TCP/IP Reference Model

Overview: history and standards

Host-to-network layer

Internet layer

Transport layer

Application layer

Internet governance

Overview

- Formal framework with similar goals and functionality to OSI
- In the TCP/IP world, the protocols and their implementations were well-developed and used before consideration of a formal reference model
  - opposite approach to OSI: ground-up vs. top-down
  - the TCP/IP reference model is sometimes referred to as the “TCP/IP protocol suite”

Overview, 2

- There are differing opinions about whether TCP/IP should be considered four or five layers, and what their names should be:
  - Kurose & Ross: application, transport, network, link (LANs & MAC (Media Access Control))
    - they use “link” in the chapter of the title but “data-link” everywhere else
  - Tanenbaum: application, transport, internet, host-to-network (MAC is a “sublayer”)
  - others: application, transport, internet, network, physical
- References:
  - Tanenbaum: vs. 1.4.2, 1.4.5, 1.5.2–1.5.4, 1.7.3, 3.6.2, 5.5, 6.1, 6.4, 7.2, 7.4, 7.5
  - Kurose & Ross: Ch 1.9, 5.8, 4.4, 4.5, 4.7, 3.2, 3.3, 3.5, 2.1, 2.3–2.6

History of TCP/IP and the Internet

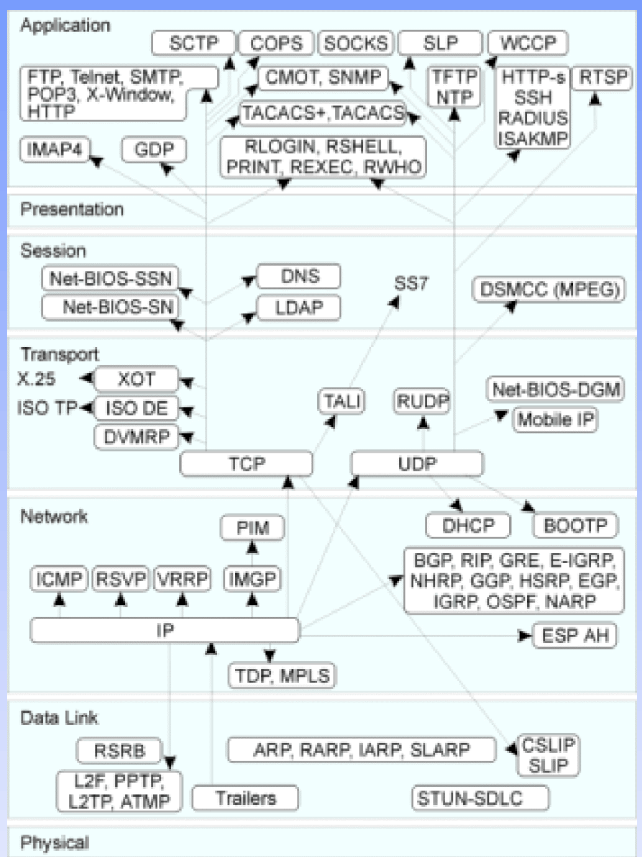
- Mid-1960's with DARPA, 15 nodes by 1972, TCP/IP not used (NCP instead)
- ARPANet grows to include defense contractors and universities with military research, 200 by 1979
- Early 1980s: CSNet, BITNet for non-military; informal internetworking with ARPANet
- 1983: TCP/IP becomes standard internetworking protocol on ARPANet
- 1986 NSFNet provides backbone for internetworking; mostly US, some Canadian
- Late 1980s, "Internet" term emerges
- 1990 ARPANet disbanded (leaving MILNet for military only); NSFNet for all other
- 1991 NSFNet allows commercial traffic; Version 1 of Web protocols (HTML, HTTP)
- 1995 NSFNet disbanded, backbone services provided commercially

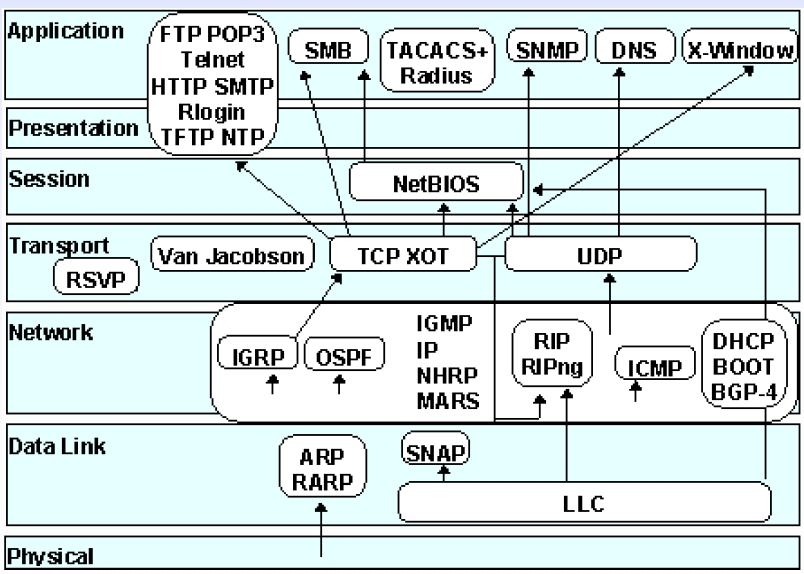
Standardization mechanisms

- TCP/IP standards historically were developed informally:
  - graduate students & academics
    - e.g. Metcalfe 1973: Ethernet
  - US military contractors
- Adoption of standards occurred by consensus of the (small) user community
- Software developed by research programs, made freely available (typically via Unix)
- To create a standard:
  - write an RFC (Request For Comments) describing the protocol
  - write software to implement the protocol
  - make the RFC and software freely available
- More on current governance practices later

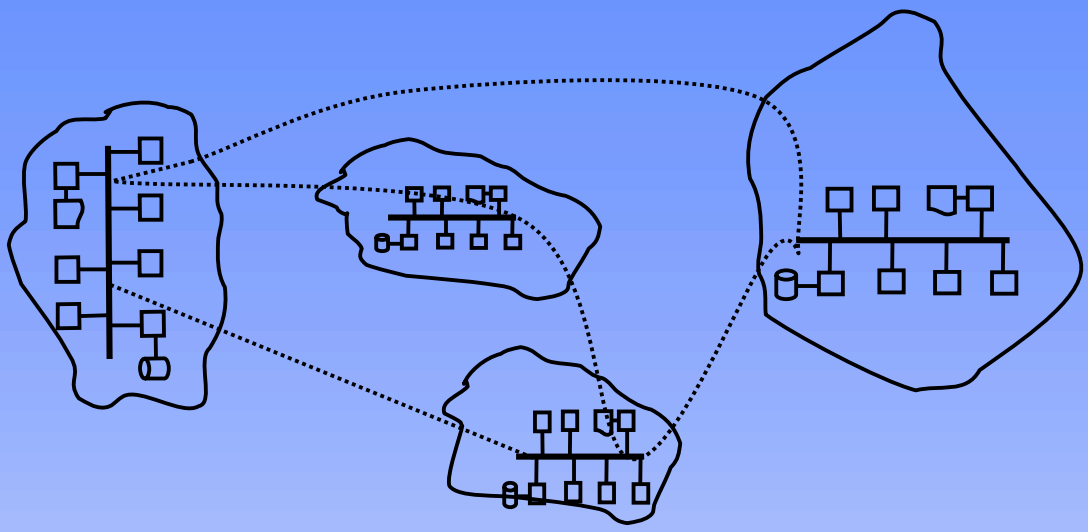
TCP/IP vs. OSI

- From protocols.com http://www.protocols.com/pbook/tcpip.htm





[Link | Host-to-network] layer



- TCP/IP is an internetworking framework concerned with connecting networks to networks
- Lowest layer is host-to-network or link
  - includes OSI physical, data link
  - deals with how a host is connected to its local environment
- TCP/IP doesn't have much to say host-to-network:
  - commonly use IEEE 802.x LAN standards
    - called the *Medium Access Control Sublayer* (MAC sublayer)

Host-to-network: protocols

- Within a LAN, host-to-network protocols are borrowed, but host-to-network layer also contains point-to-point protocols used to connect TCP/IP systems via a WAN:
  - SLIP (serial line IP)
  - PPP (point-to-point protocol)
  - predominant protocols for “dial-up internet” services
    - the host that is dialing up is equivalent to a very small network that wants to internetwork with other networks
  - also used for router-to-router WAN connections
- These protocols operate at a level equivalent to OSI data link

Host-to-network: SLIP

- SLIP: Serial Line IP
- Unsophisticated protocol:
  - sends raw IP packets over communications channel
  - end-of-frame marked by a single flag byte
  - character stuffing used to escape flags within data stream
- Problems:
  - no services like error checking, flow
  - each end of the wire must have a hard-wired permanent IP address (no dynamic assignment)
  - can only be used for IP (e.g. Novell, Microsoft protocols not possible)
- C-SLIP: Compressed SLIP
  - looks at IP, TCP PDU headers and compresses where possible
  - can be significant saving in overhead

Host-to-network: PPP

- PPP: Point-to-Point Protocol
- Designed to solve deficiencies in SLIP
- Does all the things a data link layer should:
  - error detection (CRC)
  - framing: HDLC-based, character-stuffed
  - multiple protocols
  - dynamic address assignment
  - authentication
- Related protocols:
  - LCP: Link Control Protocol for line connection and release, option negotiation
  - NCP: Network Control Protocol for negotiating network layer options (e.g. IP address assignment)
    - one NCP for each kind of network layer

[Network | Internet] layer

- Analogous to OSI network layer – purpose is to route packet through the interconnected network(s)
- Dominant protocol is IP – Internet Protocol; designed from the beginning for internetworking
- IP is an unreliable, connectionless packet-delivery service
  - IP's main purpose is to handle routing and related issues
  - data-stream is received from the transport layer, chopped into packets
  - datagrams are delivered, possibly in the wrong order; transport layer must rearrange
  - current version is IPv4; defined by RFC 791 (September 1981)

Internet layer: protocols

- Internet layer also contains *control protocols*, used to monitor operation and exchange information:
  - ICMP: Internet Control Message Protocol
  - BOOTP: BOOTstrap Protocol
  - DHCP: Dynamic Host Configuration Protocol
- Example ICMP functions:
  - echo, timestamp, information request
  - echo, timestamp, information reply
  - destination unreachable
  - etc.

Internet layer: IP addresses

- IP defines the addresses used in internetworking (hosts have to know how to identify the destination!)
- Anyone can set up **an** internet addressing universe; **the I**nternet happens to be one such universe
  - **I**nternet (as opposed to **i**nternet) addresses are assigned by a central authority
- Recall that IP is designed to facilitate internetworking i.e. the connection of independent networks
- IP addresses are designed to accommodate this:
  - 32-bit integer address; two parts:
    - network address: identifies the network
    - host address: identifies a host on the given network

Internet layer: IP addresses-2

- Three classes of addresses:
  - class A:

    - 7 bits for network address
    - 24 bits for host address
  - class B:

    - 14 bits for network address
    - 16 bits for host address
  - class C:

    - 21 bits for network address
    - 8 bits for host address
- Each address class has a finite number of networks & hosts

Internet layer: IP addresses-3

- "dotted decimal" notation:
  - divide the 32 bits into 4 8-bit groups, use a number 0..255 for each group
  - Example:
    `10000001 01100001 11010000 00010011`
    is 129.97.208.19
- The network addresses do not necessarily align with the "dotted-decimal" notation
- Some magic addresses:
  - 127.x.x.x (i.e. class A network number 127) is the *loopback address* (self-reference)
  - address all 1s refers to all hosts on the local network (broadcast)
  - host address portion all 1s is a broadcast on the indicate network
  - address all 0s refers to the local host
  - network address 0 refers to a host on the local network

Internet layer: IP addresses-4

- IP addressing reserves certain ranges of addresses:
  - leading bits 1110 (224.0.0.0 to 239.255.255.255) is reserved for multicasting (sending from one host to many hosts in one operation)
  - leading bits 1111 (240.0.0.0 to 255.255.255.255) is reserved
  - IP defines “private use” address ranges for “private internets”
    - “Address allocation for private internets” is described in RFC 1918

Internet layer: IP addresses-5

- Three address ranges are reserved:
  - Class A: 10.0.0.0 to 10.255.255.255
  - Class B: 172.16.0.0 to 172.31.255.255
  - Class C: 192.168.0.0 to 192.168.255.255
- These addresses must not be propagated or routed by any inter-system router
  - thus, addresses can be reused by many organizations without interference
  - current practice is to allocate a few public Internet addresses to routers or gateways within an organization, and then use private internet addresses internally
    - gateways or routers must do *network address translation* (NAT) for interior hosts
    - interior hosts are **not** directly part of the **I**nternet (cannot be reached from outside)
- NAT is the basis for most SOHO “internet sharing” devices on the market
  - defined in RFC 1631

Internet layer: IP addresses-6

- Hosts can have more than one network interface, each of which has its own IP address:
  - called “multi-homed” host
  - technically a router (in the generic sense), since it has to decide which interface to use to send data
- An internet address actually specifies the connection (the network interface hardware), not the host

Internet layer: IP addresses, subnetting

- Organizations may wish to subdivide their network into smaller physical regions for administrative purposes
  - sale & marketing vs. R&D vs. corporate
  - the organization has one class C address range, but several physical networks
  - networks could be bridged, but performance may suffer and organizational autonomy is lost
- Solution: divide the class C range into smaller *subnets*
  - to the outside world there is no difference: the class C address is still in the same place with the same set of hosts
  - internally, networks are physically partitioned according to the subnet ranges
  - RFC 950 describes subnetting
- NB this use of the term “subnet” conflicts with the generic “network cloud” use

Internet layer: IP addresses, subnetting-2

- Example:
  - consider class C network 192.168.1, with host addresses 192.168.1.0 to 192.168.1.255 (256 hosts; ignoring special values)
  - the binary representation consists of 24 bits of network part (3 class prefix and 21 actual network number) and 8 host bits
    - a *bit-mask* (or *network mask*) for this is: `11111111 11111111 11111111 00000000`
      - dotted-decimal: 255.255.255.0
    - each 1 bit represents part of the network specification, a 0 bit is part of the host number

Internet layer: IP addresses, subnetting-3

- Example, continued:
  - suppose we want to partition this network into four subnets of ~64 hosts each
    - construct a new mask:
      `11111111 11111111 11111111 11000000`
      - dotted decimal: 255.255.255.192
    - only 6 bits are available for host numbers, and two additional bits are considered part of the network specification
  - this arrangement gives $2^6 = 64$ hosts and $2^2 = 4$ different networks within the single public class C network
  - internally the four physical networks are numbered 0, 1, 2 and 3, each with hosts numbered 0 to 63; externally nothing has changed
  - each of the four networks would need a router to connect it to the rest of the world
    - this could be a single router with five interfaces (one external, four internal)

Internet layer: IP addresses, subnetting-4

- Example, continued:
  - the new internal network and host numbers cannot be represented with the dotted decimal notation, so the external numbering scheme is used for convenience
    - the subnetting is not apparent unless the subnet mask is known

Internet layer: IP addresses, problems

- Internet addressing has problems:
  - moving a host from one network to another changes its address – headache for laptops
    - dynamic IP address assignment (DHCP)
  - class C networks have only ~256 hosts; exceeding threshold is administrative nightmare
  - routing is address-based, which is restrictive for multi-homed hosts
  - there are not enough addresses
    - class A and B are gone, C is going fast
- Address-space solutions:
  - CIDR: Classless InterDomain Routing
    - allocating blocks of class C addresses
    - geographic zones
  - IPv6: 16-byte IP addresses
    - not compatible with IPv4
    - compatible with existing protocols

Internet layer: IP addresses, CIDR

- CIDR: Classless InterDomain Routing
  - instead of predefining classes of addresses, use a flat, classless address space
  - use a suffix "/*n*" to specify the number of bits in the network part of the address
    - e.g. a.b.c.d/22 denotes a network with 10 bits of host addressibility (32-22 = 10; 2^{10} = 1024 hosts)
    - these 10 bits can be subnetted further
  - this is equivalent to subnetting the enclosing network (a class B in this example) with a mask of 255.255.252.0
  - most ISPs are given big CIDR blocks, and subnet them to their customers in smaller blocks
    - e.g. an ISP with a /20 CDR block might issue /26 blocks of 64 hosts each

Internet layer: routing

- Routing: how to get data from one host to another
- We are describing IP routing, which assumes that the sender knows the IP address of the destination
  - to translate a name to an address uses DNS – discussion following
  - in LANs, if the destination is in the same network as the sender, use the Datalink address instead of the IP address
    - determine destination network by extracting the network number from the IP address and comparing to sender's
    - if the same network, use ARP (Address Resolution Protocol) to obtain DL address – discussion following

Internet layer: routing-2

- Given a destination IP address, routing involves two separate phases:
  - intra-AS (within an organization): interior gateway protocols
    - does not require complete topology information, uses cost measurements (e.g. hop counts)
    - RIP & RIP2 (Routing Information Protocol) were the original routing protocols that implemented straightforward distance-vector routing algorithms
    - OSPF (Open Shortest Path First) was evolved to handle more complex requirements, including “area” organization
      - developed as open, public domain std
      - much more sophisticated than RIP
    - EIGRP (Enhanced Internal Gateway Routing Protocol) is a proprietary algorithm developed by Cisco as a replacement for RIP

Internet layer: routing-3

- inter-AS (between organizations): exterior gateway protocol
  - BGP (Border Gateway Protocol)
    - assumes complete knowledge about network topology; link-state algorithms
      - BGP tables for worldwide AS use must be created & maintained
    - BGP uses “path” information rather than technical cost information
    - BGP administrators establish policies about which paths are acceptable for use
- In all cases the basic goals are the same, but technical or administrative constraints change the rules

Internet layer: routing-4

- Conceptual model for routing
  - every host is a router or not
    - if it is not, it knows the address of a router
    - if it is, it has a routing table with (destination IP, next-hop router IP, data link address) triples
    - the destination IP could be a host address or a network address
  - when a datagram arrives at a host
    - if the datagram is for this host, it is “passed up” (to transport layer)
    - otherwise:
      - if this host is not a router, silently discard the datagram
      - if this host is a router, forward (route) the datagram elsewhere

Internet layer: routing-5

- Forwarding (routing) a datagram:
  - search the routing table for an exact match of destination IP
  - if found, send the datagram to the indicated next-hop router via the data link address
  - if not found in the routing table, extract the network address and repeat
  - if still not found, look for a “default” route
  - if still not found, give up and return “host/network unreachable”
- Observation: in general, IP never knows the entire route, only the next-hop address
  - recording a route is possible
  - specifying a complete route is possible

Transport layer

- Virtually identical to OSI transport layer: provides host-to-host communication, independent of subnet
  - (subnet in the generic sense)
- Two formal protocols:
  - TCP: Transmission Control Protocol – reliable connection-oriented
  - UDP: User Datagram Protocol – unreliable datagram

Transport layer: UDP

- UDP is simple encapsulation of IP datagrams
  - unreliable, connectionless service
  - no guarantee of order
  - designed to be as minimalist as possible
    - minimal PDU overhead
    - no congestion, flow control, so applications can send data as fast as they can
- Useful in:
  - real-time applications
  - client–server query–response situations
  - any situation where a lost packet is not crucial
    - applications have the choice to implement the reliability themselves if necessary

Transport layer: TCP

- TCP performs all standard functions of reliable connection-oriented service, including:
  - data-stream fragmentation into segments, which are then packaged into IP datagrams
  - segment and datagram sequencing and re-assembly
  - data integrity (checksumming)
  - flow control (sliding window)
  - connection management (three-way handshakes)
  - symmetric close

Transport layer: TCP-2

- Special features of TCP:
  - byte stream, not message stream
  - data is usually buffered, can be bypassed (PUSH; urgent)
  - services accessed via a *socket*: an IP address and port number (16-bit number)
    - sockets are modelled on the Unix concept of a file
  - connections are full-duplex (both directions simultaneously) and point-to-point (no broadcasting)
  - does congestion control (tries to reduce throughput demands)

Transport layer: TCP-3

- TCP & UDP port numbers <1024 are "well known" (reserved), others are "registered" and "private"
- Well-known ports formerly defined by RFC process (latest was RFC 1700), now managed online at:
  http://www.iana.org/numbers.htm
  - applications are not supposed to use well-known or reserved ports, but there is no effective enforcement possible
- Some well-known TCP ports:
  - FTP: 21 (20 for data-stream)
  - SMTP: 25
  - POP: 110
  - IMAP: 143
  - NNTP: 119
  - Telnet: 23
  - WWW: 80

Transport layer: TCP-4

- Applications access TCP services via *socket library* or *socket API*:
  - socket: create a socket
  - bind: connect to a socket
  - listen: prepare for incoming connection
  - accept: wait for incoming connection
  - connect: try to establish a connection
  - send: send data over connection
  - receive: receive data from connection
  - close: close a connection
- Client-server roles:
  - bind, listen, accept: server-side
  - connect: client-side
  - socket, send, receive, close: both sides
- For example: at system-startup, SMTP server binds itself to port 25, waits for incoming connections

Application layer

- TCP/IP application layer combines function of OSI session, presentation and application layers
- Familiar user applications:
  - e-mail: SMTP (simple mail transfer protocol) and POP (post office protocol)
  - remote login: rlogin, telnet
  - file transfer: FTP (file transfer protocol)
  - WWW: HTTP & HTML
- DNS (domain name system) is an application that receives special attention:
  - converts host names to IP addresses and vice versa
  - so ubiquitous, often mistaken for a TCP or IP standard function

Applications: DNS

- IP numbers are annoying, symbolic names are preferable
- DNS is a distributed database client–server application that provides mapping service between IP numbers and host names
  - all hosts can have a name
  - a hierarchical namespace, with hierarchy segments separated by dots (tree model)
  - hierarchy reads right-to-left
  - e.g. csg.uwaterloo.ca
    - ca is the “top-level domain”
    - uwaterloo and csg are subdomains
  - strictly, the term “domain” applies to any proper suffix: ca, uwaterloo.ca, csg.uwaterloo.ca
  - it is not possible to know what a name is (host vs. subdomain) just by looking at it

Applications: DNS-2

- DNS name authority:
  - authority for each level in the hierarchy is derived from the “parent” authority
  - each sovereign nation controls its national domain namespace (defined by ISO two-character country code: .ca, .uk, .fr, etc.)
    - called ccTLDs (TLD = “top level domain”)
  - authority for the so-called “generic top-level domains” (.com, .net, .org, etc.) is granted by a non-profit international organization called ICANN
    - called gTLDs
    - formerly granted by authority of the US government via an organization called InterNIC
    - US government and InterNIC ceded authority to ICANN late 1998
  - more on this topic later

Applications: DNS-3

- DNS operation:
  - basis for DNS is a collection of independent *name servers*
  - every domain has an *authoritative name server* that is responsible for knowing how to translate the name of every host in the domain (itself or by delegation)
  - every host knows the address of at least one name server (often the authoritative name server for the host's domain)
  - every *domain server* (including authoritative ones) knows the address of a another server
    - could be a server operated by an ISP
    - probably a *root server* operated by InterNIC or a related organization, or a national domain root server

Applications: DNS-3

- Basic DNS name resolution:
  - a host ask its local nameserver; if answer unknown:
    - nameserver asks its “upstream” nameserver
      - assuming it is a root-server, it either knows the answer, or knows the authoritative server that will know
    - the root-server can either forward the request to the delegated nameserver, or reply to the original request with the address of the authoritative server
      - if the root server forwards the request, called *recursive resolution*
      - if the root server returns the address of the server, so that the original requester must query the authoritative server, called *iterative resolution*

Applications: DNS-4

- DNS operation:
  - tedious methods; can be time-consuming, susceptible to server failure, top-level servers are saturated
  - use *caching* at every step: server may already know how to resolve a particular name
  - results from cache are marked as “non-authoritative”, and are marked with an expiry date (TTL – “time to live”)
- Note:
  - hierarchical names have nothing to do with physical arrangement of network
  - DNS has nothing to do with routing

Applications: other applications

- Most other applications use a client-server model:
  - a process binds to a port, waits
  - clients connect to that port
  - commands & replies, data are exchanged
  - client disconnects, server resumes wait
- Example: Telnet (RFC 854)
  - virtual “dumb terminal” application
  - server creates a “command-shell” process for incoming connections
  - client types commands which are sent to the server and issued on the server
  - not used (much) on PC; essential on Unix
  - telnet clients are useful for debugging:
    - e.g. connect to a server port, issue protocol messages by hand

Applications: other applications-2

- Example: Electronic mail
  - a set or related standards: message formats, user agents, transfer agents
  - message formats includes envelope, header, to/from strings, enclosures (e.g. MIME), etc.
  - user agents manage interface between people and software
    - not standardized (vendor-specific), but must interpret and implement other standard protocols
  - transfer agents manage sending and receiving of data
    - SMTP (Simple Mail Transfer Protocol) is a transfer-agent protocol
    - RFC 821 (August 1982)
    - also RFC 822, message formats

Applications: other applications-3

- SMTP is a “peer-related” protocol
- SMTP process can be a client in one instance and a server in another
- Basic process:
  - user-agent software contacts SMTP at port 25; establishes sender, receiver etc.
  - SMTP (as client) contacts receiver’s SMTP process (as server)
  - two SMTPs handshake, then transfer message
  - receiver SMTP deposits message in a mailbox, file, or other repository where it can be retrieved later:
    - local user agents just read the file
    - remote user agents can use:
      - POP3: Post Office Protocol V3 (RFC 1939)
      - IMAP: Internet Message Access Protocol (RFC 2060)

Applications: other applications-4

- FTP: File Transfer Protocol (RFC 959)
  - traditional client-server
  - server process binds to port 21, waits for incoming connections
  - uses second port for data transfer (20)
- NNTP: Network News Transfer Protocol (RFC 977)
  - similar architecture to SMTP (peers)
  - client software (a *newsreader*) contacts local NNTP port (119)
  - NNTP supports *push* and *pull* delivery
    - push: send new articles automatically
    - pull: request new articles
- General principles:
  - client–server or peer
  - transaction or connection-based
  - end-to-end communication with standardized protocols

Internet governance – so who's in charge now?

- Several possible answers:
  - nobody
  - everybody
  - governments (via ISO, ITU & related)
  - big corporations (who own the wires)
  - self-appointed committees
- Some of the players:
  - IANA: Internet Assigned Numbers Authority (www.iana.org)
    - managed IP numbers and domain names on behalf of the US government until 1998
      - principle agency was InterNIC (www.internic.net)
      - InterNIC still exists as a central database of gTLD domain name registrant information
    - IANA still exists as a central database of ccTLD registrar information and protocol & well-known number information

… who's in charge, 2

- The big player is now ICANN: Internet Corporation for Assigned Names and Numbers (www.icann.org)
  - an international not-for-profit with representation by geographic territory and economic influence and international standards bureaucracy
  - delegates by geographic territory:
    - ARIN: Americas Registry for Internet Numbers
      - reference: www.arin.net
      - Americas and sub-Saharan Africa
    - RIPE: Reseaux IP European
      - reference: www.ripe.net
      - Europe, Middle East, Africa
    - APNIC: Asia Pacific Network Information Centre
      - www.apnic.net
      - Asia, India, Australasia

… who's in charge, 3

- ARIN, RIPE and APNIC:
  - successors to IANA per ICANN, 1997
  - responsible for the allocation and management of
    - IP addresses
    - root servers
    - protocol port assignments
  - maintains databases of IP addresses and Autnonomous System Numbers
    - provides "number-to-name" translations
    - AS number assignments
    - routing information registry (eg BGP table generation)

… who's in charge, 4

- More players:
  - IETF: Internet Engineering Task Force
    - international in scope, primarily technical
    - composed of academic and industry representatives
    - includes ISOC (Internet Society), IAB (I. Architecture Board), IESG (I. Engineering Steering Group), IRTF (I. Research Task Force)
    - currently responsible for Internet standards and the RFC process (as described in RFC 2026)
    - reference: www.ietf.org
  - W3C: World Wide Web Consortium
    - standards related to the Web
      - HTML, DHTML, XML, etc
      - HTTP
    - not responsible for browsers

… who's in charge, 5

- None of these have anything to do with managing the allocation of Internet names
  - ICANN now has the responsibility of granting authority over names, but doesn't do the management itself
  - nations have control over their own namespace (as defined by their two-letter ISO country code)
  - generic top-level domains (.com etc.) are managed privately (sanctioned by ICANN)
- ICANN approved a new set of gTLDs in November 2000:
  - originally: .com, .net, .org, .edu, .gov, .mil, .int
  - new gTLDs: .aero, .biz, .coop, .info, .museum, .name, .pro
  - there is supposed to be some useful meaning to these, but the market will decide

… who's in charge, 6

- For gTLDs, ICANN approves Registrar organizations
  - the most significant of these is Network Solutions, the privatized InterNIC organization (now owned by Verisign)
  - there are hundreds more, new ones every day
- For ccTLDs, ICANN defers to national governments who can
  - run things themselves
  - set up NPOs
  - licence to private sector
- Rules of eligibility and conflict resolution for gTLD names handled by ICANN with assistance from WIPO (World Intellectual Property Organization (www.wipo.org)
  - trying to influence rules for ccTLDs, too

… who's in charge, 7

- CIRA
  - Canadian Internet Registration Authority
    - www.cira.ca
    - manages .ca on behalf of the government
    - NPO created 1998, replaced volunteer organization operated by UBC
  - prior to CIRA, .ca had arcane eligibility rules
    - rules are now similar to most countries
      - residency requirement
      - pay the fee
  - CIRA does not do registrations itself, it just sets the rules and manages the .ca root server
    - CIRA certifies registrars who do the actual registrations

… who's in charge, 8

- Other national agencies:
  - .ac; Ascension Island; www.nic.ac
  - .zw; Zimbabwe; www.zptc.co.zw (?)
  - for everything between, see

    http://www.iana.org/cctld/cctld-whois.htm

- Some nations maintain residency requirements ("Real and Substantive Connection"), others have sold the rights to the highest bidder, e.g.
  - .tv (Tuvalu), .to (Tonga), .fm (Federation of Micronesia)