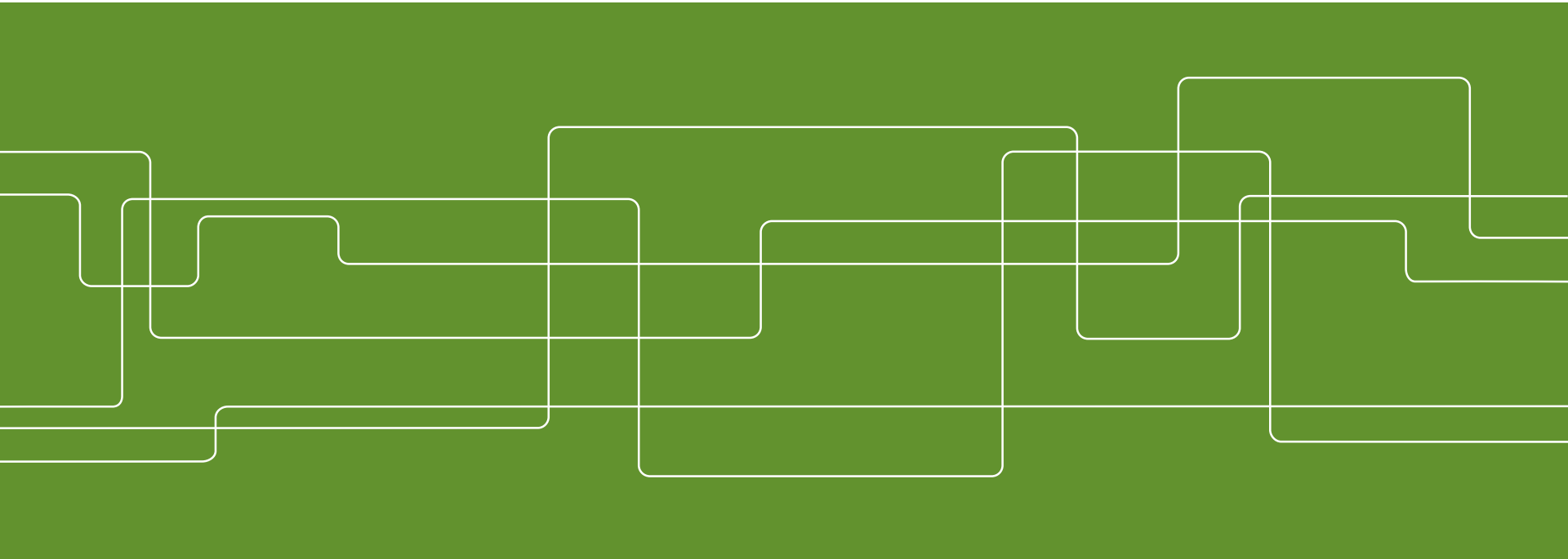
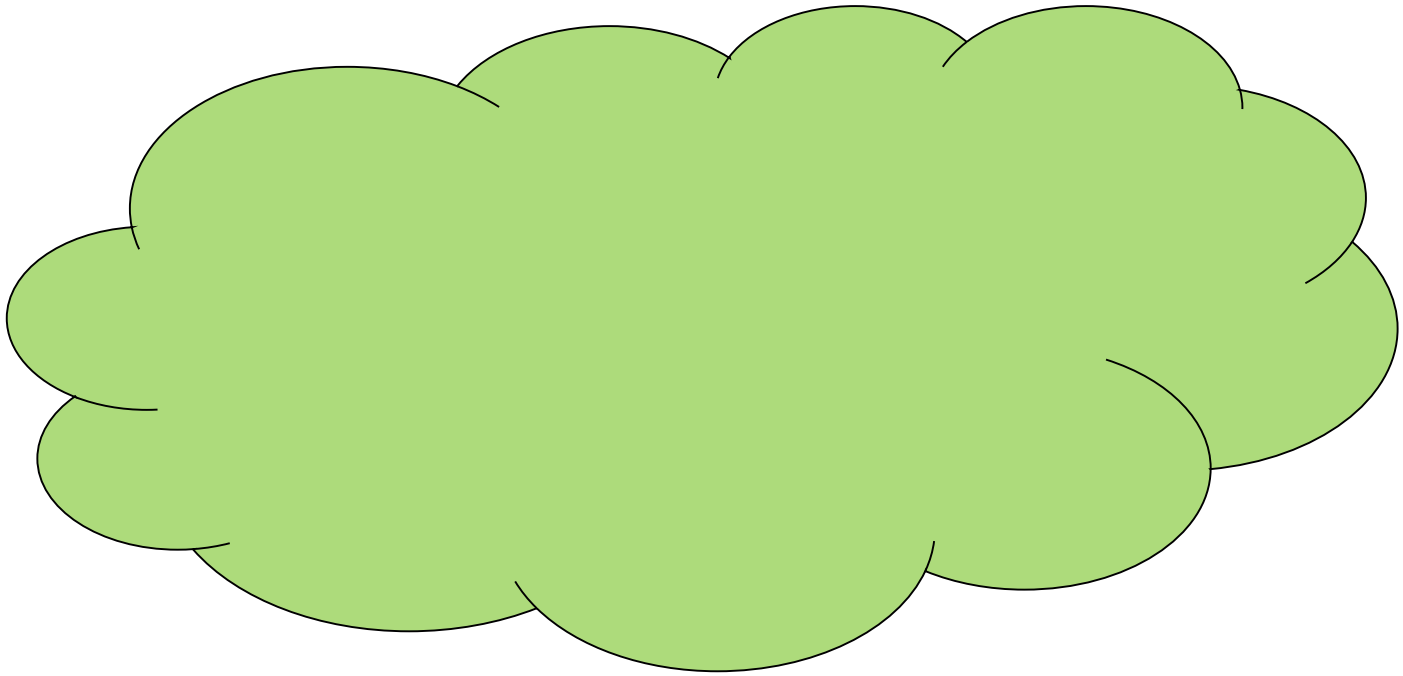
IK2215 Advanced Internetworking

Lecture 1—Recapitulation

Markus Hidell



What is the Internet?



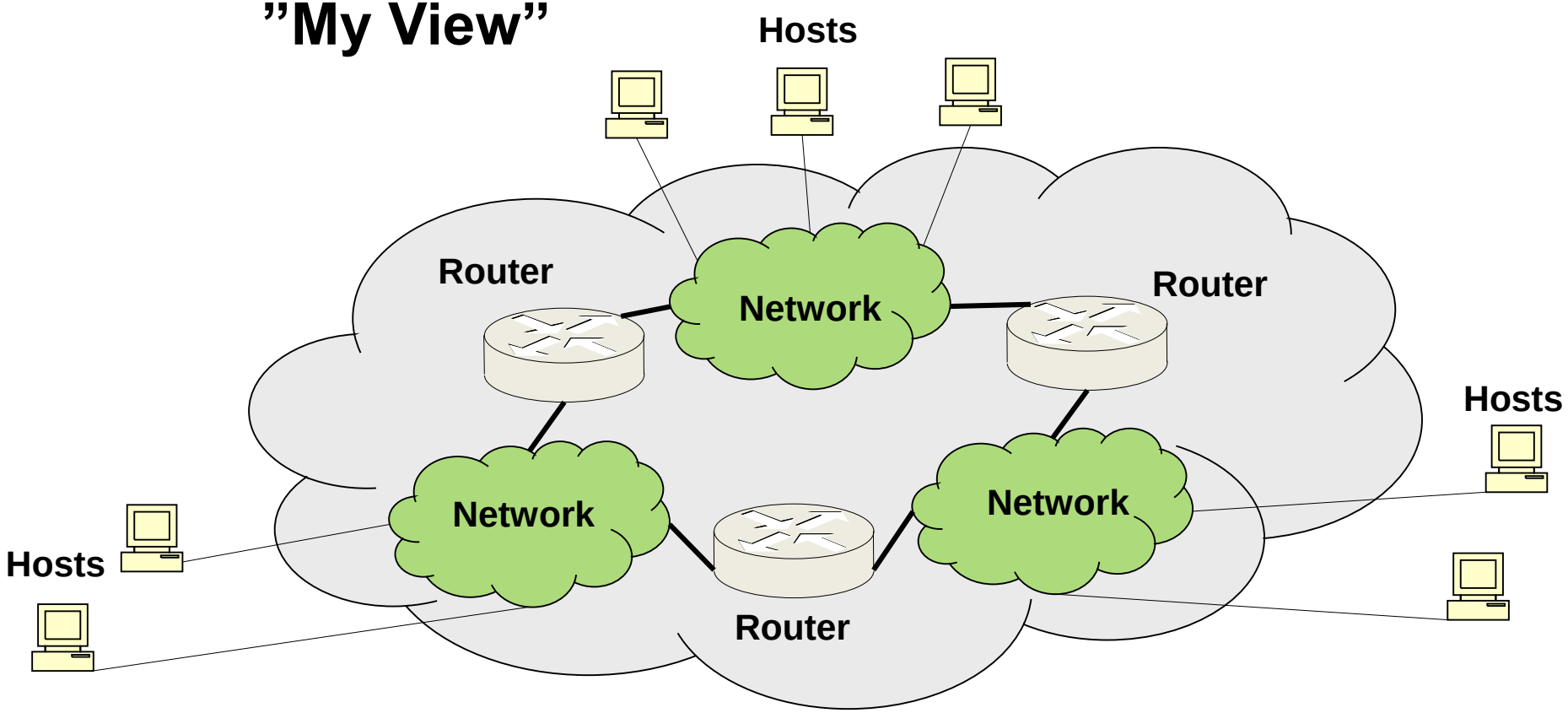
A network?

The web?

Routing?

Domain Name System?

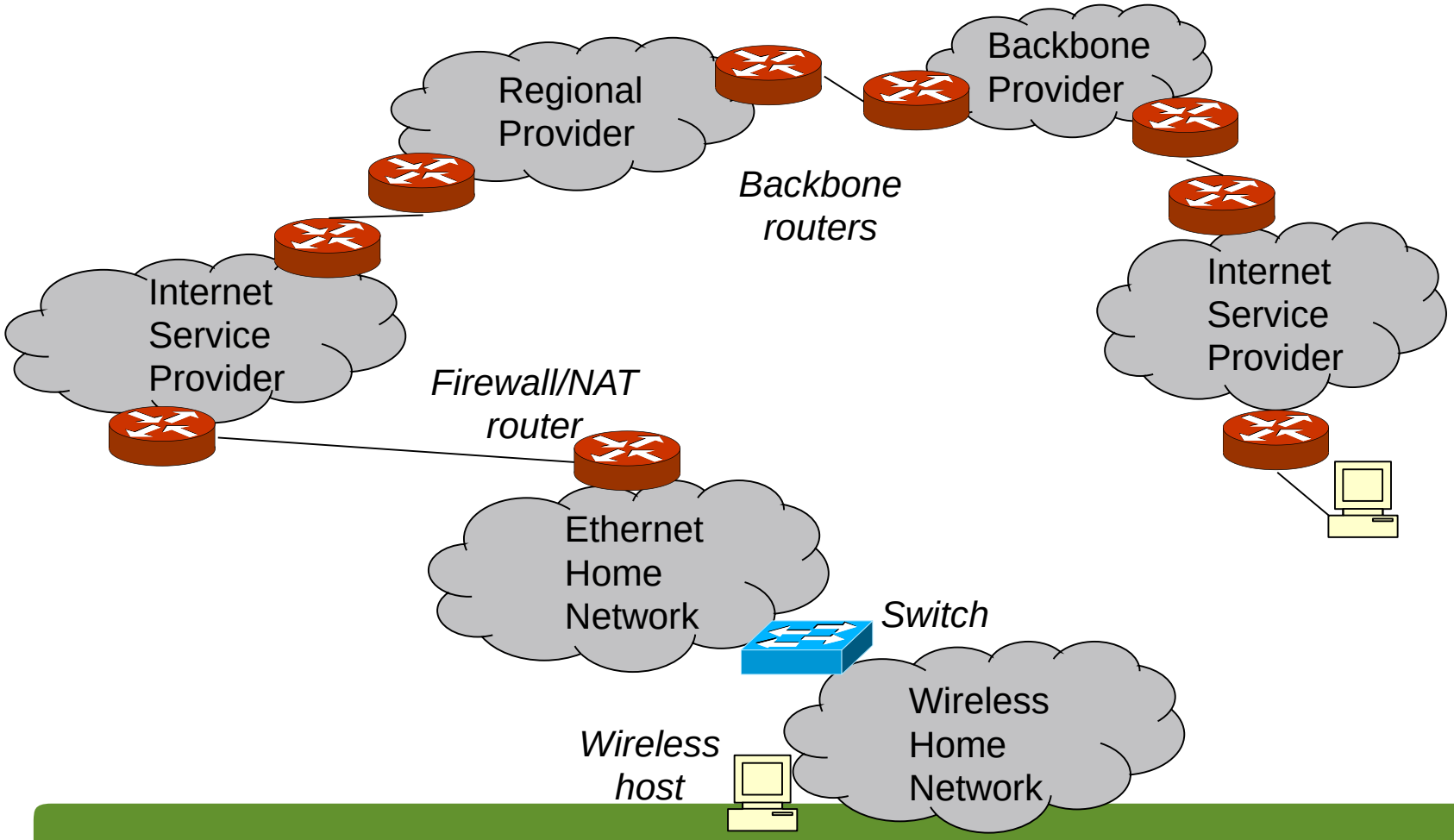
”My View”



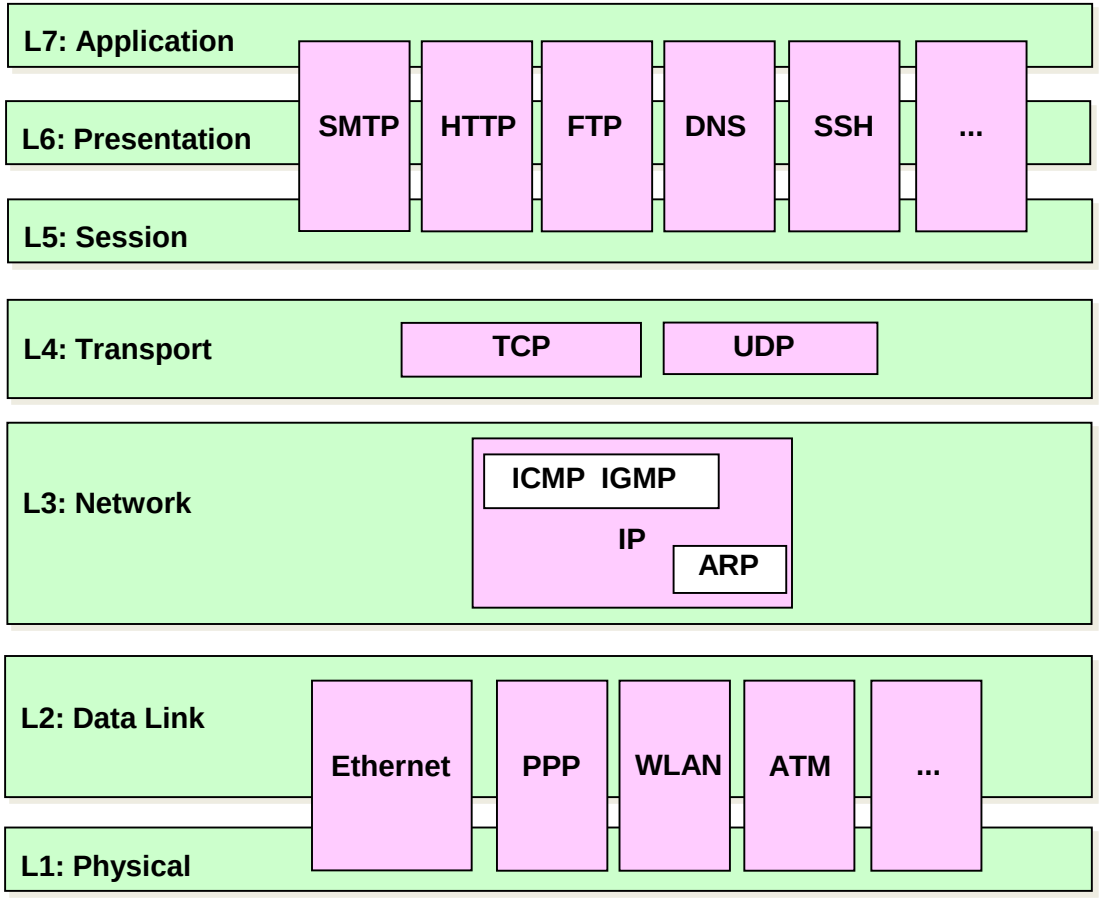
The Internet—hosts, routers, networks, and the TCP/IP protocols

Hosts send data packets that are forwarded by routers

End-users, ISPs, Internet Backbones



OSI and the TCP/IP Protocol Stack



The End-to-End Argument

A specific application-level function should not be built into the lower levels of the system.

The functions “in” the Internet are simple and general.

The bulk of functions are in software at the “edge”.

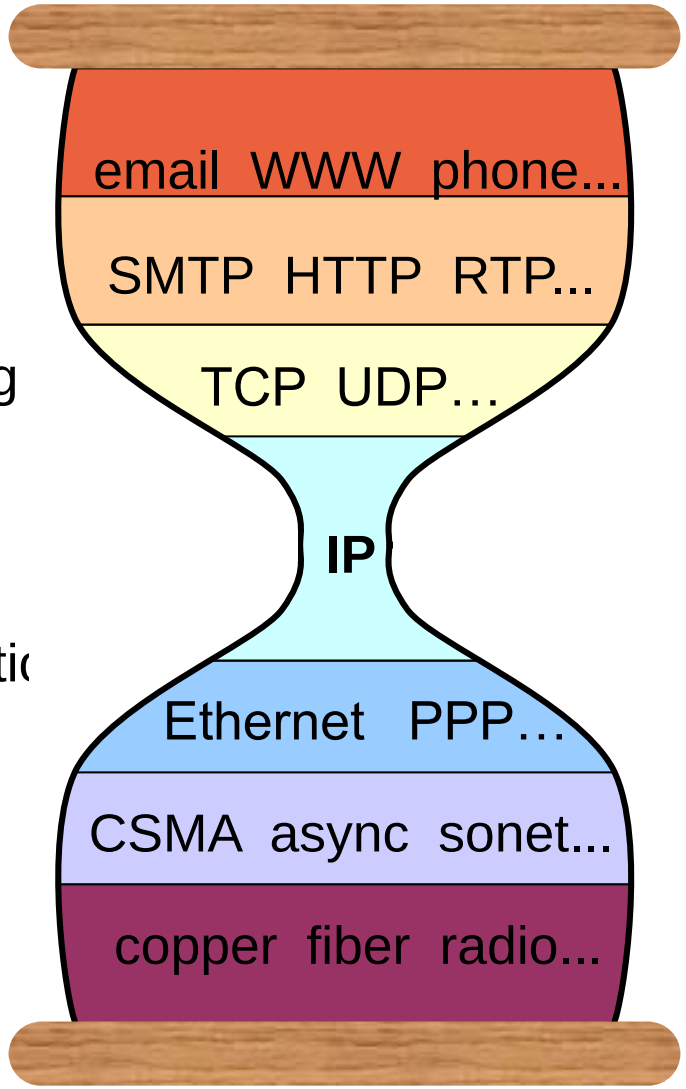
The complexity of the core network is reduced.

Generality in the network increases the chances that new applications can be added.

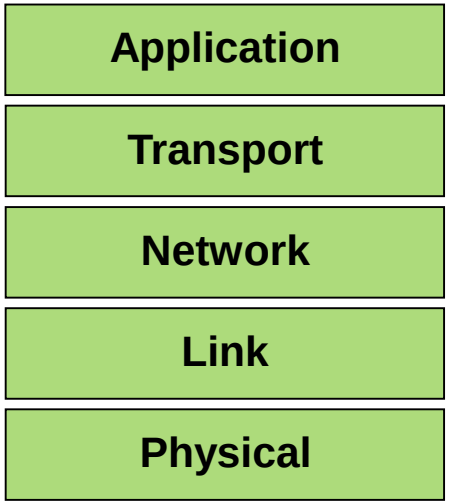
Saltzer, Reed, Clark, 1984, Blomenthal & Clark 2001

The Hourglass Model

- Anything over IP—IP over anything
- All applications depend on IP
- IP runs over all networks
- IP is at the heart of all communicati



The TCP/IP Protocol Stack



Application: supporting network applications

- FTP, SMTP, HTTP

Transport: process-process data transfer

- TCP, UDP

Network: routing of datagrams from source to destination

- IP, routing protocols

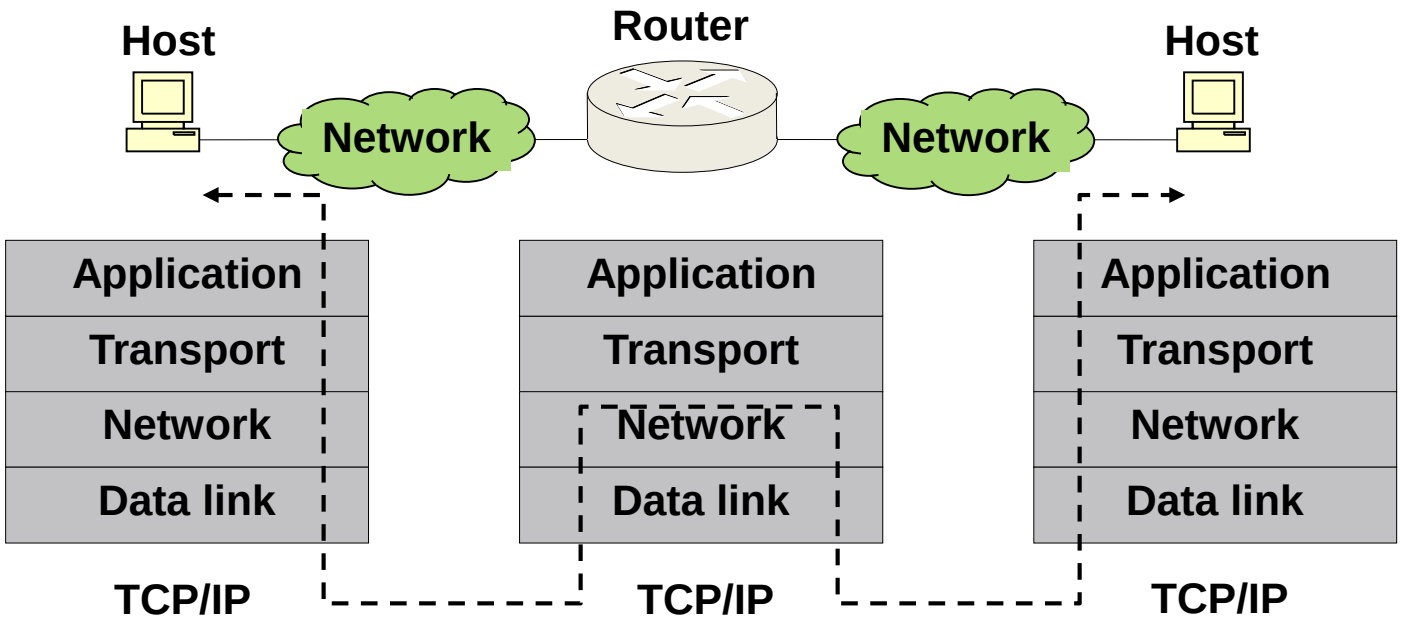
Link: data transfer between neighboring network elements

- PPP, Ethernet

Physical: bits “on the wire”

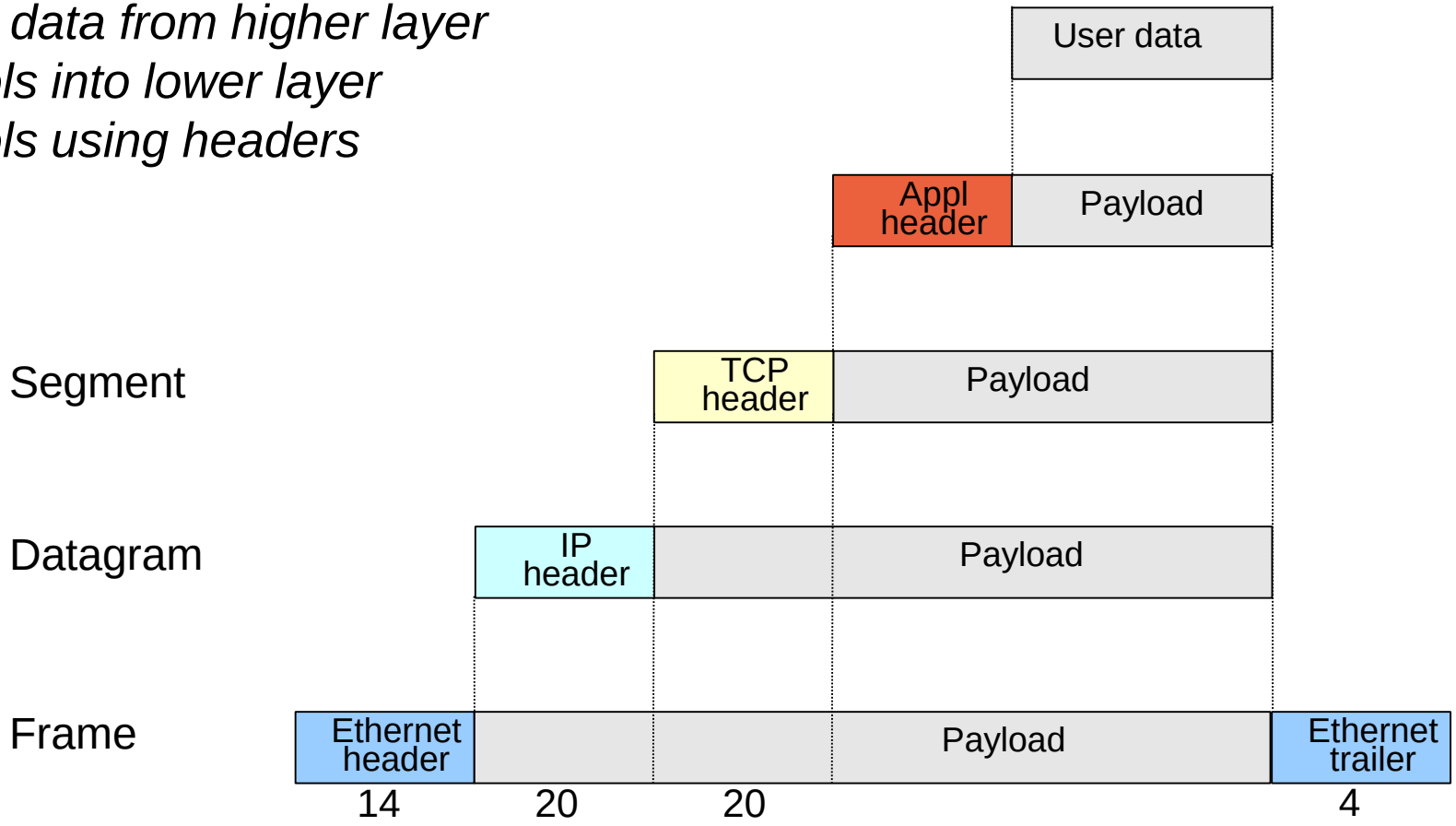
In some literature, Link and Physical layers are merged into one layer

TCP/IP Networking

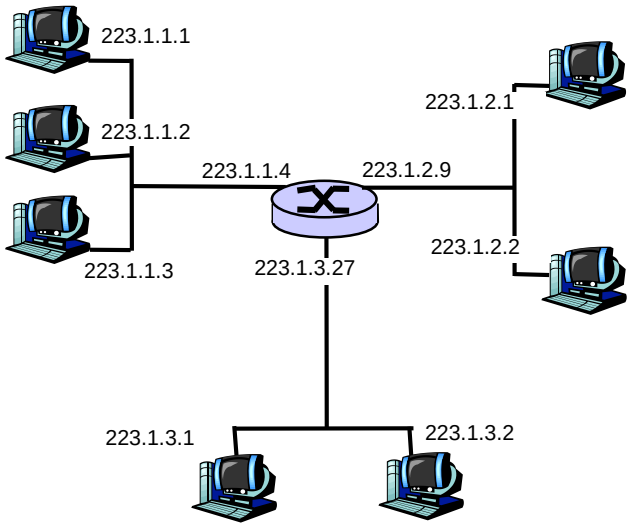


Packet Encapsulation/Decapsulation

Include data from higher layer protocols into lower layer protocols using headers



IP Addressing



IP address:

- 32-bit identifier for host or router *network interface*

Network interface: connection between host/router and physical link

- Router typically has multiple network interfaces
- Host typically has one network interface (or few)
- IP addresses associated with each network interface

223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

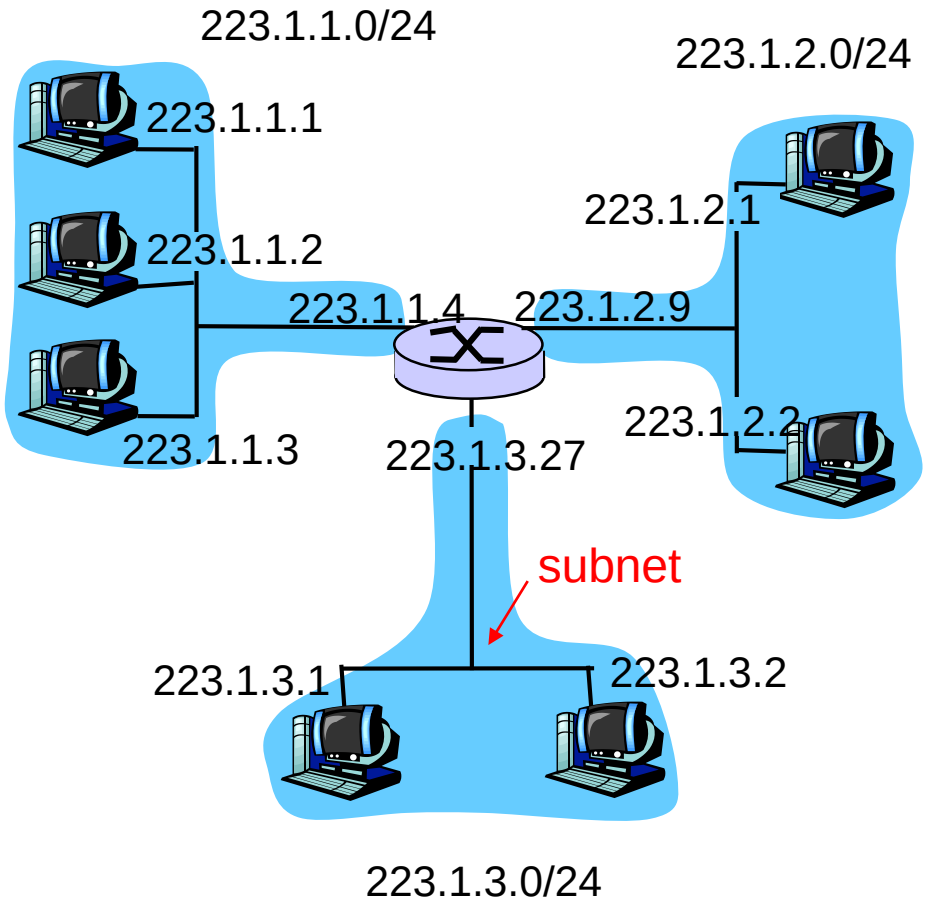
IP Subnets

IP address

- Subnet part (high order bits)
- Host part (low order bits)

What's a subnet ?

- Device interfaces with same subnet part of IP address
- Can physically reach each other without intervening router



Network consisting of 3 subnets
Subnet mask: 255.255.255.0 or /24

IP address AND subnet mask = Net ID: 223.1.3.2 AND 255.255.255.0 = 223.1.3.0

IP Addressing and CIDR

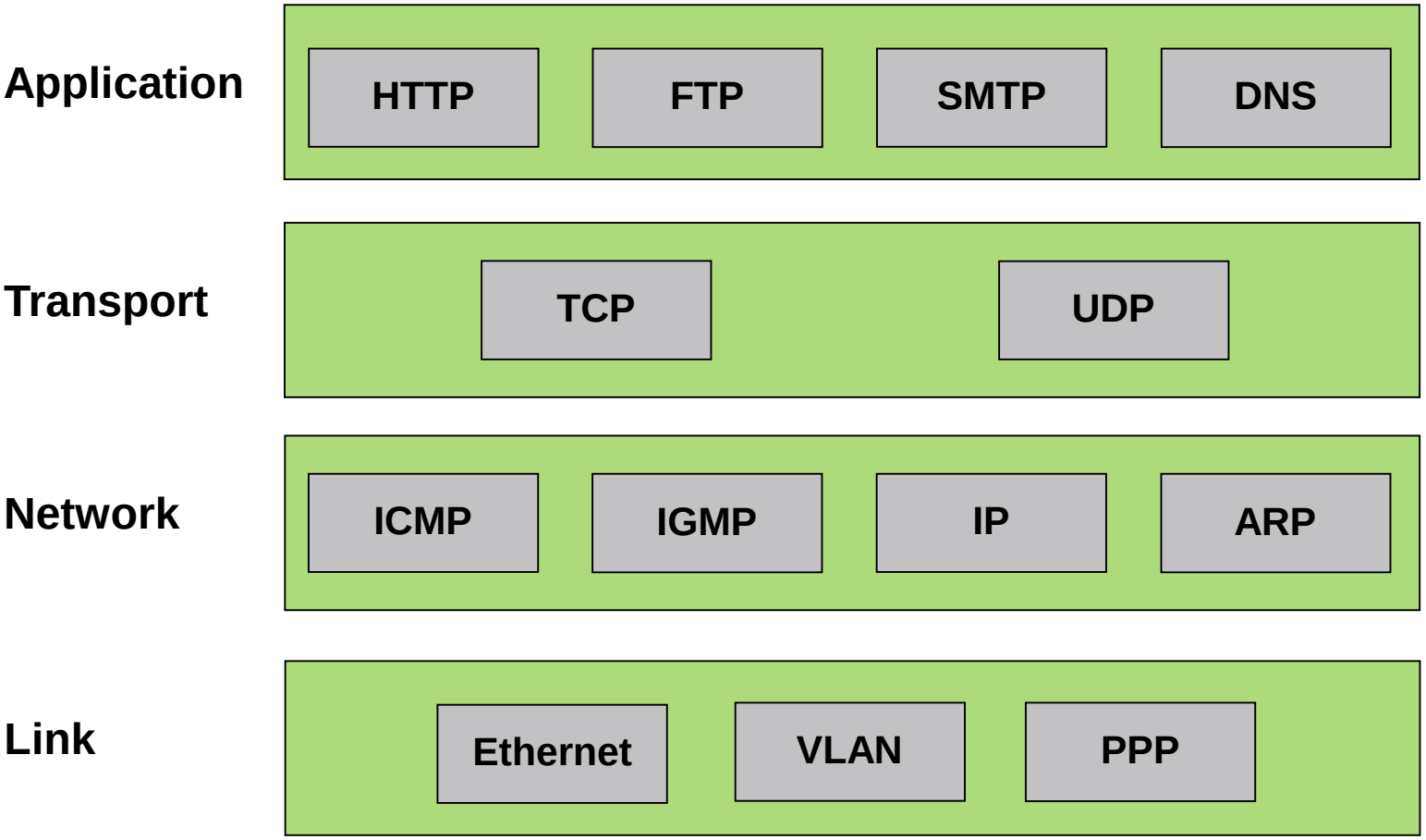
CIDR: Classless Inter-Domain Routing

Subnet portion of address of arbitrary length

Address format: a.b.c.d/x, where x is # bits in subnet portion of address

200.23.16.0/23

Some Basic TCP/IP Protocols



Network Layer

Connection-Oriented Services

- The network layer establishes a connection between a source and a destination
- Packets are sent along the connection.
- The decision about the route is made *once* at connection establishment
- Routers/switches in connection-oriented networks are stateful

Connectionless Services

- The network layer treats each packet independently
- Route lookup for each packet (routing table)
- IP is connectionless
- IP routers are stateless

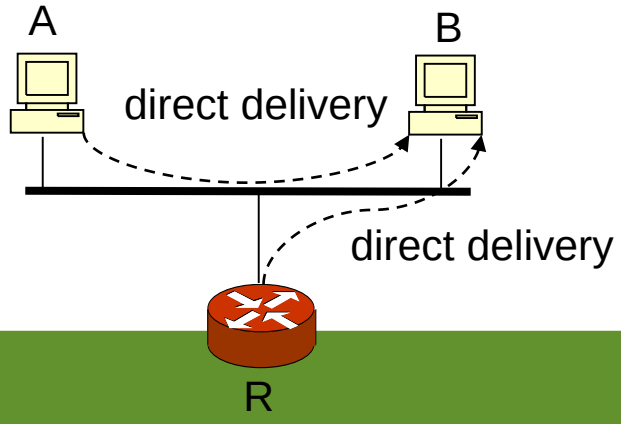
Direct and Indirect Delivery

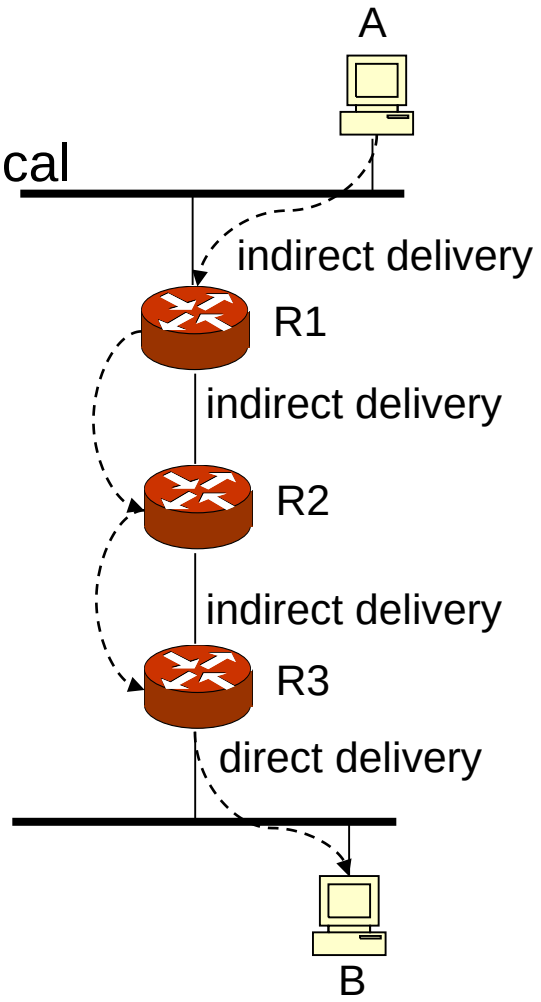
Direct delivery

- The final destination is connected to the same physical network as the sender.
- IP destination address and local interface has same netmask
- Map IP address to physical address: ARP

Indirect delivery

- From router to router, last delivery is direct
- Destination address and routing table: Routing



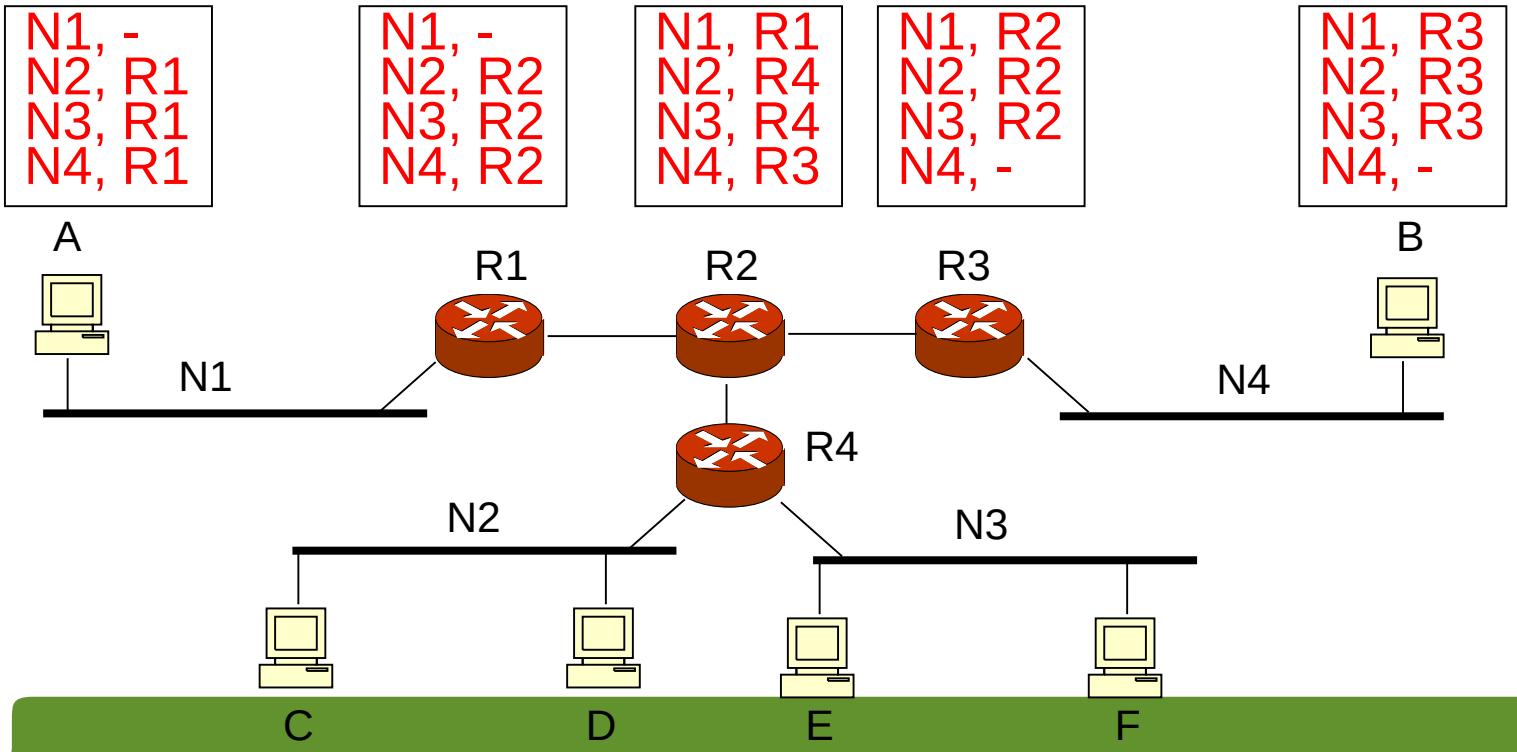


Next-hop Routing

How do you hold information about route from A to all other hosts?

- A → R1 → R2 → R3 → B

Table of *host/network address* and *next-hop* in every node



IP—RFC 791

Following the end2end argument, only the absolutely necessary functionality is in IP

- Best-effort service: unreliable and connectionless
- Application or Transport layer handles reliability

How to deliver datagrams over multiple links (hops) in an internetwork?

- Addressing
- Best-effort delivery service
  - Forwarding of packets from one link to another
- Error handling

IPv4 Header

Version
HLEN – Header Length
Type of Service
Total Length
- Header + Payload

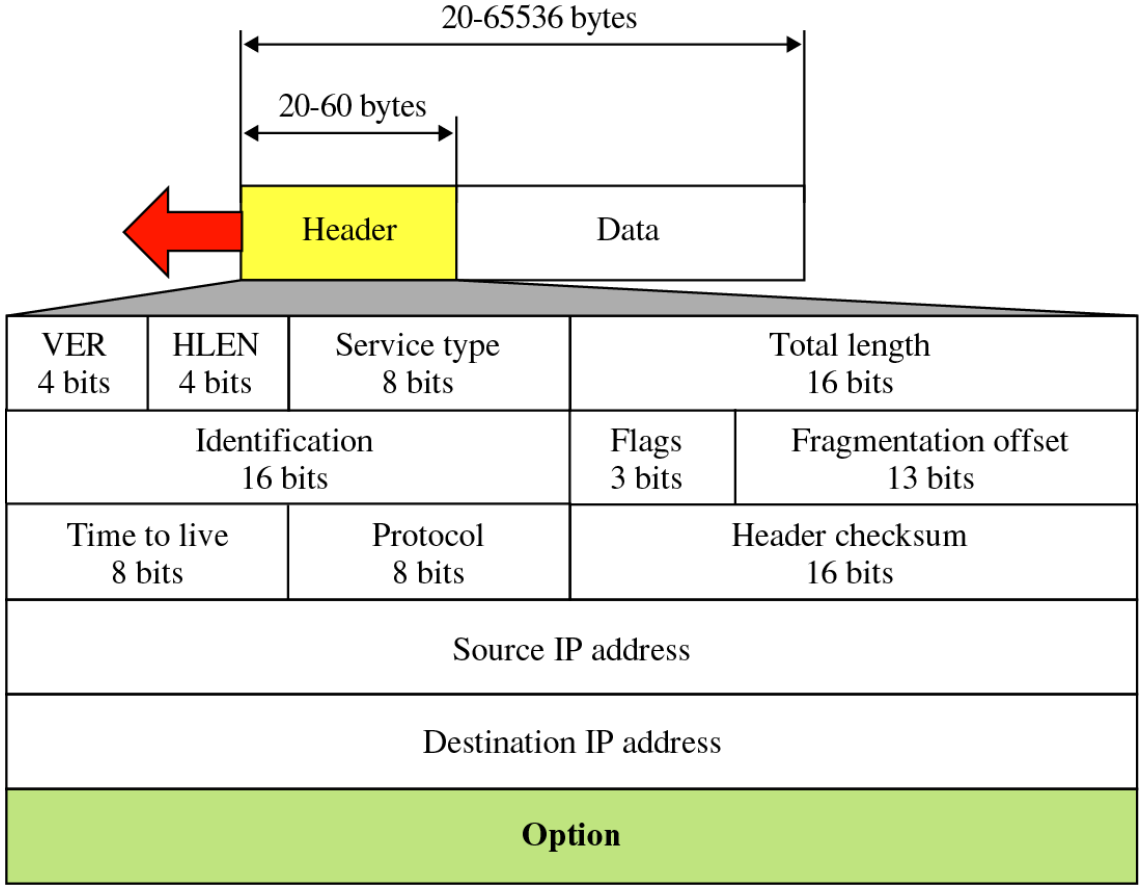
Fragmentation
- ID, Flags, Offset

TTL – Time To Live
- Limits lifetime

Protocol
- Higher level protocol

Header checksum
IP Addresses
- Source, Destination

Options



ICMP—RFC 792

ICMP—Internet Control Message Protocol

- Signalling protocol for IPv4, uses IP for transfers
- Report IP problems back to sender
- Control and management

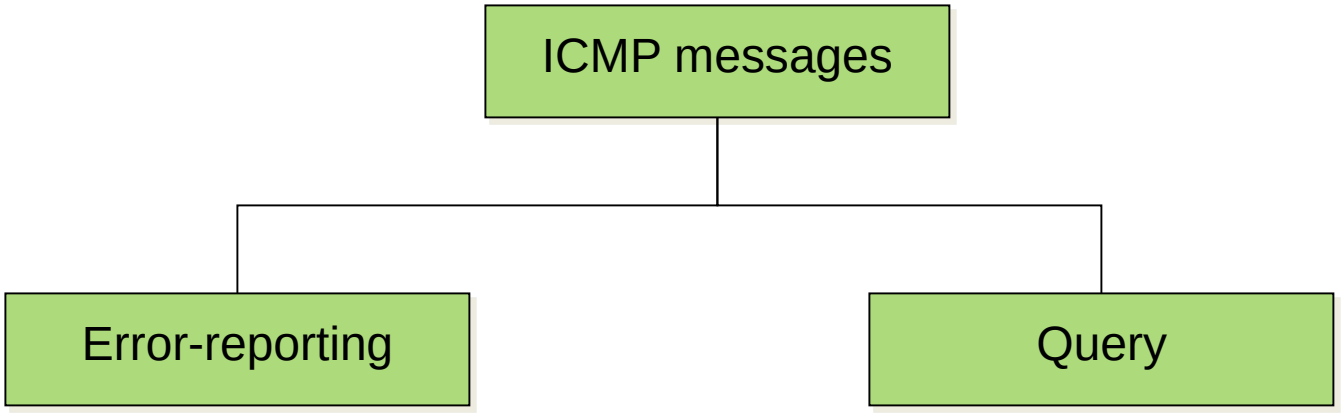
Query ICMPs

- Control purposes
- Examples: Echo, Router advertisement, Timestamp, etc.

Error ICMPs

- Sent when an error in IP detected
- Includes the first 8 bytes of the data field of the original datagram which caused the error.
- *Not* sent for: icmp errors, broadcasts, fragments, etc .
- Examples: Dest unreachable, Redirect, etc.

ICMP Messages



Type	Message
3	Destination unreachable
4	Source quench
11	Time exceeded
12	Parameter problem
5	Redirection

Type	Message
8/0	Echo request/reply
13/14	Timestamp request/reply
17/18	Address mask request/reply
10/9	Router solicitation/advertisement

Transport Layer

Responsible for end-to-end delivery of entire messages

Service-point addressing (Protocol Port or Port Number)

- Address the specific running process on a computer

Segmentation and Reassembly

- Divide message into transmittable segments and reassemble message at receiver

Connection Control

- For connection-oriented transport protocols

End-to-end Flow Control (not link level flow control)

End-to-end Error Control (not link level error control)

TCP/IP Transport Layer Protocols

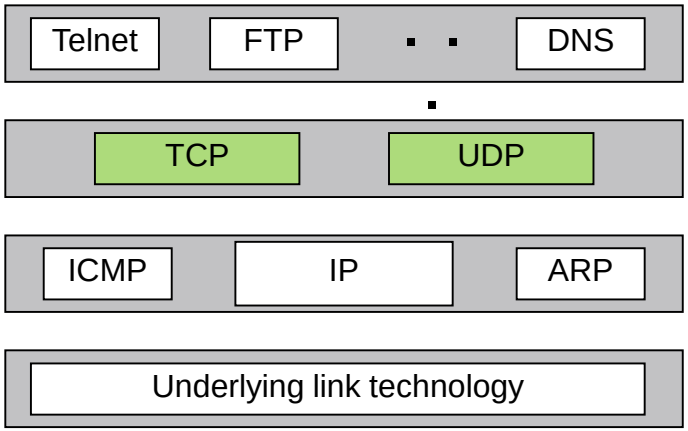
2 transport layer protocols in the TCP/IP stack

UDP – User Datagram Protocol

- Connectionless unreliable service

TCP – Transmission Control Protocol

- Connection-oriented reliable stream service



UDP—RFC 768

Datagram-oriented transport layer protocol

Provides connectionless unreliable service

Provides optional end-to-end checksum covering header and data

Provides no feedback to control data rate

An UDP datagram is silently discarded if checksum errors

UDP messages can be lost, duplicated, or arrive out of order

Application programs using UDP must deal with reliability problems

- DNS, DHCP, SNMP, NFS, VoIP, etc. use UDP
- An advantage of UDP is that it is a base to build your own protocols on

TCP—RFC 793

TCP is a connection-oriented transport protocol

TCP connection: full duplex connection between exactly two end-points

- Broadcast and multicast are not applicable to TCP

TCP provides a reliable byte stream service

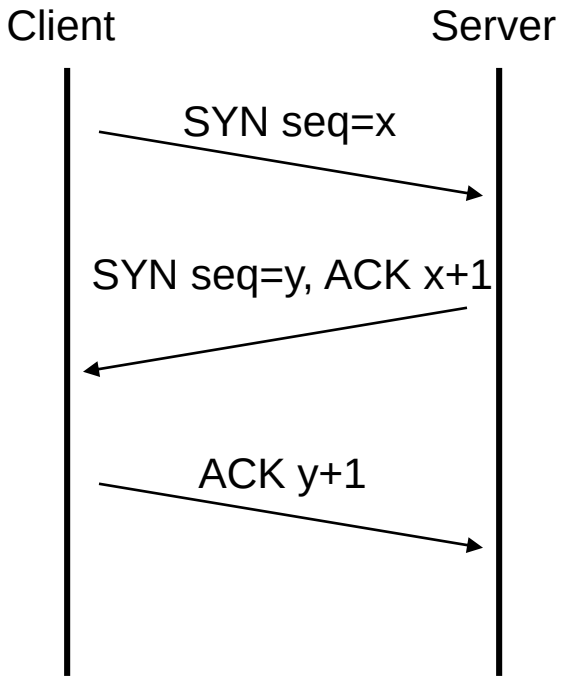
- A stream of 8-bit bytes is exchanged across the TCP connection
- No record markers inserted by TCP
- The receiving (reading) end cannot tell what sizes the individual writes were at the sending end
- TCP decides how much data to send (not the application); each unit is a **segment**

Lots of applications have been implemented on top of TCP

- TELNET (virtual terminal), FTP (file transfers), SMTP (email), HTTP

TCP Connection Establishment

TCP uses a three-way handshake to establish a connection



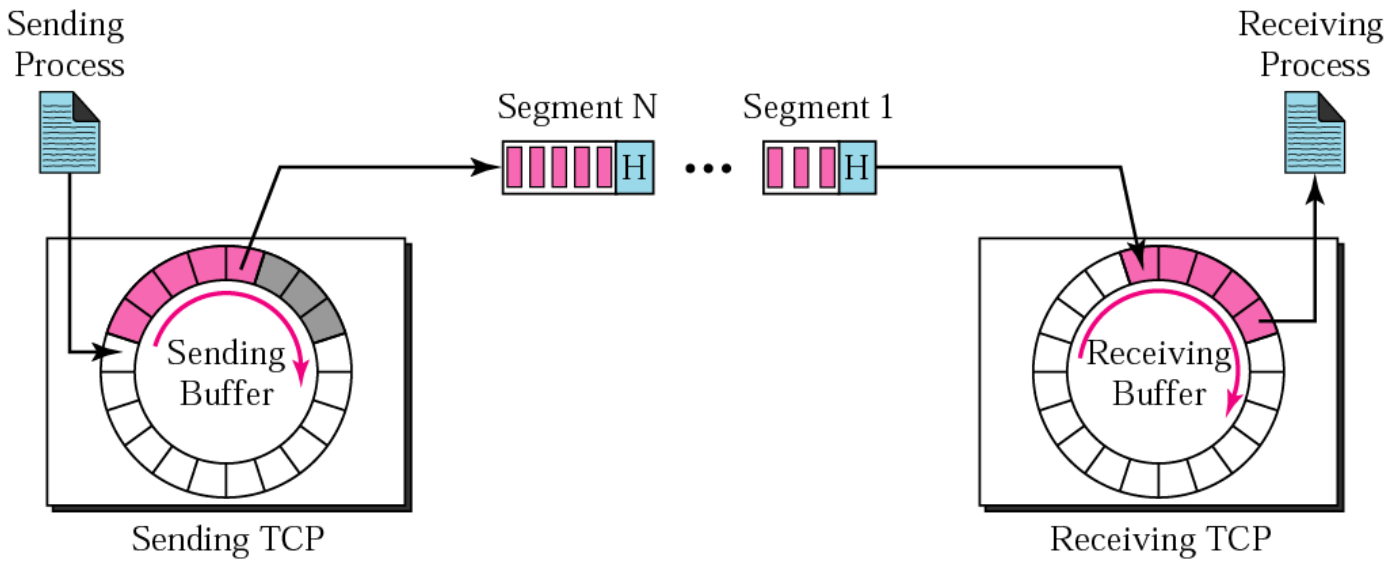
3-way handshake:

- Guarantees both sides ready to transfer data
- Allows both sides to agree on initial sequence numbers

Initial sequence number (ISN) must be chosen so that each incarnation of a specific TCP connection between two end-points has a different ISN.

Normally, client initiates the connection

Byte Streams, Buffers, and Segments



White area: empty locations, ready to be filled

Gray area: bytes that have been sent but not ACKed

Colored area:

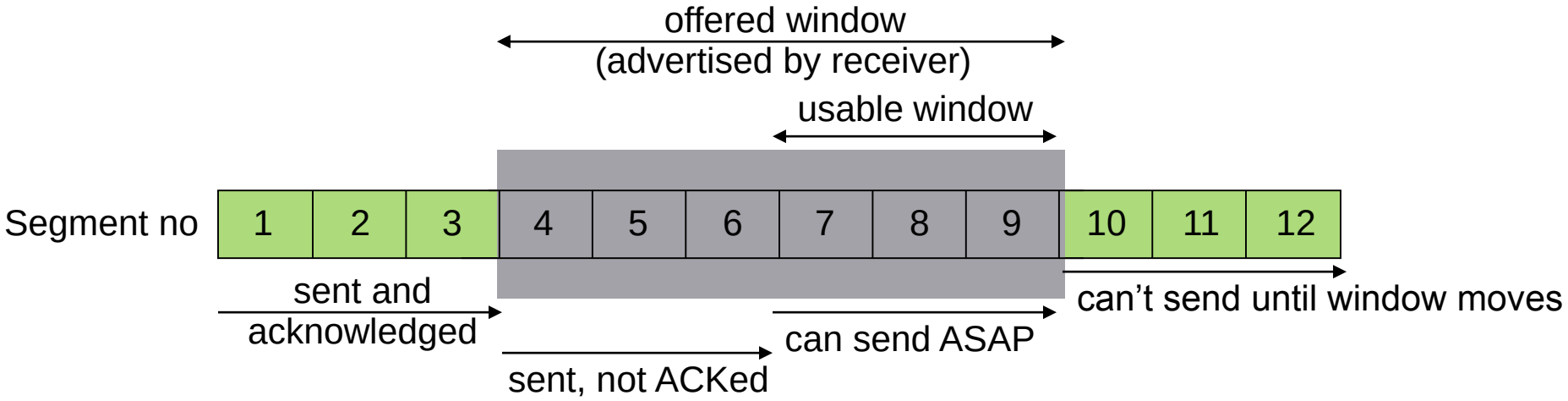
- bytes to be sent by the sending TCP
- bytes to be delivered by the receiving TCP

TCP Flow Control—Sliding Windows

Receiver: *offered window* – acknowledges data sent and what it is prepared to receive

- receiver can send an ACK, but with an offered window of 0
- later, the receiver sends a *window update* with a non-zero offered window size

Sender: *usable window* - how much data it is prepared to send immediately



Congestion

Previos section about flow control assumed the network could always deliver the data as fast as it was created by the sender

However,

Each router along the way has buffers, that stores the incoming packets before they are processed and forwarded

If packets are received faster than they can be processed, congestion might occur and packets could be dropped

Lost packet means lost ACK, and sender retransmits

Retransmission will add to congestion → network collapses

Thus, the *network* has to have a way to decrease window size

Congestion Control

Congestion Window

- Sender's window size is not only determined by the receiver, but also by the congestion in the network

Sender maintains 2 window sizes:

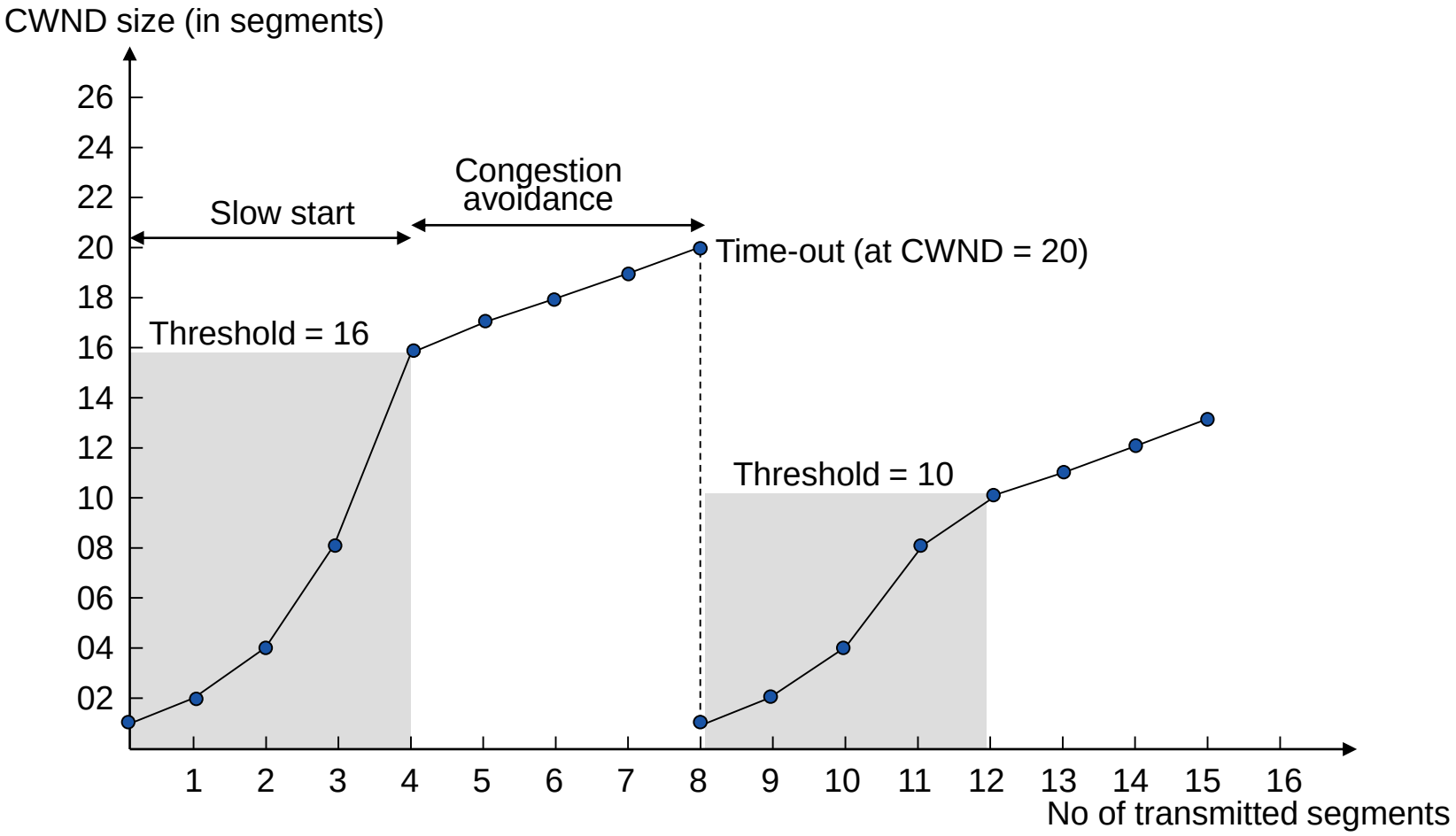
- Receiver-advertised window
- Congestion window (CWND)

Actual window size = min(rcv window, CWND)

To deal with congestion, sender uses several strategies:

- Slow start
- Additive increase of CWND
- Multiplicative decrease of CWND

Slow Start and Congestion Avoidance



Summary

Topics covered today should be familiar to you

If not, you have some catching up to do!

Good basic TCP/IP book:

- *TCP/IP Protocol Suite*, 2nd edition by Behrouz A. Forouzan, McGraw-Hill, ISBN 0-07-246060-1