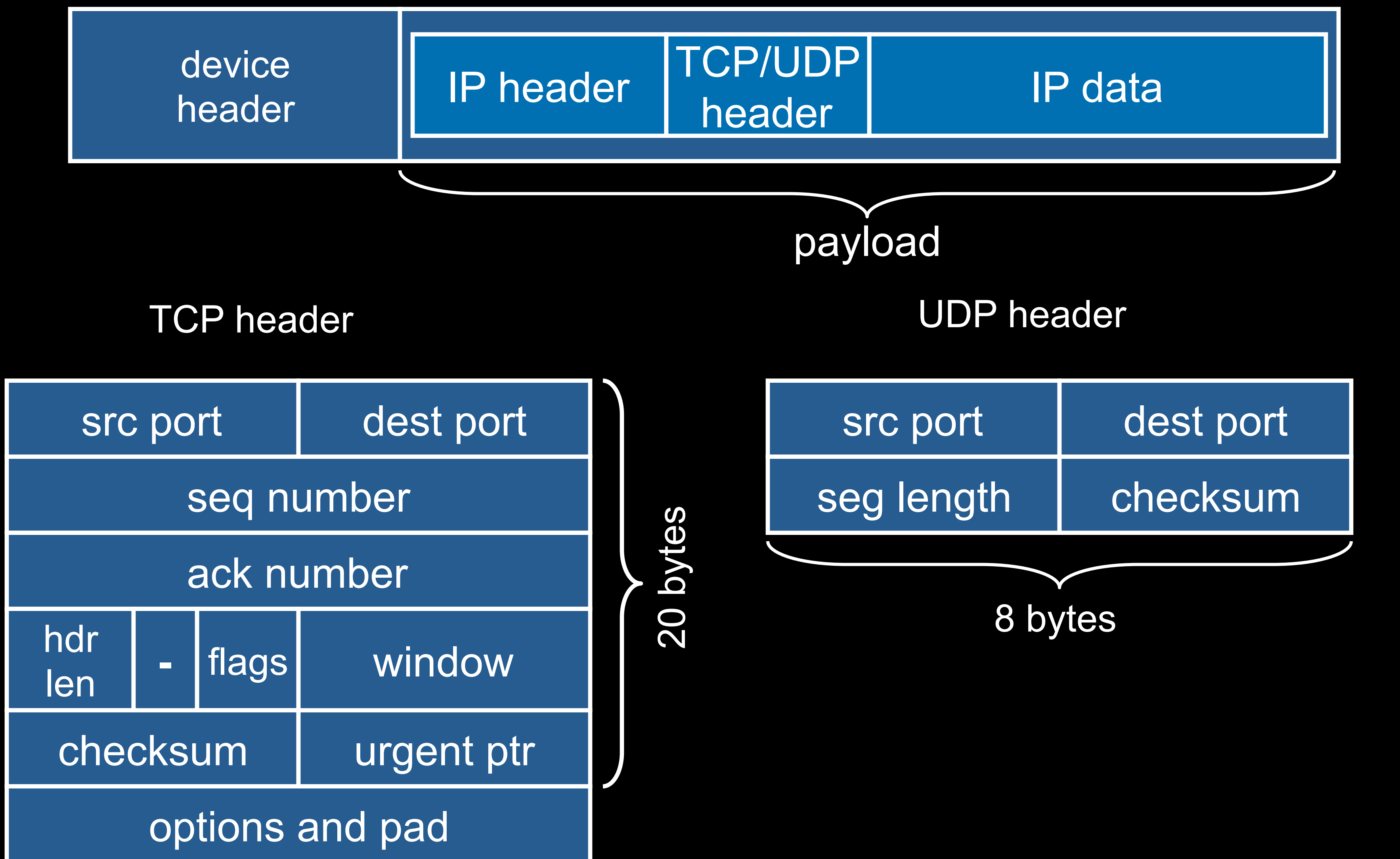
- Datagram service (connectionless)
- Data may be lost
- Data may arrive out of sequence
- Checksum for data but no retransmission
  - Bad packets dropped

IP header

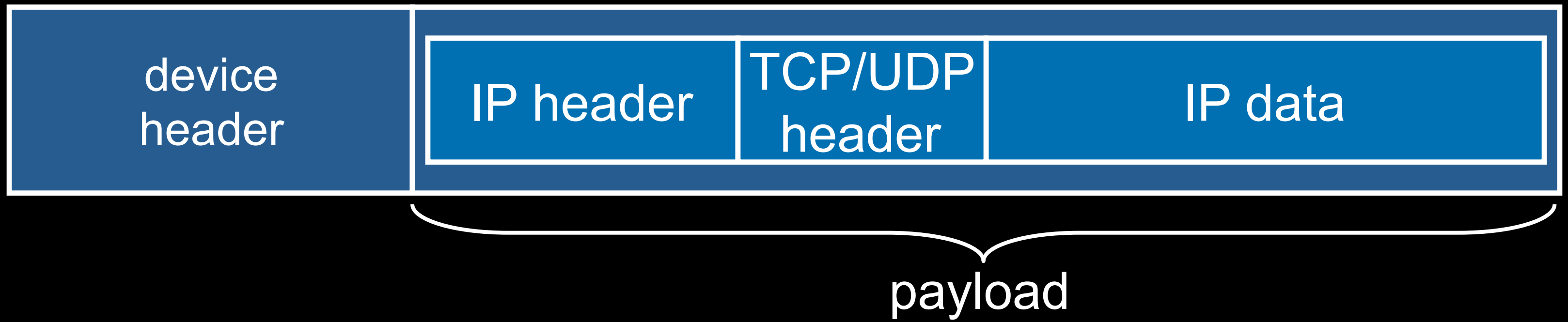

device header
IP header
TCP/UDP header
IP data
payload

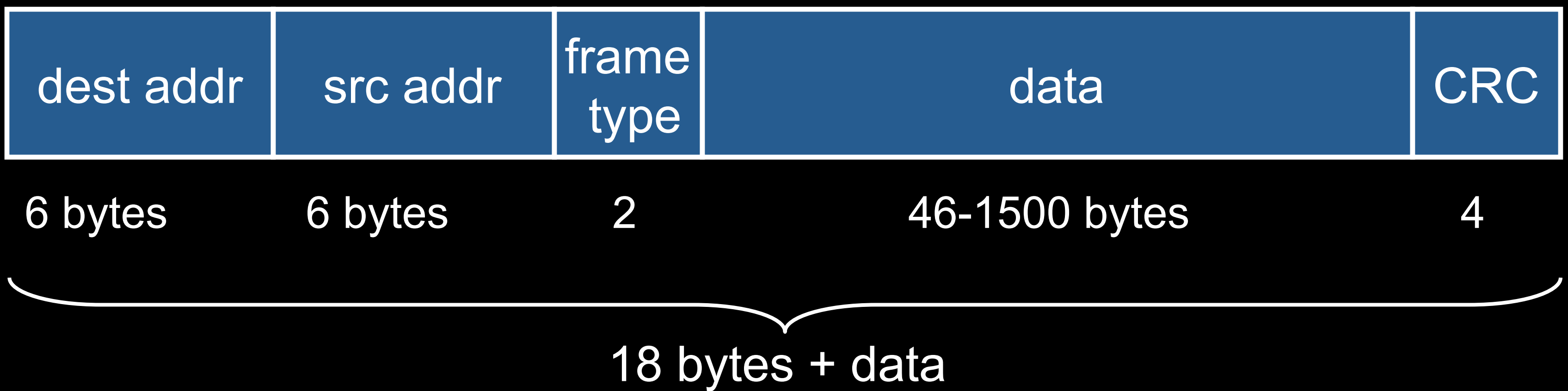

vers
hlen
svc type (TOS)
total length
fragment identification
flags
fragment offset
TTL
protocol
header checksum
source IP address
destination IP address
options and pad
20 bytes

Headers: TCP & UDP



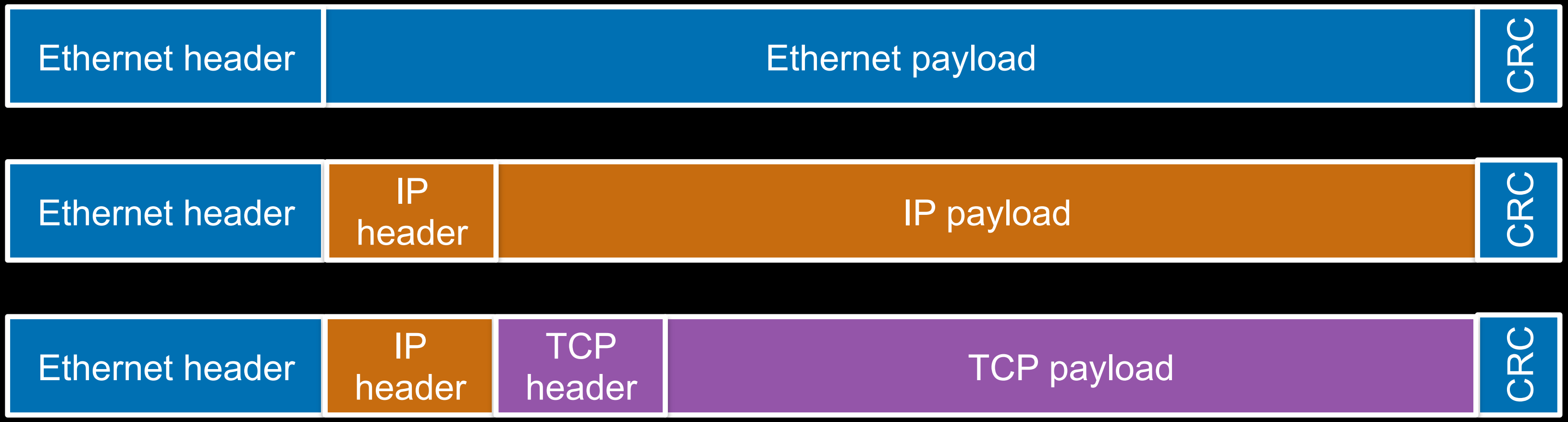
Device header (Ethernet II)





Protocol Encapsulation

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



The End