Applications and Network Management

Outline

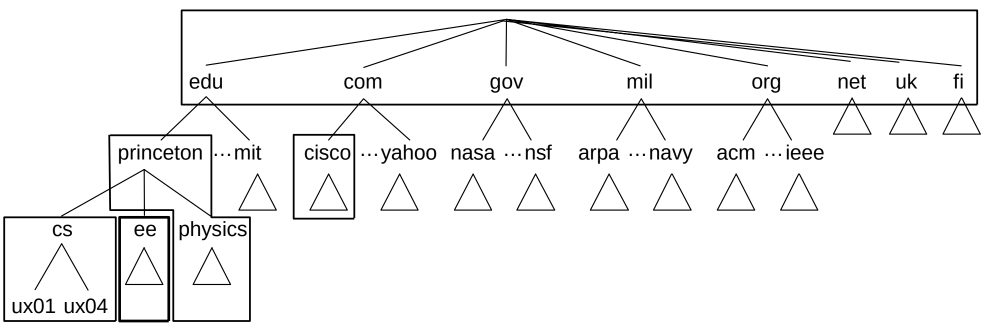
- Name service (DNS)
- Electronic mail (SMTP, MIME)
- World wide web (HTTP)
- Network management (SNMP)
- Network services

Name service (DNS)

- Maps user-friendly names into router-friendly addresses
  - middleware: fills the gap between applications and the underlying network
  - transported using UDP, port number 53
- Host names
  - variable length and mnemonic
  - typically contain no information that helps network to locate the host
- IP addresses
  - fixed-length numeric address
  - may have routing information embedded in them
- Terms:
  - **namespace** = set of possible names, flat or hierarchical
  - naming system maintains a collection of **bindings** of names to values
  - given a name, a **resolution mechanism** returns the corresponding value
  - a **name server** is an implementation of the resolution mechanism
  - DNS (Domain Name System) = name service in Internet

DNS domain hierarchy

- First level hierarchy
  - domains for each country + edu, com, gov, mil, org, net
  - DNS first level managed by Internet Corporation for Assigned Names and Numbers (ICANN), also manages address allocations
- Hierarchy is partitioned into subtrees, zones
  - zone corresponds to fundamental implementation unit in DNS (i.e., a name server)



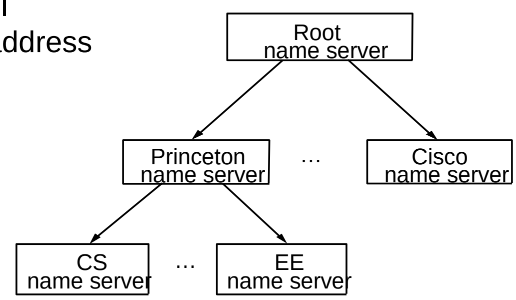
DNS domain hierarchy (cont)

- Zones implemented in two or more name servers (redundancy)
  - clients send queries to name servers
  - servers response with final answer or pointer to another server
- Name binding database consists of resource records
  - format: <Name, Value, Type, Class, TTL>
  - Type: how Value is interpreted,
    - A: means that Value is an IP address, name-address mapping
    - NS: Value contains name for host that knows how to resolve the name
    - CNAME: Value is a canonical name for host, used to define aliases
    - MX: Value gives the domain name for a host running a mail server
  - Class: only widely used class IN (Internet)
  - TTL: how long resource record is valid (used by servers that cache resource records from other servers)
  - can use alias for company web server $\Rightarrow$ web server to be changed without remote users being affected
  - MX allows administrators to change the mail host without changing user email addresses

DNS domain hierarchy (cont)

- Root name server: NS record for each 2nd level server + A record that translates name into IP address

    <princeton.edu, cit.princeton.edu, NS,IN>
    <cit,princeton.edu, 128.196.128.233, A, IN>



- At 2nd level, records contain either final answers or pointer to 3rd level name servers

  <cs.princeton.edu, gnat.cs.princeton.edu, NS, IN>
  <gnat.cs.princeton.edu, 192.12.69.5, A, IN> (pair like above)

  <jupiter.physics.princeton.edu, 128.196.4.1, A, IN> (final record)

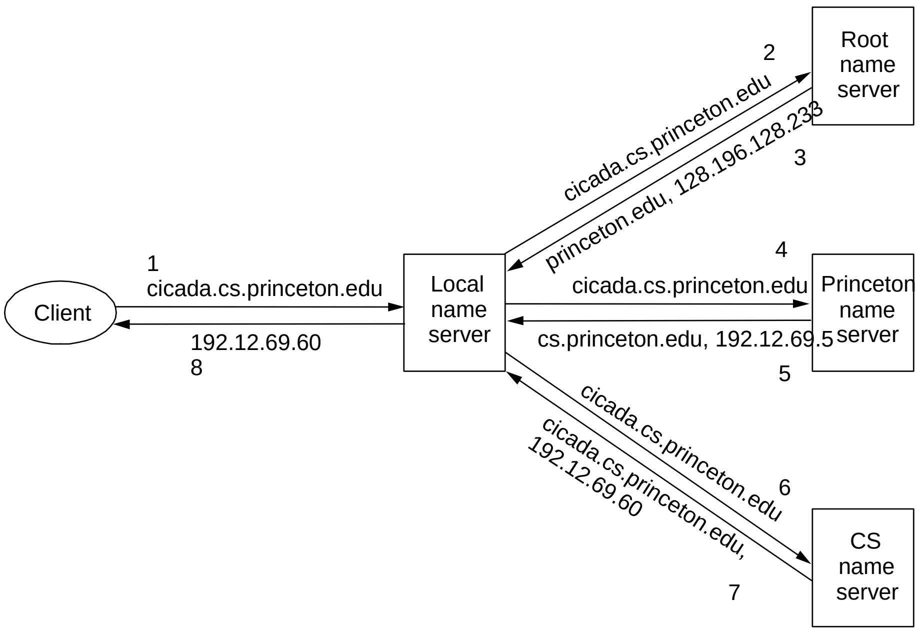
- Lowest level contains final records, aliases for hosts (CNAME) and MX records

Name resolution

- How did the client locate the root server in the first place?
  - name-to-address mapping for one or more name servers is well know (published outside the naming system itself)
  - in practice, client program initialized with the address of a local name server
    - client makes a query to local server ⇒ local server makes queries further
    - advantages
      - only the servers need to know about root name servers
      - local server gets to see the responses (can cache these)
  - on a host running DNS (in Unix), try “nslookup” or “host <hostname>”

- Note: Internet has identifiers at 3 levels - domain names, IP addresses, and physical network addresses
  - users give domain names in applications ⇒ applications use DNS to translate these into IP addresses ⇒ IP does forwarding at each router, so it maps IP addresses into another (next hop router) ⇒ IP engages ARP to translate the next hop IP address into a physical address

Name resolution (cont)



Numbers (1-8) show the sequence of steps in the process

Outline

- Name service (DNS)
- Electronic mail (SMTP, MIME)
- World wide web (HTTP)
- Network management (SNMP)
- Network services

Traditional Applications and Protocols

- Traditional = elastic data traffic, without timeliness requirements
  - real time traffic treated later...

- SMTP: Simple Mail Transfer Protocol
  - exchange of electronic mail
  - RTC 822 and MIME define the format of email messages
- HTTP: HyperText Transport Protocol
  - communication between Web browsers and Web servers
  - HTML specifies the form or the Web pages
- SNMP: Simple Network Management Protocol
  - querying (and modifying) the state of remote network nodes
  - MIB (management information base) defines the variables that can be queried

Electronic mail (SMTP, MIME)

- Mail service consists of
  - a mail reader,
  - a message transfer protocol (SMTP) and
    - SMTP = Simple Message Transfer Protocol
  - companion protocols RFC 822 & MIME

- Mail access protocol: retrieval from server
  - reader programs: Netscape Messenger, Outlook, Eudora, Mozzilla
  - POP3: Post Office Protocol (RFC 1939)
    - authorization (agent ⇔ server) and download
    - downloads mails to your own local host
  - IMAP: Internet Mail Access Protocol (RFC 1730)
    - more features (more complex)
    - manipulation of inbox and stored messages on server
  - HTTP: Hotmail , Yahoo! Mail, etc.

Electronic mail (cont)

- Message format:
  - RFC 822: message has two parts - a header and a body
    - both in ASCII text
  - MIME: extends RFC 822 so that message can contain all sorts of data
    - data still presented as ASCII text
  - ASCII format $\Rightarrow$ human can pretend to be an smtp client

- Message header:
  - series of <CRLF>-terminated lines (carriage-return + line-feed)
  - separated from message body by blank line
  - each header line contains a Type and a Value separated by a colon
    - To: student@hut.fi
    - Subject: lecture notes

MIME

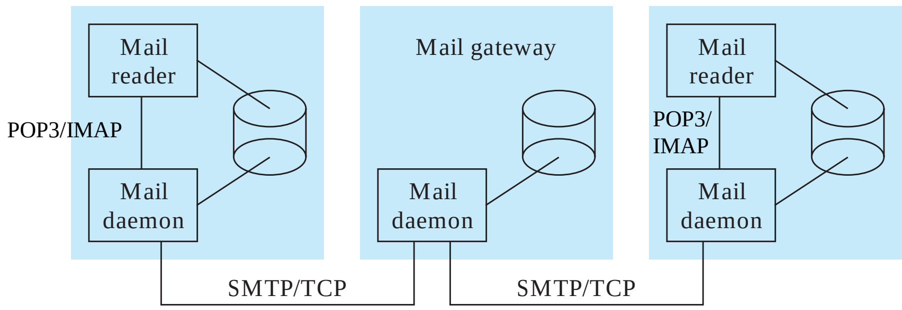
- Extends RFC 822 to allow email messages to carry audio, video, images, Word documents etc.
- Consists of 3 basic pieces
  - collection of header lines
    - extend the original set defined in RFC 822
    - ex. MIME-version, Content-Description, Content-Type, Content-Transfer-Encoding..
  - definitions for a set of content types
    - ex. image/gif, image/jpeg, text/plain, text/richtext, application/postscript, application/msword
  - a way to encode various data types so that they can be shipped in an ASCII mail message
    - base64 coding of binary data into ASCII: map every 3 bytes of the original binary data into 4 ASCII characters

MIME Example (text + attached file)

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="XXXXboundary text"
From: John Doe <jdoe@microsoft.com>
To: Jane Doe <janedoe@nowhere.edu>
Subject: ...
Date: Tue, 04 Feb 2003 20:15:00 -0200

This is a multipart message in MIME format.

--XXXXboundary text
Content-Type: text/plain

this is the body text

--XXXXboundary text
Content-Type: text/plain; filename="test.txt"
Content-Disposition: attachment;

this is the attachment text

--XXXXboundary text--
```

Message transfer (SMTP)

- E-mail delivery
  - mail reader ⇒ message to mail daemon ⇒ daemon uses SMTP running over TCP to get message to a daemon in another machine ⇒ this daemon puts the message into user's mailbox
  - SMTP uses TCP on port 25
- Mail traverses many mail gateways that store and forward email msgs
  - mail gateway vs. IP router? IP router stores datagrams in memory and tries to retransmit them for a short period of time (fraction of seconds), mail gateway buffer messages on disk and try resending for days or so



SMTP Example

- Exchange between sending host cs.princeton.edu and receiving host cisco.com (responses in italics)
  - you can be an SMTP client by first starting a TCP connection at a mail server in port 25, "telnet <servername> 25", then use the commands below

```
HELO cs.princeton.edu
250 Hello daemon@mail.cs.princeton.edu [128.12.169.24]

MAIL FROM:<Bob@cs.princeton.edu>
250 OK

RCT TO:<Alice@cisco.com>
250 OK

RCT TO:<Tom@cisco.com>
550 No such user here

DATA
354 Start mail input; end with <CTRL>.<CTRL>
Blah blah blah …
<CTRL>.<CTRL>
250 OK

QUIT
221 Closing connection
```

Outline

- Name service (DNS)
- Electronic mail (SMTP, MIME)
- World wide web (HTTP)
- Network management (SNMP)
- Network services

World Wide Web (HTTP)

- Web is a collection of cooperating clients and servers
  - everyone uses same protocol, HTTP
  - web browser used to open web pages
    - URL (Uniform Resource Locator) specifies location of object on the web (e.g., http://www.hut.fi/index.html)
  - opening a URL makes the browser open a TCP connection to port 80 to the given location, e.g., www.hut.fi, and the file index.html would be downloaded to your machine using HTTP over TCP
  - like SMTP, HTTP is a text oriented protocol

- Each HTTP message has the general form

```
START_LINE <CRLF>
MESSAGE_HEADER <CRLF>
<CRLF>
MESSAGE_BODY<CRLF>
```

HTTP, request message

- First line of HTTP message:
  - operation, Web page operation should be performed on, version of HTTP
  - for example, getting our laboratory's homepage manually

```
> telnet www.netlab.hut.fi 80

GET /index.html HTTP/1.1
Host: www.netlab.hut.fi
```

- Operations
  - OPTIONS request information about available options
  - GET retrieve document identified in URL
  - HEAD retrieve metainformation about document in URL
  - POST give information to server
  - PUT store document under specified URL
  - DELETE delete specified URL
  - TRACE loopback request message
  - CONNECT for use by proxies

HTTP, response message

- START_LINE: version of HTTP, 3-digit response code, text string giving reason for response
  - HTTP/1.1 202 Accepted
  - HTTP/1.1 404 Not Found

- Response message contains one or more MESSAGE_HEADER lines (additional information) and the requested page (HTML document, nontextual data encoded using MIME)

Code	Type	Example Reason
1xx	Info	request received, continuing process
2xx	Success	action successfully received, understood and accepted
3xx	Redirection	further action must be taken to complete the request
4xx	Client Error	request contains bad syntax or cannot be fulfilled
5xx	Server Error	server failed to fulfill an apparently valid request

HTTP and TCP connections

- HTTP version 1.0 made a separate TCP connection for each data item
  - waste of resources, especially when most items are small sized
- HTTP version 1.1 allows persistent connections: client and server can exchange multiple request/response messages over the same TCP connection
  - good:
    - eliminates the connection setup overhead
    - client can send multiple request messages -> TCP's congestion window mechanism operates more efficiently (not necessary to do slow start for each request)
  - bad:
    - neither the client nor server knows how long to keep a particular TCP connection open (problem for servers with thousands of connections)
    - client and server must watch if the other side has elected to close the connection (recall, both sides need to close the TCP connection)

Caching

- WWW cache = web proxy
- Benefits:
  - pages from nearby cache can be displayed quickly
  - can reduce servers' load
- Implementation at several (hierarchical) layers:
  - in user's browser
  - user's site can support a single sitewide cache (takes advantage of pages previously downloaded by other users)
  - ISPs may have their own caches
- Cache needs to make sure it is not responding with an out-of-date version of the page
  - server may assign an expiration date (Expires header field) to each page
  - HTTP conditional requests by using, i.e., If-Modified-Since message header

Outline

- Name service (DNS)
- Electronic mail (SMTP, MIME)
- World wide web (HTTP)
- Network management (SNMP)
- Network services

Network Management (SNMP)

- Possible tasks:
  - monitor faulty equipment in the network
  - keep track of the load on various nodes (need for new routers or links?)
  - etc., etc.
- Nodes in the network are distributed $\Rightarrow$ use the network to manage the network
  - need a protocol for reading (and writing) state information on different network nodes
- Simple Network Management Protocol (SNMP)
  - request/reply protocol that supports GET and SET messages
  - runs on top of UDP
  - client program uses SNMP to request information, SNMP server running on a node replies
  - depends on companion specification Management Information Base (MIB) that describes object structure of network elements

Management Information Base (MIB-II)

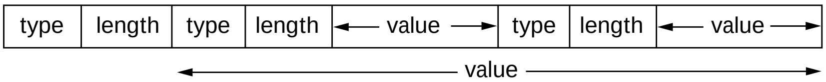
- Defines MIB variables, the information that can be retrieved from a network node (object oriented model)
- Variables organized into 10 groups, for example:
  - System: general parameters; where node is located, how long it has been up, system's name…
  - Interfaces: info about all network interfaces attached to this node; physical address of each interface, how many packets have been sent/received on each interface
  - Address translation: info about Address Resolution Protocol (ARP)
  - IP: IP variables such as routing table, number of successfully forwarded datagrams, statistics about datagram reassembly...
  - TCP: info on TCP traffic; number of passive/active opens, number of timeouts, default timeout…
  - UDP: info on UDP traffic; number of UDP datagrams…
  - OTHER GROUPS: ICMP, EGP, SNMP
  - Example: 1.3.6.1.2.1.4.3 = IpInReceives (nof IP packets)
    - 1.3.6.1.2.1 MIB db identifier, 4.3 = IP group, 3rd variable

Abstract Syntax Notation One (ASN.1)

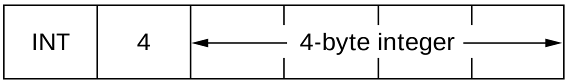
- MIB related needs:
  - client uses precise syntax to indicate which MIB variable is needed
    - ASN.1 defines an object identification scheme
    - MIB uses this to get a globally unique identifier to each MIB variable
  - server uses precise representation for values it returns
    - ASN.1/BER (Basic Encoding Rules) defines a representation for different data types
    - MIB defines the type of variable, and then value is encoded according to ASN.1/BER
- SNMP client puts the ASN.1 identifier for the MIB variable it wants in the request message $\Rightarrow$ server maps identifier into a local variable $\Rightarrow$ retrievers the value $\Rightarrow$ uses ASN.1/BER to encode the value before sending it back to the client
- Lists and tables (and other compound types) retrieved using SNMP GET-NEXT operation (returns value & ID for next variable)
  - Bulk operation exists, as well (GET-BULK)

ASN.1 / BER

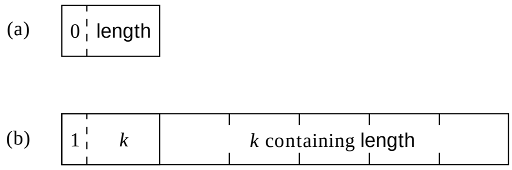
- Each data item is presented by a triple
  - <type, length, value>, TLV-coding
  - type: typically an 8-bit field
  - length: specify how many bytes make up the value



compound type



a 4-byte integer



Representation for length:
a) 1-byte,
b) multibyte (if length > 127 bytes)

Outline

- Name service (DNS)
- Electronic mail (SMTP, MIME)
- World wide web (HTTP)
- Network management (SNMP)
- Network management and operation

Components of network management

- SNMP is a protocol for managing and monitoring network elements
  - just a partial solution to the “whole problem”
- Network management comprises much more…

Trad. network management functions

- Fault management
  - identification, isolation, fixing
  - done by requesting, testing and analyzing reports
- Configuration management
  - naming of components, properties of components, states of components
- Accounting
  - important for operators
- Performance management
  - knowledge on what part of network needs to be improved
- Security management

Other aspects

- Chargeback
  - charge only for used resources
- Systems management
  - system is more than just the network elements
- Cost management
  - costs consist of maintenance, Mean Time Between Failures, Mean Time To Repair
  - focus on reliability, functionality, manageability

- TMN = comprehensive framework for network management
  - set of standards developed by ITU (M.3xxx series), based on OSI management (CMIP)
  - SNMP can be used to address some of the issues within TMN (not all)

Network operation

- Focus on reliability and availability
  - reliable components, backups
  - extra components to give fault tolerance, reliable power supply
- Costs
  - equipment (20%), technical support (20%), maintenance (15%), help desk for users (45%)
- Services offered by the network system
  - data storage
    - basic service, possibly also authorization
  - databases
    - searches, different data types, data with management
  - printers
  - data delivery (network)
    - supports contacts and co-operation
  - applications

Network operation (cont)

- Storing the data
  - online: available immediately (< 1 s), on disk
  - nearline: automatically available (< 1 min), on tape or cd-rom
  - offline: available manually, on tape
  - hierarchy in storing: online $\Rightarrow$ nearline $\Rightarrow$ offline

- Backups protect from
  - failure in storage media, program errors / viruses, human errors, physical threats (fire, water…) IF AND ONLY IF making backups is systematic and backups are physically safe