Logistics

- Class web page relatively complete.
  - Project description, paper description, schedule through week 4, etc…
- I will post papers as they are assigned.
- Project teams assigned. Slight reshuffling late Monday after a couple of people dropped at the last minute.
- Programming assignment #1 will be assigned Thursday.

Today

- Finish chapter 3 on networking principles.
- Get into chapter 4, talking about TCP/IP and interprocess communications.

Network principles

- What are some of the important principles in network technologies?
  - Packet transmission
  - Data streaming
  - Switching schemes
  - Protocols
    - Software layers and protocol “stacks”
  - Routing algorithms and hardware
  - Congestion control
  - Internetworking

Packet transmission and streaming

- Data is moved from one point to another in two different ways typically.
  - Packets: Data moves through the network in fixed length packets that contain the contents of the message, and information relevant for routing and transport protocols.
  - Streaming: Data flows through the network in a more continuous fashion with guarantees and bounds on performance measures like latencies. Popular for multimedia applications.

Switching schemes

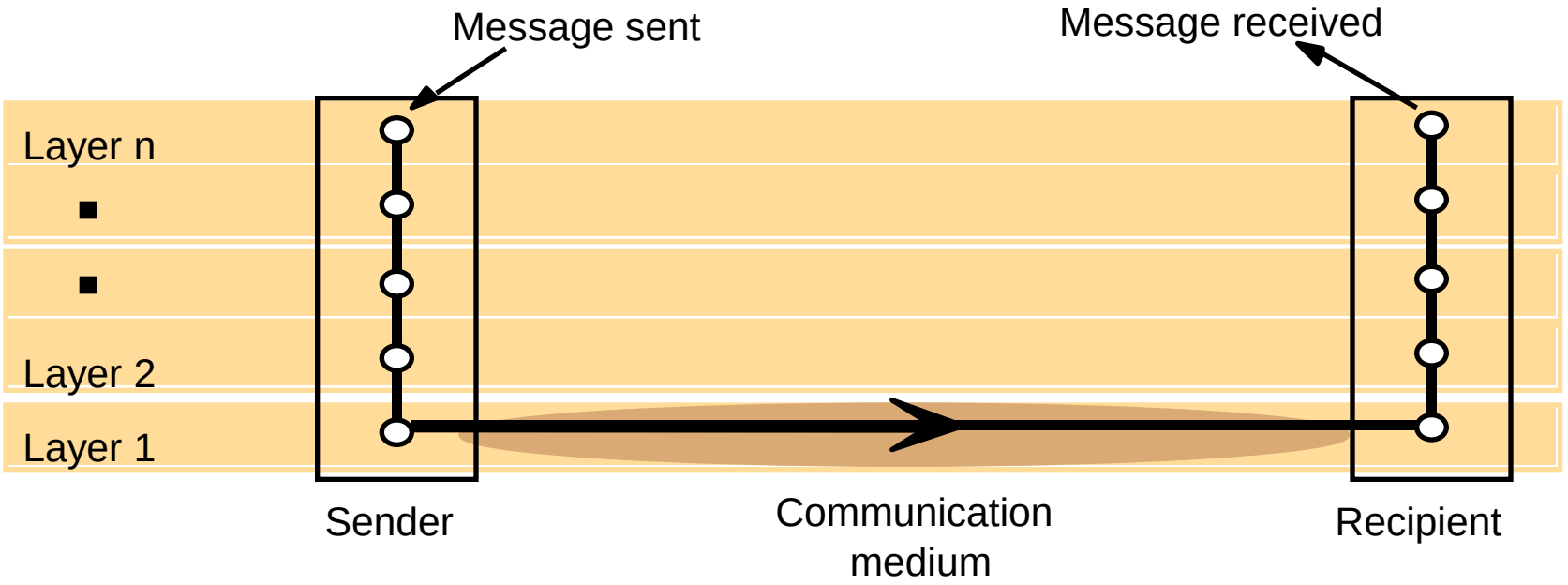
- Broadcast
  - Send to everyone, rely on intended receiver to accept message and others to toss it out.
  - Example: Ethernet
- Circuit switching
  - Inspired by old telephone switching techniques.
  - A path through the network is set up and data streams through.
- Packet switching
  - Store-and-forward
- Frame relay
  - Similar to circuit switching + packet switching
  - Store only enough of packet to make routing decision and then pass data through like a circuit switched network.

Protocols

- A protocol defines a set of rules and formats used when processes communicate with each other.
- Two important parts:
  - 1. Specification of the sequence of messages that are to be exchanged.
  - 2. Specification of the format of these messages.
- Protocols exist for many different levels of abstraction, from low-level blocks of bits on the wire to high-level, application specific abstractions (like HTTP, SMTP, FTP, etc…)

Protocol layers

- Network software protocols are separated into well defined layers.



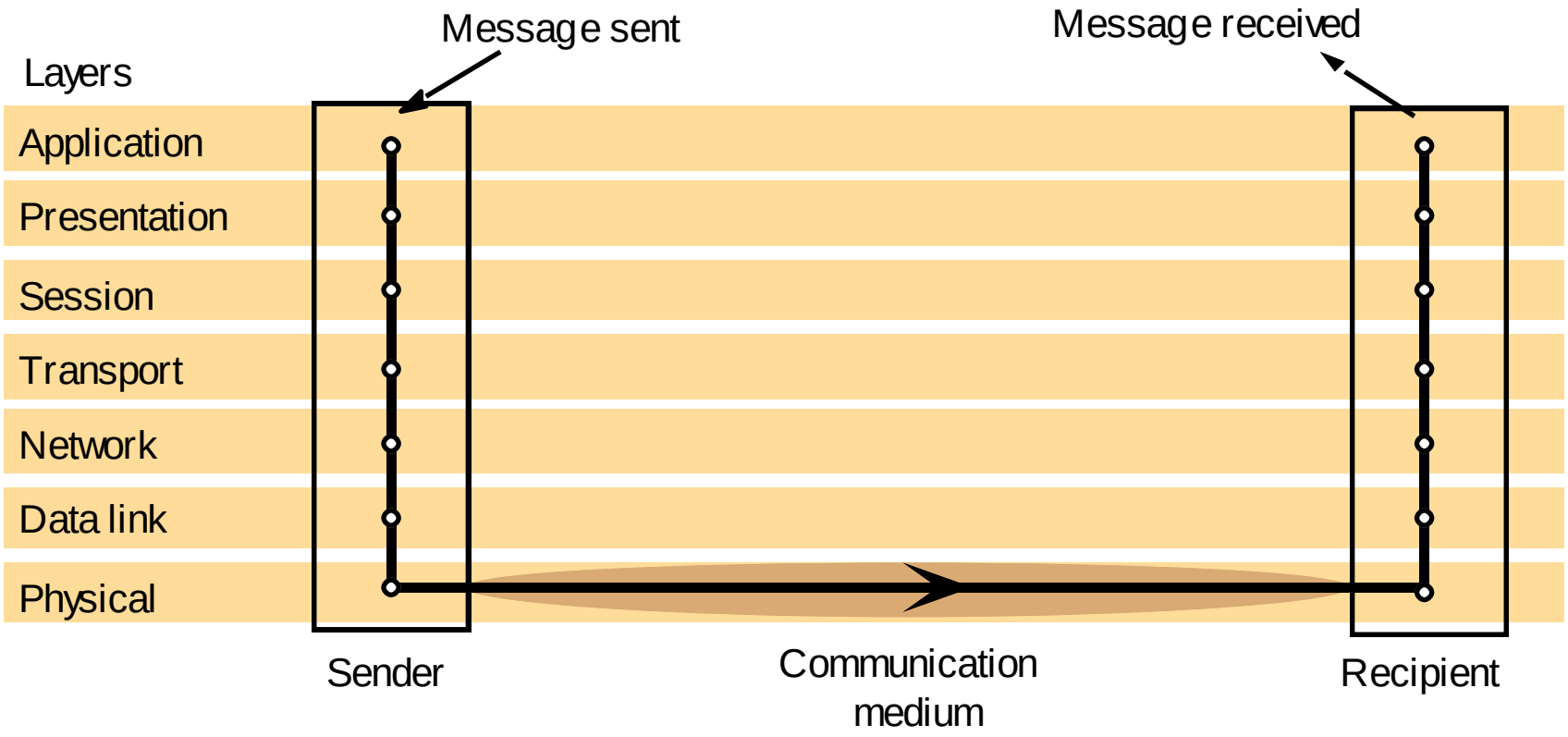
Protocol layers

- The layers are based on abstractions.
- Low levels: Abstraction above the wire protocol.
  - E.g.: Ethernet
- Middle levels: Abstraction of raw data transmission and reliability.
  - E.g.: TCP, IP, UDP
- Higher levels: Abstractions related to specific activities.
  - E.g.: FTP, HTTP
- Highest levels: Abstractions related to specific applications.
  - E.g.: Web services

Standard stack: ISO OSI

- Open Systems Interconnection stack, standardized by ISO.
- 7 layers
  - Application
  - Presentation
  - Session
  - Transport
  - Network
  - Data link
  - Physical

ISO OSI



ISO OSI Layers

Layer	*Description*	*Examples*
Application	Protocols that are designed to meet the communication requirements of specific applications, often defining the interface to a service.	HTTP, FTP, SMTP, CORBA IIOP
Presentation	Protocols at this level transmit data in a network representation that is independent of the representations used in individual computers, which may differ. Encryption is also performed in this layer, if required.	Secure Sockets (SSL),CORBA Data Rep.
Session	At this level reliability and adaptation are performed, such as detection of failures and automatic recovery.	
Transport	This is the lowest level at which messages (rather than packets) are handled. Messages are addressed to communication ports attached to processes, Protocols in this layer may be connection-oriented or connectionless.	TCP, UDP
Network	Transfers data packets between computers in a specific network. In a WAN or an internetwork this involves the generation of a route passing through routers. In a single LAN no routing is required.	IP, ATM virtual circuits
Data link	Responsible for transmission of packets between nodes that are directly connected by a physical link. In a WAN transmission is between pairs of routers or between routers and hosts. In a LAN it is between any pair of hosts.	Ethernet MAC, ATM cell transfer, PPP
Physical	The circuits and hardware that drive the network. It transmits sequences of binary data by analogue signalling, using amplitude or frequency modulation of electrical signals (on cable circuits), light signals (on fibre optic circuits) or other electromagnetic signals (on radio and microwave circuits).	Ethernet base- band signalling, ISDN

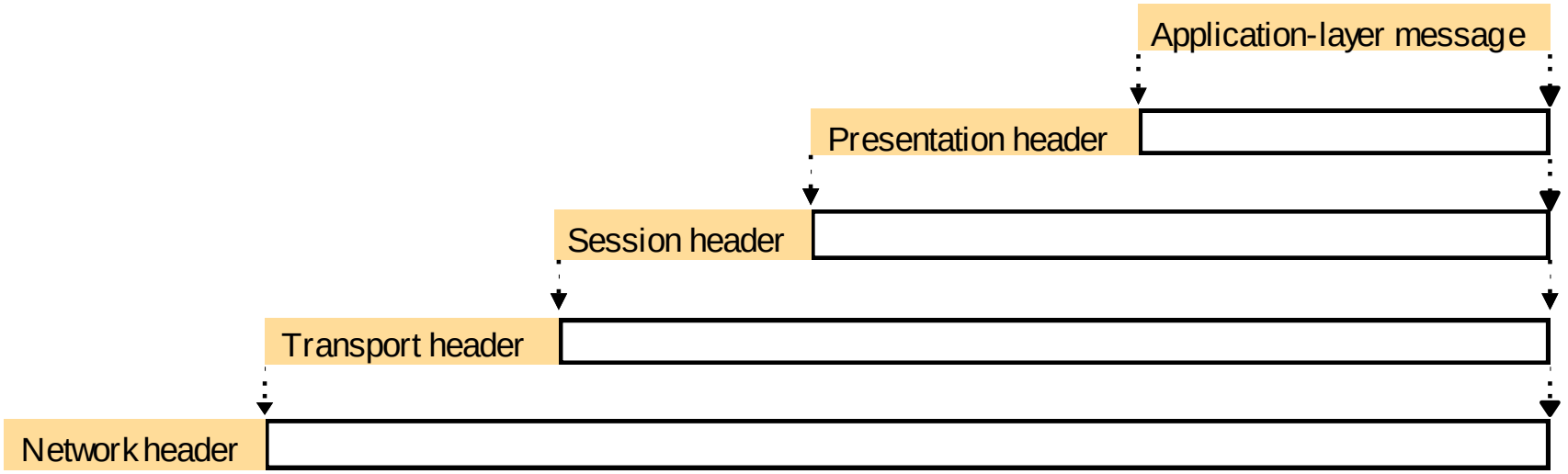
Protocol stacks: Pros and Cons

- Pro:
  - Abstraction allows for different layers to be swapped in and out.
  - Separation of concerns.

- Con:
  - Performance degradation. All messages must traverse stack on both receiver and sending side.
  - Results in observed performance in applications being less than the advertised performance of the hardware.

Encapsulation

- Lower levels encapsulate messages from higher levels.



Encapsulation

- What is nice about encapsulation?
- The message being encapsulated isn't treated as anything more than data by the wrapper.
- This allows you to design layers without having to specify much at all for the other layers.

- We see this in action daily:
  - HTTP over TCP/IP over ethernet.
  - FTP over TCP/IP over ethernet.
  - AppleTalk over ethernet.
    - Ok, this one is a bit old. ☺ Just an example of multiple protocols over ethernet.

Ports and addressing

- The transport layer is responsible for providing ports and addresses.
- Ports are essentially mailboxes or addressing units on a single machine that is associated with a specific program.
  - Program “foo” wants to receive data, so it opens up port 1234 from the outside world.
  - Outside processes connect to 1234, and don’t have to know much about the process “foo”.
- Addresses are provided for hosts also to allow them to find each other.
- The combination of a port and a host address allows processes to find other processes in a well defined, organized way.

Routing

- Routing algorithms and hardware are what determine the path packets take through the network to get from the originator to the receiver.
- The network is considered to be a graph with nodes being routers, switches and hosts. The edges are the links between them. The edges are typically weighted with a cost of traversing the path based on performance characteristics of the link.
- A route is a path through this graph.

- Routes can be computed statically or dynamically. Adaptive routing is a dynamic scheme that addresses changing conditions in the network due to load variations and failures within the network.
  - The internet is based on adaptive routing for the most part.

Routing algorithms

- In ch. 3.3, you will see an example of a simple algorithm for routing.
- Tables are maintained at routers that represent the link to take to get closer to the destination. Traversing a link is often called taking a “hop”.
- Routers periodically update their tables based on conditions changing, and exchange them with each other.
- The routing algorithm must deal with undesirable states such as forming loops, and routing around parts of the network that vanish.

Congestion control

- Congestion control addresses dynamic loads on the network.
- If a burst of activity occurs, we can find parts of the network get bogged down and we risk things like packet loss and buffer exhaustion.
- Congestion control algorithms are built into protocols and network layers to gracefully deal with this.
- Techniques might include holding packets when a node further along the path gets saturated, or dynamically scaling message delays or timeouts.

Internetworking

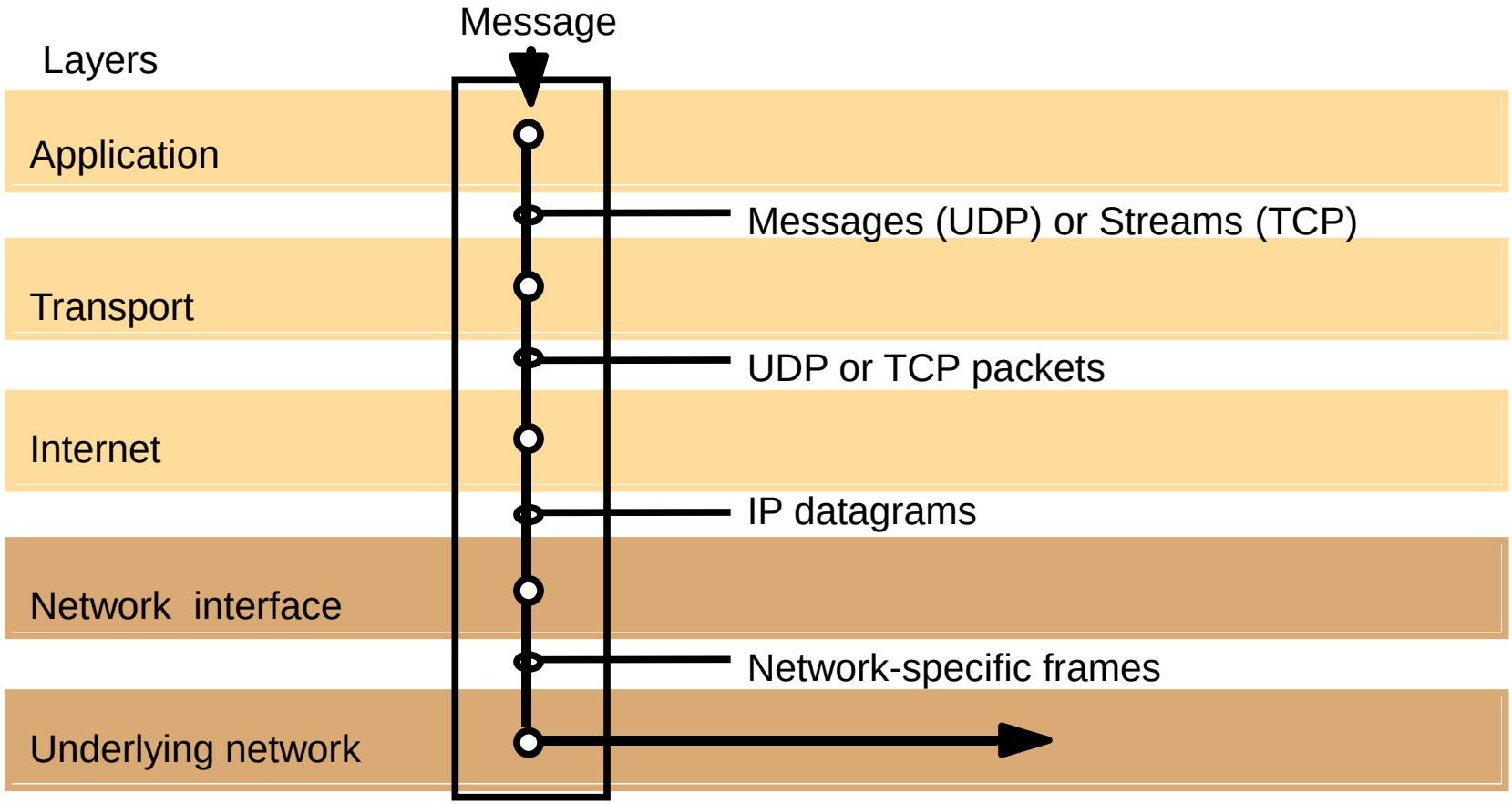
- Integrated networks composed of many subnetworks connected together.
- The internet is a prime example of this.
- Internetworking requires a method to find hosts on different subnets (addressing), and a protocol for communicating between subnets (such as IP).
- These live above the actual subnet technology so that heterogeneous types of networks can all interoperate.

Protocols for the Internet

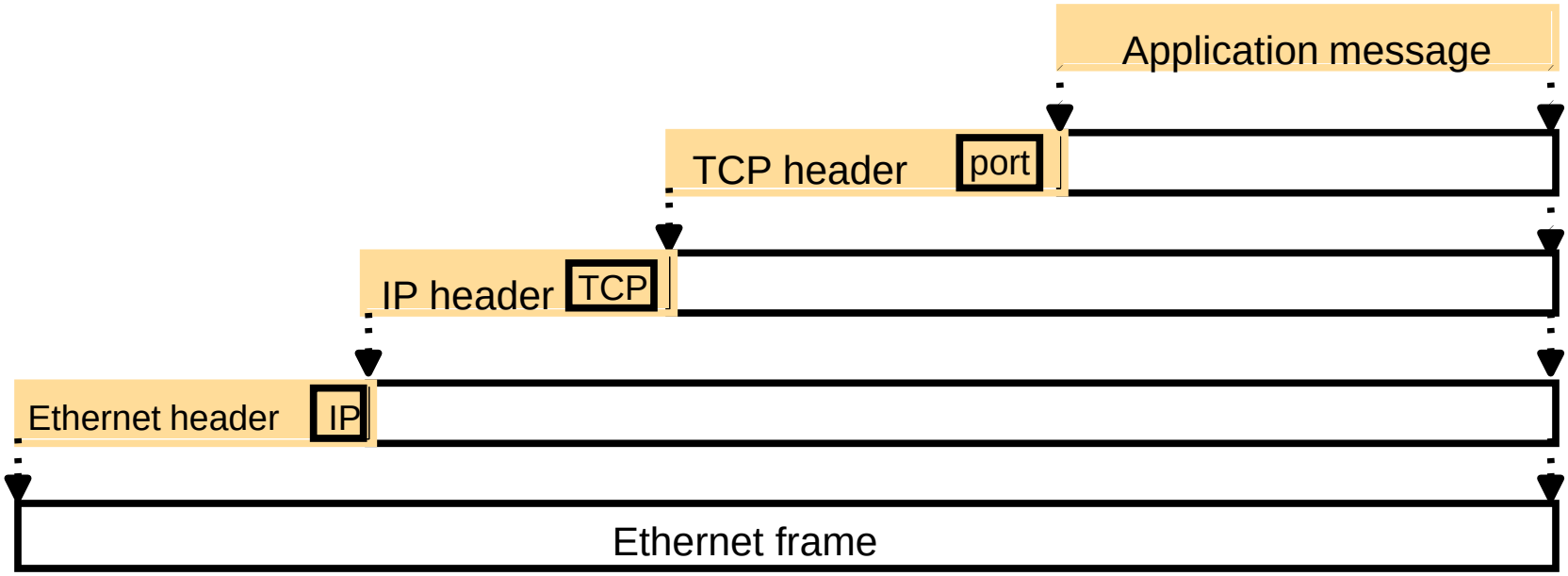
- The main internet protocols originated in the 1970s on the ARPANET project.
- The TCP/IP protocol was an important development of that project.
- TCP: Transmission Control Protocol
- IP: Internet Protocol

- The internet protocols are an example of a successful solution to the openness issue we discussed last time. The open standards allowed for widespread adoption, and the internet we see today.

TCP/IP and the protocol stack



TCP/IP and encapsulation



Successful abstraction

- The TCP/IP specification is nice because it doesn't specify anything about what happens below the Internet datagram layer.
- IP packets can be encapsulated within packets that are transmitted over most every lower level network technology.

Transport protocols

- The dominant transport protocols are TCP and UDP.
- TCP
  - Transport Control Protocol
  - Connection-oriented, reliable protocol
- UDP
  - User Datagram Protocol
  - No guarantee of reliable transmission

- These are layered above IP.

IP addressing

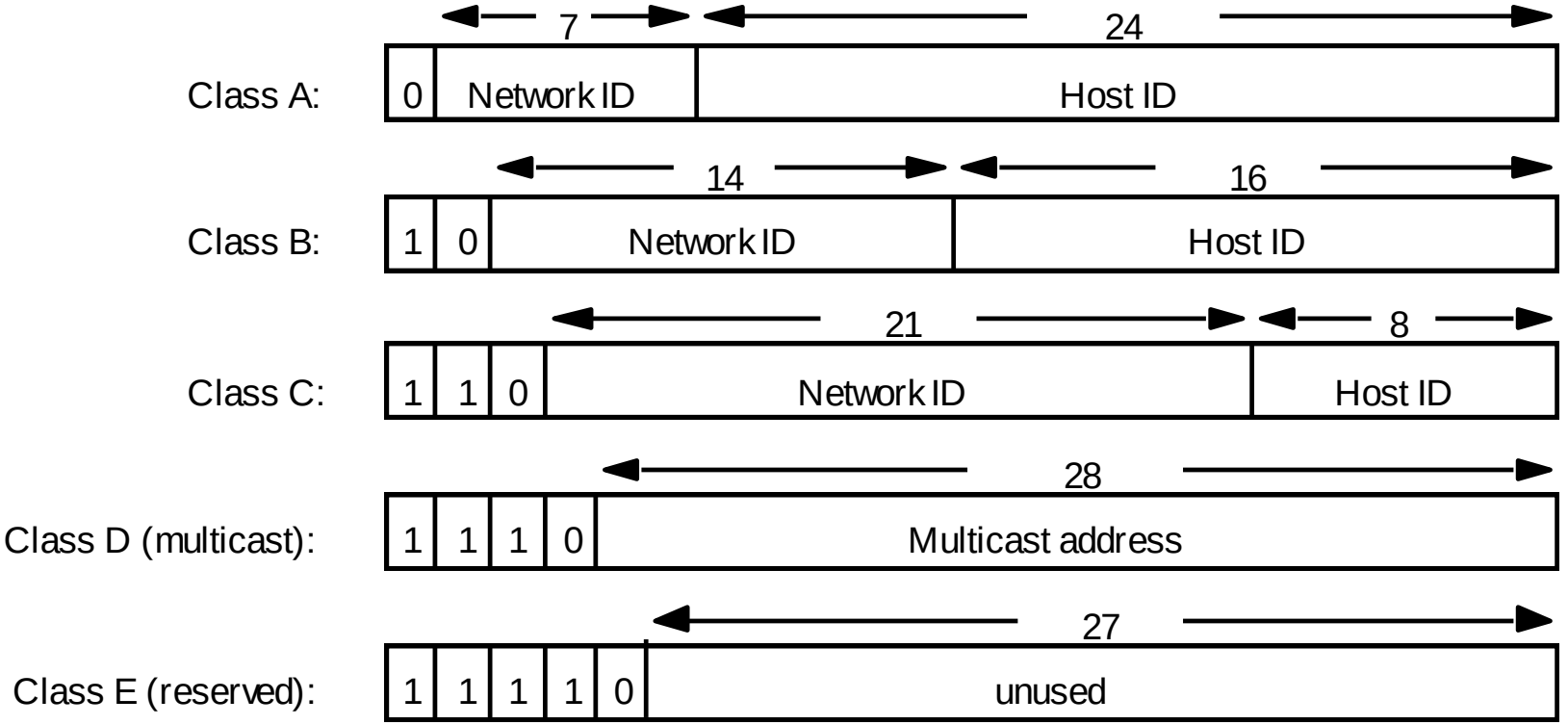
- Network layer protocol.
- Responsible for routing and addressing.

- We are all likely familiar with IP addresses : the numeric addresses of machines on the internet, such as 128.223.32.35.

- The addressing problem was one of the early challenges to building large scale, scalable networks.

IP addressing

- IP addresses are built out of four octets, or 8-bit numbers.

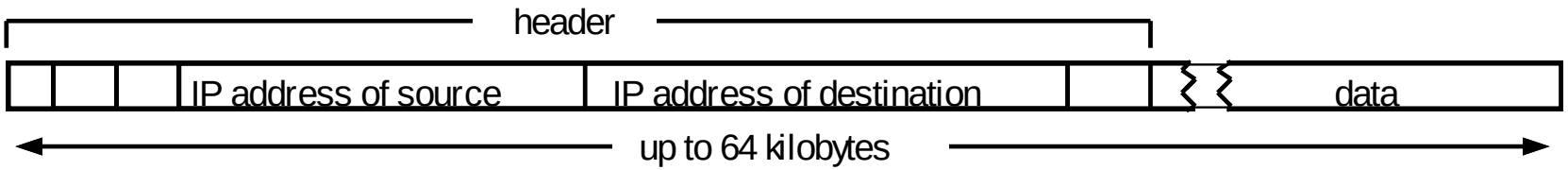


IP addressing

- IP assigns one address to each host in the Internet.
- The classes of addresses were intended to meet the needs of different organizations.
  - Class A: Huge organizations (e.g.: NSFNet), 2^24 hosts per subnet.
  - Class B: Large organizations with more than 255 hosts.
  - Class C: Less than 255 hosts.
  - Class D: Multicast
  - Class E: Reserved for future use.
- In recent times, this scheme has reached the limits of scalability. It turned out not to be the most efficient use of the address space for the modern Internet. Then again, the ARPANET designers never saw much of the current usage pattern in the future of their network.

IP packet structure

- The IP protocol is complex, but the basic structure has the form:

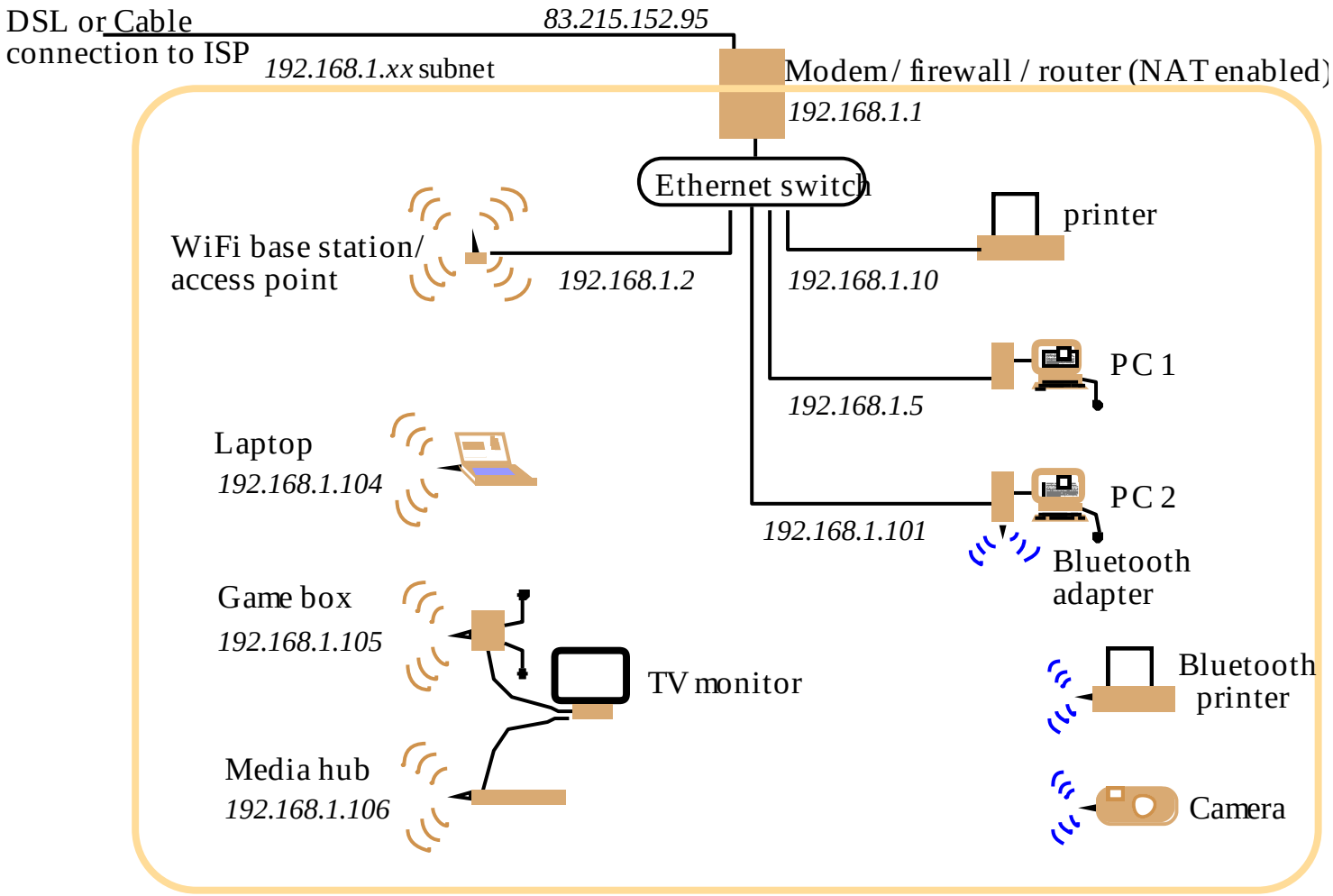


- The IP packet has enough information for a router in the network to make decisions related to how to move the packet from the source to the destination.
- Deeper details on the packet structure in Chapter 3.4.2 and 3.4.3 for routing.

Intermediate solutions for scalability

- As I mentioned last week, solutions exist to allow the internet to scale with large numbers of personal area networks and local area networks.
- Network Address Translation (NAT) enabled routers (such as your Linksys wireless router at home) use reserved addresses, such as 192.168.1.x, for your devices, hiding them behind a single “real” IP address that it holds.
- This removes the need for these devices to have their own IP addresses in the global pool, allowing for single real IP addresses to be multiplexed to multiple devices.

NAT at work



IPv6

- IPv4 is the current IP in use on the Internet. IPv6 was designed primarily to tackle the address exhaustion of IPv4.
- 128-bit address space.
  - Tananbaum: 7x10^23 IP addresses per square meter of Earth.
  - Huitema: More conservatively, 1000 IP addresses per square meter. (Including oceans)
- Either way: IPv6 has far more address space than IPv4.

Migrating IPv4 to IPv6

- Given that IP is the fundamental protocol of the Internet, changing it is not easy.
- The approach: build “islands” of IPv6 routers between IPv4 ones, and slowly grow the IPv6 ones.
- Fortunately, the IPv4 address space is embedded in the IPv6 space, so IPv6 routers can handle IPv4 traffic.
- There is no real technical issue here: it’s just a matter of time to get developers and device manufacturers to switch over.

Process-to-process communication

- IP is concerned with host-to-host communication.
- Distributed systems require process-to-process communication above this.
- TCP and UDP serve this purpose.
- TCP and UDP handle dispatching packets to ports on a host once IP delivers the data.

UDP

- UDP is very basic.
- UDP is very close to the IP layer, so it incurs less overhead than TCP.
- UDP is unreliable. Beyond a checksum to deal with data integrity, UDP doesn't guarantee that a packet won't be dropped.
- There is no acknowledgement that messages are received.

- Has a use in specific applications if the developers can benefit from the lower overhead in the presence of unreliability.

TCP

- TCP provides reliable transport.
- Stream-oriented, so arbitrarily long messages can be sent (vs. the fixed length messages of UDP).
  - Handshaking occurs to establish a bidirectional channel between processes on either end of the connection.
  - Sequencing: One can determine the position of packets in the larger stream.
  - Flow control: As mentioned earlier, one of the design concerns with networks is congestion control. TCP defines a method for timing out and retrying transmission when congestion occurs.
  - Buffering is provided by TCP.
  - Checksums for data integrity checks.

Naming

- The internet also provides domain name services.
- Named entities in the network are called domains.
- The symbolic name of a domain is, obviously, a domain name.
- Host and subnet names are prefixes on these domains.
- Hierarchical naming.
- E.g.: www.cs.uoregon.edu