Internet Architecture

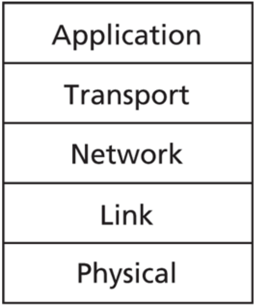
Daniel Zappala

CS 460 Computer Networking
Brigham Young University

How do you Build an Internet?

- Everyone in the world should be able to communicate using any application they want

Internet Architecture

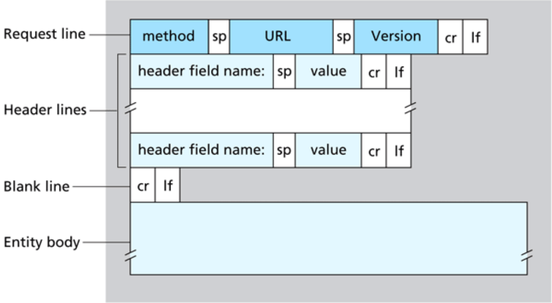


- an architectural model that separates communication protocols into layers
- layering helps to build complex systems
  - split large system into smaller pieces
  - identify each layer's functionality and interfaces
  - can change a layer's implementation as long as interfaces remain the same

Protocols

- a formal definition of how two or more entities communicate
- includes
  - **syntax**: format of messages
  - **semantics**: actions taken when a message is sent or received
  - **events**: actions taken when an event occurs

Protocol Example

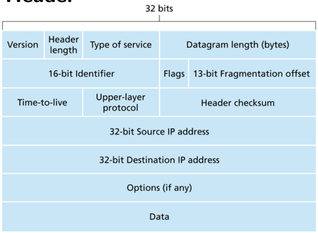


- HTTP Request message format
  - sent in ASCII format
  - *request line*: method, URL, version
  - *header lines*: additional method parameters
  - ends with a carriage return and line feed
- actions: what happens when a server gets a request?

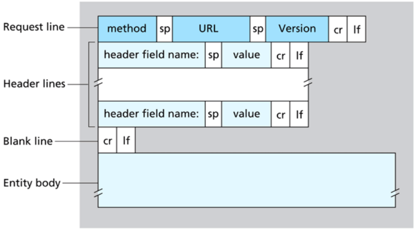
Binary versus ASCII Protocols

- link, network and transport layer protocols exchange messages coded in binary
  - conserve space for small packets or expensive bandwidth
  - requires standardizing a byte-level format
  - must be careful about transmitting numeric values in network byte order
- application-layer exchange messages coded in ASCII
  - large messages, cheap bandwidth
  - easier to write, debug, extend

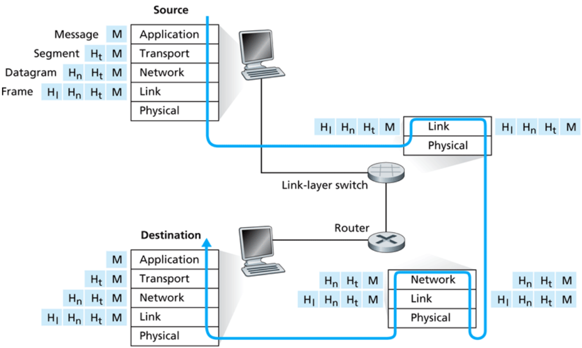
IP Header



HTTP Header



Encapsulation and Decapsulation



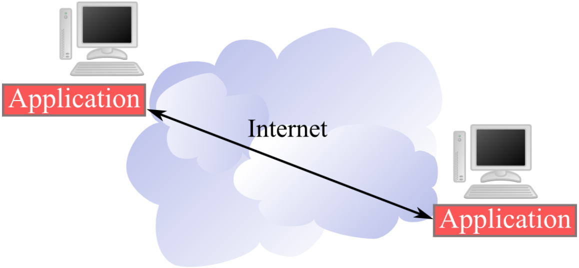
- When sending down the stack, each layer appends a header to the data it receives
- Intermediate computers may process only one or two layers
- When sending up the stack, each layer removes its header

Theory Versus Practice

- an architectural model helps to define the functionality of each component and the interfaces between components
- a particular implementation is free to combine layers or create new layers to create a more efficient or flexible system

The Layers

Application Layer

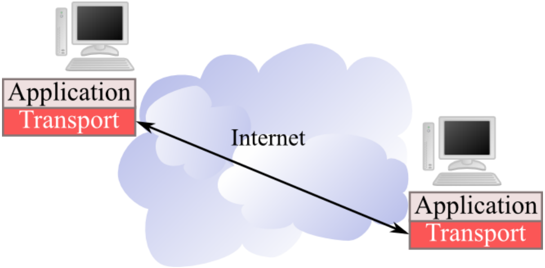


- includes all applications
- treats the Internet as a service that provides a virtual, reliable link between two computers

Application Layer Services

- query-response: basic services
  - DHCP
  - DNS
- client-server communication: a server provides a service to clients
  - HTTP: the World Wide Web
  - SMTP: e-mail
- peer-to-peer communication: host collaborate to share content, acting as both clients and servers
  - Gnutella (and variants): file searching and sharing among peers
  - BitTorrent: file distribution from a well-known source
  - Coral: peer-to-peer web caching
- cloud computing: client-server with a distributed system
  - computing: Amazon Web Services, Google App Engine
  - file sharing: Dropbox, Google Drive

Transport Layer



- delivers data between hosts on the Internet
- treats the Internet as a service that provides a virtual, but unreliable link between two computers

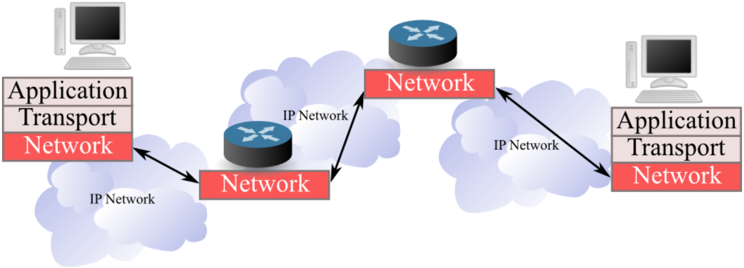
Transport Layer Services: TCP

- connection-oriented: requires state setup at sender and receiver
- provides a reliable, ordered byte stream
  - reliable: retransmits any segments that are lost
  - ordered: buffers and re-orders segments before delivery to application
  - byte stream: transfers bytes, not messages
- provides flow control: avoid overflowing the receiver's buffer
- provides congestion control: avoid persistently overflowing network buffers
- applications: web, file transfer, remote login, email

Transport Layer Services: UDP

- connectionless: no state setup
- unreliable: lost packets are not re-sent
- no flow control
- no congestion control
- applications: query-response (DNS, DHCP), streaming media (voice, video), some peer-to-peer protocols

Network Layer



- forwards packets between computers and routers on the Internet

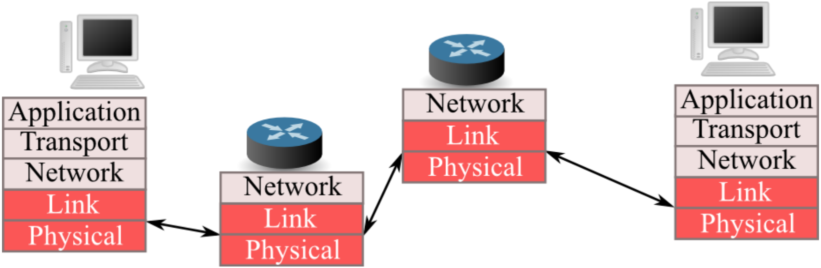
Network Layer Services: IP

- common protocol needed to interoperate with other computers on the Internet
- implements a best-effort service model - routers make their best effort to deliver all packets, but packets may be
  - delayed (long queues in the network)
  - dropped (queue overflow)
  - duplicated (mistaken retransmission by TCP)
  - re-ordered (packets may take different paths)
- reliability and ordering are the responsibility of TCP

Network Layer Services: Routing

- routing protocols decide which path to use when sending packets to a given destination
  - organized hierarchically: BGP in the backbone, anything you want (OSPF, IGRP, RIP) in your own network
  - create and manage a routing table with potentially many paths to each destination
- choose one path for each destination at any point and create a forwarding table with these paths
- routers use the forwarding table to choose an outgoing link for each packet

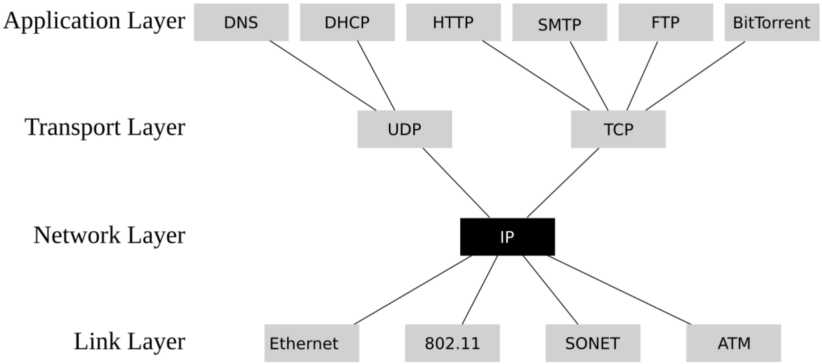
Link and Physical Layers



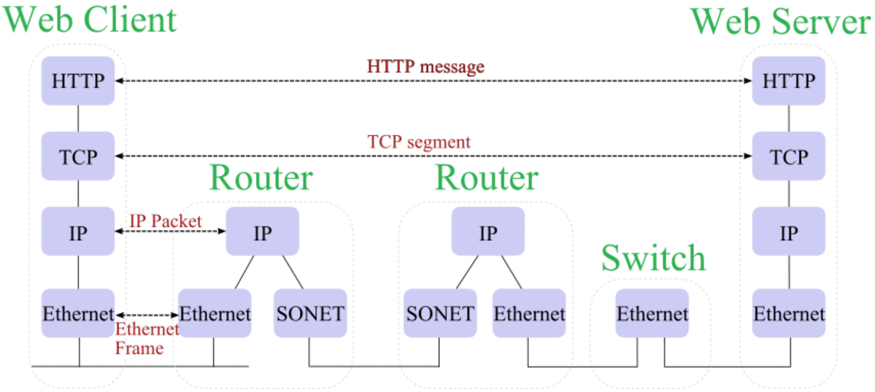
- link layer: sends a frame on one link
- physical layer: sends bits on one link

Putting it Together

The Internet Hourglass



The Internet at each Hop



Standardization

- standards are essential to interoperability on the Internet
- Internet Engineering Task Force *www.ietf.org*
  - standardizes Internet protocols: IP, TCP, HTTP, etc
  - open to all to participate, free of charge
  - relies on working code and rough consensus
- W3C *www.w3c.org*
  - standardizes web protocols and formats
  - industry-oriented consortium
  - requires approved and paid membership ($6,350 - $63,500 per year)
  - many standards do not require Internet-wide deployment

Security

Why is the Internet so Vulnerable to Attacks?

- The Design Philosophy of the DARPA Internet Protocols, Clark, 1988
- fundamental goal
  - develop an internetwork for existing networks
- second-level goals
  1. survivability
  2. multiple types of service (delay vs bandwidth, reliable vs datagram)
  3. variety of networks
  4. distributed management
  5. cost effective
  6. host attachment with low effort
  7. accountable resources

Security Was Not Considered

Cat And Mouse

- security is a bandaid for the Internet
- constant game of cat and mouse
- numerous attacks
  - malware (e.g. viruses, worms) to create botnets
  - denial-of-service attacks (DoS, DDoS)
    - exploit vulnerabilities
    - bandwidth flooding
    - connection flooding
  - packet sniffing
  - IP spoofing
  - ...and many more

History

1961 - 1972: Early Packet-Switching Principles

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency
- 1969: first ARPAnet node operational
- 1972:
  - ARPAnet demonstrated publicly
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes

1972-1980: Internetworking, New and Proprietary Networks

- 1970: ALOHAnet satellite network in Hawaii
- 1973: Metcalfe's PhD thesis proposes Ethernet
- 1974: Cerf and Kahn - architecture for interconnecting networks define today's Internet architecture
    - minimalism, autonomy - no internal changes required to interconnect networks
    - best effort service model
    - stateless routers
    - decentralized control
- late70's: proprietary architectures: DECnet, SNA, XNA
- late 70's: switching fixed length packets (ATM precursor)
- 1979: ARPAnet has 200 nodes

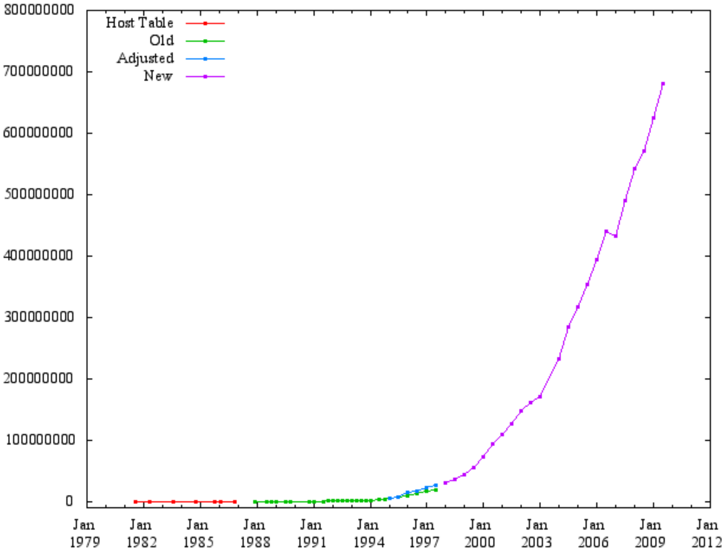
1980-1990: New Protocols, a Proliferation of Networks

- 1983: deployment of TCP/IP
- 1982: SMTP e-mail protocol defined
- 1983: DNS defined for name-to-IP-address translation
- 1985: FTP protocol defined
- 1988: TCP congestion control
- new national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks

1990, 2000's: Commercialization, the Web, New Apps

- early 1990's: ARPAnet decommissioned
- 1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)
- early 1990s: Web
  - hypertext [Bush 1945, Nelson 1960's]
  - HTML, HTTP: Berners-Lee
  - 1994: Mosaic, later Netscape
- late 1990's: commercialization of the Web
- late 1990's - 2000's:
  - more killer apps: instant messaging, P2P file sharing
  - network security to forefront
  - backbone links running at Gbps

Internet Growth (1981-2006)



Why Did the Internet/IP Win?

- relied on rough consensus and working code
    - implementations available
    - design influenced by experience: performance
    - fluid and open standardization body (IETF)
- open (rather than proprietary) architecture
- timing: need research, then standards, then lots of money invested
- the right technology: best-effort service model, common building block, with reliability in transport layer