Protocols

- **precise rules that govern communication between two parties**
- **TCP/IP: the basic Internet protocols**
- **IP: Internet protocol (bottom level)**
  - all packets shipped from network to network as IP packets
  - no guarantees on quality of service or reliability: "best effort"
  - each physical network has its own format for carrying IP packets
- **TCP: transmission control protocol**
  - creates a reliable 2-way data stream using IP
    errors are detected and corrected
  - most things we think of as "Internet" use TCP
- **"application-level" protocols, mostly built from TCP**
  - HTTP (web), SMTP (mail), SSH (secure login), FTP (file transfer), …
- **UDP: user datagram protocol**
  - simple unreliable datagram protocol (errors not detected)
  - used in DNS, remote file systems, ...

Packets

- **packet: a sequence of bytes carrying information**
  - usually over a network connection
- **bytes have a specific sequence, format, organization**
  - usually as specified in a protocol
- **typical network packet includes**
  - source (where it comes from)
  - destination (where it goes to)
  - size or length information (how big is the data part)
  - miscellaneous information (type, version, info to detect errors, ...)
  - the data itself ("payload")
- **typical sizes range from**
  - a few bytes
  - 150-1500 (Ethernet packets)
  - 100-65000 (IP packets)

What's in an IP packet

- **a "header" that contains**
  - protocol version, type of packet, length of header, length of data
  - fragmentation info in case it was broken into pieces
  - time to live: maximum number of hops before packet is discarded
    - each gateway decreases this by 1
  - source & destination addresses (32 bits for IPv4, 128 bits for IPv6)
  - checksum of header information
    - redundant info to detect errors in header information only, not data itself
  - etc.; about 20-40 bytes in header
- **actual data**
  - up to 64 KB of payload
  - IPv4:

version	type	hdr len	total len	frag	TTL	source address	dest address	chk	data...

IP: Internet Protocol

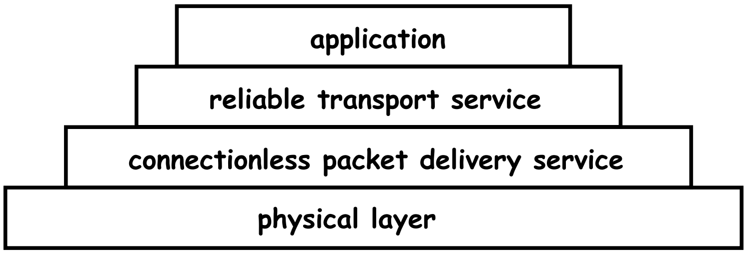
- **IP provides an unreliable connectionless packet delivery service**
  - every packet has full source & destination addresses
  - every packet is independent of all others
- **IP packets are *datagrams***
  - individually addressed packages, like postcards in the postal system
    - "connectionless"
  - stateless: no memory from one packet to next
    - each packet is independent of others, even if in sequence and going same place
  - unreliable: packets can be lost or duplicated ("best effort" delivery)
  - packets can be delivered out of order
  - contents can be wrong (though error rates are usually very low)
  - no speed control: packets can arrive too fast to be processed
  - limited size: long messages have to be split up and then reassembled
- **higher level protocols use IP packets to carry information**
- **IP packets are carried on a wide variety of physical media**

TCP: Transmission Control Protocol

- **a reliable 2-way byte stream built with IP**
- **a TCP connection is established to a specific host**
  - and a specific "port" at that host
- **each port provides a specific service**
  - SSH = 22, SMTP = 25, HTTP = 80, ...
- **a message is broken into 1 or more packets**
- **each TCP packet has a header (src, dest, etc) + data**
  - header includes checksum for error detection, and sequence number to preserve order and detect missing or duplicated packets
- **each TCP packet is wrapped in an IP packet and sent**
  - has to be positively acknowledged to ensure that it arrived safely
    - otherwise, re-send it after a time interval
- **TCP is the basis of most higher-level protocols**

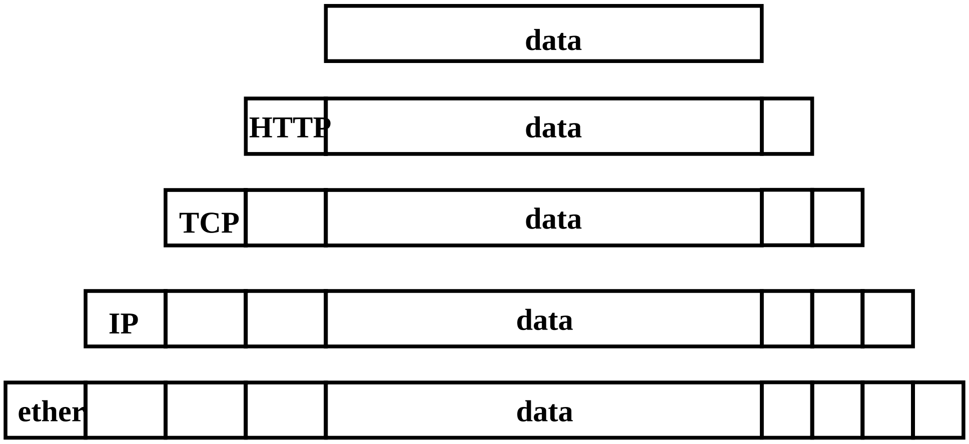
Higher level protocols

- **SSH: secure login**
- **SMTP: mail transfer**
- **HTTP: hypertext transfer -> Web**
- **protocol layering:**
  - a single protocol can't do everything
  - higher-level protocols build elaborate operations out of simpler ones
  - each layer uses only the services of the one directly below
  - and provides the services expected by the layer above
  - all communication is between peer levels: layer N destination receives exactly the object sent by layer N source


application
reliable transport service
connectionless packet delivery service
physical layer

Encapsulation

- **each piece of data at one level is wrapped up with a header and sent as a packet at the next lower level**
- **lowest level is what moves across specific network**

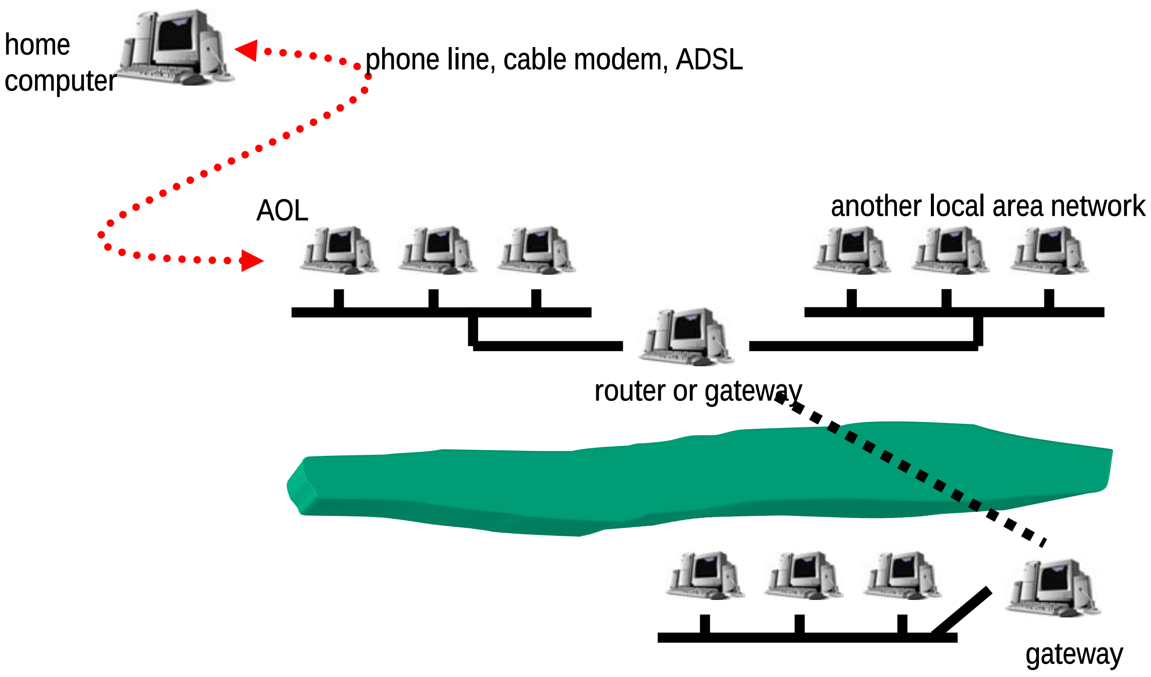


How are things connected?

- **local nets connected to local Internet Service Provider (ISP)**
- **these in turn connect to regional ISPs**
- **and then to larger ones like UUNet, AT&T, Sprint, …**
- **traffic exchanged at Internet exchanges**
  - large and small, formal and informal, profit and non-profit

- **bandwidth (bit-carrying capacity) of connections is usually higher the larger the ISP**
  - phone line analog modem 56 Kbps (you to your ISP)
  - cable modem, DSL 500 Kbps - 3MBps (you to your ISP)
  - telephone lines 1.5-45 Mbps (local ISP, big company to ISP)
  - optical fiber 155 Mbps and up (large carriers)

Typical home connection



Coping with bandwidth limits

- **data flows no faster than the slowest link**
- **limits to how much data can pass per unit time**
  - no guarantees about packet delivery
  - no bandwidth, delay or quality of service guarantees

    IP telephony is hard because voice traffic requires limited delay, jitter

    video is somewhat easier but needs a lot more bandwidth
- **caching**
  - save previous data so it doesn't have to be retrieved again
- **compression, encoding**
  - to improve use of available bandwidth
  - don't send redundant or unnecessary information

    text, code, etc., can be compressed and recreated exactly

    music, pictures, movies are compressed with some information discarded
- **home connectivity**

– telephone modem	56 Kbps
– ADSL, cable modem	1-4 Mbps
– wireless	1-50 Mbps
– fiber	50 Mbps?

Internet Ideas

- **packets versus circuits**
  - different models (mail vs phone)
- **names and addresses**
  - what is it called, how to find it
- **routing**
  - how to get from here to there
- **protocols and standards**
  - Internet works because of IP as common mechanism
    higher level protocols all use IP
    specific hardware technologies carry IP packets
- **layering**
  - divide system into layers
    each of which provides services to next higher level
    while calling on service of next lower level
  - a way to organize and control complexity, hide details

Internet technical issues:

- **privacy & security are hard**
  - data passes through shared unregulated dispersed media and sites
    scattered over the whole world
  - it's hard to control access & protect information along the way
  - many network technologies (e.g., Ethernet, wireless) use broadcast
    encryption necessary to maintain privacy
  - many mechanisms are not robust against intentional misuse
  - it's easy to lie about who you are
- **service guarantees are hard**
  - no assurance of reliable delivery, let alone of bandwidth, delay or jitter
- **some resources are running low**
  - especially IPv4 addresses
  - IPv6 (the next generation) uses 128-bit addresses
    acceptance has been slow but is growing

- **but it has handled exponential growth amazingly well**