Internet Foundations
Internet Foundations

- brief review of history
- Internet standardization
- Internet architecture
- basic Internet protocols: IP, UDP, TCP, …

Internet Standards
Who makes the rules?

ITU.T (itu.int): telecom standards by 16 study groups:
  E. Overall network operation, telephone service (E.164)
  G. Transmission systems and media, digital systems and networks (G.711)
  H. Audiovisual and multimedia systems (H.323)
  V. Data communication over the telephone network (V.24)
  X. Data networks and open system communications (X.25)

IETF and IESG (ietf.org): (Internet Engineering Task Force, … Steering Group)
  develop/bless protocols (“RFCs”)
  open admission, but not quite egalitarian

W3C: HTML, XML, …
Internet Operational Bodies

**ISOC**: membership organization; legal “home” of IETF

**IANA**: (Internet Assigned Numbers Authority) assigns numbers, top-level domains

**NANOG**: North American Network Operators Group

**ICANN**: administers IANA, registrars

**RIPE, ARIN, APNIC**: hands out blocks of addresses, regionally
Internet Architecture Board: IAB

- architectural oversight
- process appeals
- elected by ISOC through nominations committee

Internet Engineering Steering Group (IESG): approves standards, composed of area directors
IETF Areas

- applications (28 WGs): calendar, HTTP, LDAP, MIME, NNTP, …
- general (1): POISSON
- internet (15): IPv6, IP over x, interface MIBs
- operations and management (23): MIBs, routing policy, benchmarking
- routing (17): multicast, RIP, BGP
- security (14): S/MIME, TLS, PGP, XML security
- transport (24): RTP, SIP, RTSP, RSVP admission, TCP
- user services (4): handbooks, guides, standard policies
IETF Working Groups

• headed by chair(s) designated by AD
• should be single, well-defined topic
• discussions on public mailing list
• small groups of authors do detail work
• meet at IETF (three times a year)
• possibly interim meetings
• done ➔ dissolve
IETF standards process

- IETF working group
- Internet Architecture Architecture Board
- Internet Engineering Steering Group
- Individuals

RFC editor

 WG chair
draft-ietf-wg-#*

WG chair approves

check for format
WG chair approval

I-D

Internet Drafts editor

revise

WG last call
IETF last call

Internet Engineering Steering Group

RFC

standards-track RFC

Proposed

Draft

Standard (STD)

Best Current Practice (BCP)

Informational

Experimental

Historic
Standardization process

RFC 2026:

1. new topic ➔ BOF at IETF meeting
2. if response, create working group with charter
3. create Internet drafts = temporary (≤ 6 months) working drafts
4. status and discussion presentations at IETF meetings
5. working group last call
6. IETF last call
7. IESG “votes” (by consensus)
8. published as RFC: proposed standard
9. 2 implementations + ≥ 6 months ➔ draft standard
10. operational experience + 4 months ➔ Internet standard (STD)
RFCs

- ASCII + PostScript, no charge (www.normos.org)
- published RFCs never change (no IP-1994)
- also:
  - experimental
  - informational (possibly “FYI”)
  - historic(al)
- anybody can submit RFC, but editor can filter for content, conflict with existing work
- check the April 1 ones… (RFC 1149)
Internet Access and Infrastructure
Who pays for the Internet?

NAP
public peering
MAE-E, MAE-W, ...

international lines

national network
(UUnet, Sprint, ...)

private peering

regional network
(Applied Theory)

ISP
company
university
ISP

proxy

AOL

log-in via modem

56 kb/s - 45 Mb/s

LAN
Network Access and Interconnection

- Ethernet
- firewall
- local telephone company
- 56kb/s - 2Mb/s
- point-of-presence (POP)
- modem concentrator
- T3
- regional network
- NAP
- national network
- phone lines+ node
- phone company
- telephone switch
- PC
- modem company
- company

- R
- R
Example: UUnet Backbone
Large Consumer ISPs

- 201 million Internet users in the world, 112.4 million in U.S. and Canada (1 subscriber = 2.5 users!).
- many lease facilities (e.g., AOL)

<table>
<thead>
<tr>
<th>company</th>
<th>subscribers (10^6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOL</td>
<td>20</td>
</tr>
<tr>
<td>EarthLink+Mindspring</td>
<td>3</td>
</tr>
<tr>
<td>NetZero</td>
<td>3</td>
</tr>
<tr>
<td>Prodigy</td>
<td>2</td>
</tr>
<tr>
<td>CompuServe</td>
<td>2</td>
</tr>
<tr>
<td>AT&amp;T Worldnet</td>
<td>1.8</td>
</tr>
<tr>
<td>Microsoft Network</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Mostly modem; 300,000 DSL and 1.1 million cable modem (end 1999).
Home Networking

phone lines  1-10 Mb/s, higher frequency than DSL
power lines  < 10 Mb/s
wireless  1-11 Mb/s (IEEE 802.11) in 2.4 GHz band
Carriers

About 40 backbones, use various right-of-ways

- UUnet (MCI)
- Sprint
- GlobalCrossing mostly cross-oceanic 20,000 (U.S.)
- AT&T Level3 railroad? 11,000 (goal: 16,000)
- PSINet leased
- Qwest railroad
- Williams pipelines 25,000
- Enron pipelines, HV

Peering: Equinix
## Internet Access

<table>
<thead>
<tr>
<th>method</th>
<th>media</th>
<th>downstream</th>
<th>upstream</th>
</tr>
</thead>
<tbody>
<tr>
<td>modem</td>
<td>POTS</td>
<td>≤ 53 kb/s</td>
<td>33.6 kb/s</td>
</tr>
<tr>
<td>Intercast</td>
<td>VBI</td>
<td>150 kb/s</td>
<td>modem</td>
</tr>
<tr>
<td>ISDN</td>
<td>POTS</td>
<td>128 kb/s</td>
<td>128 kb/s</td>
</tr>
<tr>
<td>DSL</td>
<td>POTS</td>
<td>160 kb/s</td>
<td>160 kb/s</td>
</tr>
<tr>
<td>ADSL</td>
<td>POTS</td>
<td>0.6…9 Mb/s</td>
<td>16…640 kb/s</td>
</tr>
<tr>
<td>cable modem</td>
<td>CATV</td>
<td>10 Mb/s</td>
<td>1 Mb/s</td>
</tr>
<tr>
<td>T1</td>
<td>copper</td>
<td>1.5 Mb/s</td>
<td>1.5 Mb/s</td>
</tr>
<tr>
<td>T3</td>
<td>fiber, copper</td>
<td>45 Mb/s</td>
<td>45 Mb/s</td>
</tr>
</tbody>
</table>
Network utilization

Averaged over one week:

- local phone line 4%
- U.S. long distance switched voice 33%
- Internet backbones 10-15%
- private line networks 3-5%
- LANs 1%

- peak personal-use hours: 5-11 pm
- “world wide wait”: web servers? DNS? NAPs? access?
- average speed: 40 kb/s
ISP Service

- average connect time: 310.3 min/month home, 417.4 min/month work → $3.85 hour
- 66 MB average transfer/month → 33 c/MB
- 10:1 modem concentration ratio
- T1: 500 GB/month each direction → 0.3c/MB ($1500/month)
- but ISP T1 utilization ≈ 40-45%
- ISP costs: $2.50/month for phone line, $2/month for equipment depreciation, $0.20/month for network
- fiber: $30,000-$50,000/mile
## ADSL Limits

<table>
<thead>
<tr>
<th>name</th>
<th>Mb/s</th>
<th>distance (ft)</th>
<th>km</th>
<th>&lt;80%</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS1 (T1)</td>
<td>1.544</td>
<td>18,000</td>
<td>4.5</td>
<td>&lt;80%</td>
</tr>
<tr>
<td>E1</td>
<td>2.048</td>
<td>16,000</td>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>DS2</td>
<td>6.312</td>
<td>12,000</td>
<td>3.0</td>
<td></td>
</tr>
<tr>
<td>E2</td>
<td>8.448</td>
<td>9,000</td>
<td>2.3</td>
<td></td>
</tr>
<tr>
<td>1/4 STS-1</td>
<td>12.960</td>
<td>4,500</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>1/2 STS-1</td>
<td>25.920</td>
<td>3,000</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>STS-1</td>
<td>51.840</td>
<td>1,000</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>OC-3</td>
<td>155.000</td>
<td>100</td>
<td>0.03</td>
<td></td>
</tr>
</tbody>
</table>
### ADSL Pricing Example

Bell Atlantic, May 1999:

<table>
<thead>
<tr>
<th>downstream (kb/s)</th>
<th>upstream (kb/s)</th>
<th>rate</th>
<th>ISP</th>
</tr>
</thead>
<tbody>
<tr>
<td>640</td>
<td>90</td>
<td>$39.95</td>
<td>$10</td>
</tr>
<tr>
<td>1.6</td>
<td>90</td>
<td>$59.95</td>
<td>$40</td>
</tr>
<tr>
<td>7.1</td>
<td>680</td>
<td>$109.95</td>
<td>$80</td>
</tr>
</tbody>
</table>
Cable plant architecture

Cable plant architecture

- coax cable: < 1 GHz bandwidth, typically 500 MHz
- 35 TV channels in typical older CATV systems
- 500–2000 homes for single headend or fiber node
- head-end to residence < 80 km
- fiber node to residence < 350 m
- US: 67% of households have, 95 mio. residence “passed”
Cable modems

- always-on, but maybe temporary IP addresses
- hybrid fiber coax
- CMTS (cable modem termination system) = “headend”
- Ethernet interface to user’s PC
- but: conversion to bidirectional amplifiers, power
- DHCP + network address translation (NAT)
- conversion cost: $200–$800
- standards:
  - Data-Over-Cable Service Interface Specification (DOCSIS)
  - IEEE 802.14: ATM MAC
– Multimedia Cable Network System Partners (MCNS): contention
– Davic (Europe)
Cable plant architecture

Cable modem network modes
Cable modem: downstream

- one or more 6 MHz channels in 54–550 MHz range
- typical bit consumption (no A/V): 40 kb/s, 4 kb/s upstream
- 30-50% active ➔ 420 customers per channel
- 64 QAM (6 bits/symbol) ➔ ≤ 30 Mb/s
- newer equipment: 256 QAM ➔ 40 Mb/s
Cable modem: upstream

- 5–42 MHz (usually band < 3 MHz, typically 200 kHz)
- noise aggregation ➔ QPSK with 2 bits/symbol (5 Mb/s)
- actual throughput: 768 kb/s
- can’t use Ethernet-style CDMA.
- TDMA variation: headend asks for potential senders
  - headend returns grant: $2^k$ 6.25 µs mini slots
- send 6-byte request to transmit ➔ delay variation!
- encryption: 40/56 bit DES
Cable modems: IEEE 802.14 vs. MCNS

IEEE 802.14

MCNS

Concatenation
Cable modems: access delay

# Performance comparison

Keynote web retrieval performance (April 1999):

<table>
<thead>
<tr>
<th>Technology</th>
<th>5-11pm</th>
<th>8am-5pm</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSL (128/384)</td>
<td>3.55</td>
<td>4.30</td>
</tr>
<tr>
<td>Cable modem</td>
<td>3.97</td>
<td>3.68</td>
</tr>
<tr>
<td>T-1</td>
<td>1.83</td>
<td>2.36</td>
</tr>
</tbody>
</table>

- both shared somewhere

RTT delay:

<table>
<thead>
<tr>
<th></th>
<th>average</th>
<th>max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISDN</td>
<td>(\approx) 10 ms</td>
<td></td>
</tr>
<tr>
<td>CM</td>
<td>20-45 ms</td>
<td>860</td>
</tr>
<tr>
<td>ADSL</td>
<td>(\approx) 20–30 ms (50 ms for 1.5/224)</td>
<td>63</td>
</tr>
</tbody>
</table>
Some Terminology

internet: collection of packet switching networks interconnected by routers

(the) Internet: “public” interconnection of networks

dend system = host: computer that is attached to the network ↔ router; usually one network interface

router = gateway = intermediate system: routes packets, several interfaces

subnetwork: part of an internet (e.g., single Ethernet)

firewall: router placed between an organization’s internal internet and a connection to the external Internet, restricting packet flows to provide security.
# Internet WAN Physical Layers

<table>
<thead>
<tr>
<th></th>
<th>Gb/s</th>
<th>remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Giga Ethernet</td>
<td>1.25</td>
<td>fiber</td>
</tr>
<tr>
<td>T-3</td>
<td>0.045</td>
<td>fiber, TP or coax</td>
</tr>
<tr>
<td>OC-3c</td>
<td>0.155</td>
<td>fiber</td>
</tr>
<tr>
<td>OC-12</td>
<td>0.622</td>
<td>fiber</td>
</tr>
<tr>
<td>OC-48</td>
<td>2.4</td>
<td>fiber</td>
</tr>
<tr>
<td>OC-192</td>
<td>10</td>
<td>fiber</td>
</tr>
</tbody>
</table>
Dense Wavelength Division Multiplexing

- multiple optical λ in single fiber
- 1.6 to 2 Tb/s per fiber
- interfaces typically 622 Mb/s to 10 Gb/s
Link-Layer Mechanisms Used

Roughly in order of popularity:

- ATM
- IP over SONET (synchronous optical network)
- frame relay
- gigabit Ethernet (with range extenders)
- T1, T3
Asynchronous Transfer Mode (ATM)

- 48-byte cells plus 5-byte header
- routing by label swapping
- virtual circuits (VCs) and paths (VPs)
- in-order delivery, but cells can be lost
- adaptation layers:
  - AAL1: continuous bit rate (CBR); “circuit emulation”
  - AAL2: multiplexed low-delay voice
  - AAL3/4: data (rarely used)
  - AAL5: IP packet in several cells
Frame Relay

- variable-length packets
- permanent or switched virtual circuits (PVC, SVC)
- typically, lower bandwidth (≤ 45 Mb/s)
- popular as access mechanism, corporate networks
Internet Link Layers

- **UDP**
  - RFC 768

- **TCP**
  - RFC 761/2001/2581

- **IPv4**
  - RFC 791

- **Ethernet**
  - (RFC 894)

- **SONET**
  - (in progress)

- **SNAP**
  - RFC 768

- **LLC**
  - RFC 2684

- **AAL5**

- **ATM**

- **PPP**
  - RFC 2364

- **AAL5**

- **SONET**

- **DWDM**
  - fiber

- **OIF**
  - (in progress)
Wireless Access

- Industrial, Scientific, Medical (ISM) bands (unlicensed): 902–928 MHz (US only), 2.4 GHz, 5.8 GHz
- analog cellular: 800 MHz
- PCS: 1.9 GHz

Wireless Ethernet:

- 900 MHz, 2.4 GHz, or 5 GHz
- 1 or 2 Mb/s, soon 5.5 Mb/s or 11 Mb/s
- collision-based, with reservation (RTS/CTS)
- IEEE 802.11 = FH or DS

Cellular Digital Packet Data (CDPD):

- pauses in AMPS voice traffic
## Wireless access

<table>
<thead>
<tr>
<th>Technology</th>
<th>band</th>
<th>mod.</th>
<th>rate</th>
<th>open range (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td></td>
<td></td>
<td>8.0 k/bs</td>
<td></td>
</tr>
<tr>
<td>GSM data</td>
<td>1.9 GHz</td>
<td>TDMA</td>
<td>9.6 kb/s</td>
<td></td>
</tr>
<tr>
<td>CDPD</td>
<td></td>
<td></td>
<td>19.2 kb/s</td>
<td>km</td>
</tr>
<tr>
<td>Metricom Ricochet</td>
<td>902-928 MHz</td>
<td>FH</td>
<td>28.8 kb/s</td>
<td>300-450</td>
</tr>
<tr>
<td>Bluetooth</td>
<td>2.4 GHz</td>
<td>FH</td>
<td>432 kb/s</td>
<td>10</td>
</tr>
<tr>
<td>802.11</td>
<td>2.4 GHz</td>
<td>DS</td>
<td>1 Mb/s</td>
<td>540</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2 Mb/s</td>
<td>400</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4 Mb/s</td>
<td>195</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.5 Mb/s</td>
<td>120</td>
</tr>
</tbody>
</table>
Internet Traffic

- 5,000-8,000 TB/month or 15.4–24.7 Gb/s
- long-distance calls: 525 GDEM or 64 Gb/s
- all the world’s telephones: 600 Gb/s
- almost all (90%?) of the traffic is TCP
worldwide traffic (Gb/s)

- data
- voice

Year:
- 1996
- 1997
- 1998
- 1999
- 2000
- 2001
- 2002
Voice vs. Data Traffic

- local vs. LANs vs. private networks
- capacity vs. traffic
- hop length of data traffic < voice
- link utilization (higher for voice)
- revenue
# Protocol Contributions

<table>
<thead>
<tr>
<th>proto</th>
<th>src</th>
<th>dest</th>
<th>pkts</th>
<th>bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP</td>
<td>http</td>
<td></td>
<td>35%</td>
<td>66.4%</td>
</tr>
<tr>
<td>TCP</td>
<td>http</td>
<td></td>
<td>33%</td>
<td>7%</td>
</tr>
<tr>
<td>TCP</td>
<td>nntp</td>
<td></td>
<td>1.8%</td>
<td>3.8%</td>
</tr>
<tr>
<td>TCP</td>
<td>ftp</td>
<td></td>
<td>1.4%</td>
<td>3.2%</td>
</tr>
<tr>
<td>TCP</td>
<td>smtp</td>
<td></td>
<td>1.8%</td>
<td>1.9%</td>
</tr>
<tr>
<td>TCP</td>
<td>nntp</td>
<td></td>
<td>1.3%</td>
<td>1.5%</td>
</tr>
<tr>
<td>UDP</td>
<td>dns</td>
<td>dns</td>
<td>3.1%</td>
<td>1.0%</td>
</tr>
</tbody>
</table>

April 1997, NLANR
Internet Names and Addresses
Names, addresses, routes

Shoch (1979):

**Name** identifies what you want,

**Address** identifies where it is,

**Route** identifies a way to get there.

Saltzer (1982):

**Service and users:** time of day, routing, …

**Nodes:** end systems and routers

**Network attachment point:** $\geq 1$ per node $\rightarrow$ multihomed host vs. router

**Paths:** traversal of nodes and links

binding = (temporary) equivalence of two names
### Internet names and addresses

<table>
<thead>
<tr>
<th></th>
<th>example</th>
<th>organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC address</td>
<td>8:0:20:72:93:18</td>
<td>flat, permanent</td>
</tr>
<tr>
<td>IP address</td>
<td>132.151.1.35</td>
<td>topological (mostly)</td>
</tr>
<tr>
<td>Host name</td>
<td><a href="http://www.ietf.org">www.ietf.org</a></td>
<td>hierarchical</td>
</tr>
</tbody>
</table>

Type of mapping:
- **DNS**: many-to-many
- **ARP**: 1-to-1
## Mappings in the Internet

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Category</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>whois</td>
<td>domain name</td>
<td>owner description</td>
</tr>
<tr>
<td>LDAP</td>
<td>key (name)</td>
<td>address, other info</td>
</tr>
<tr>
<td>YP</td>
<td>name</td>
<td>data item</td>
</tr>
<tr>
<td>DNS</td>
<td>host name</td>
<td>IP addresses</td>
</tr>
<tr>
<td></td>
<td>IP address</td>
<td>host name</td>
</tr>
<tr>
<td>atmarp</td>
<td>IP address</td>
<td>ATM NSAP</td>
</tr>
<tr>
<td>ARP</td>
<td>IP address</td>
<td>Ethernet address</td>
</tr>
<tr>
<td>RARP</td>
<td>MAC address</td>
<td>IP address</td>
</tr>
</tbody>
</table>
The Internet Domain Name System

We’ll talk about *name resolution* later…

```
host name
(has IP address)
```

```
lupus.fokus.gmd.de
```

Anywhere from two to $\infty$ parts
Internet (IP) Addresses

Each Internet host has one or more globally unique 32-bit IP addresses, traditionally consisting of a network number and a host number:

<table>
<thead>
<tr>
<th>Network</th>
<th>Host</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>Class A</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>Class B</td>
</tr>
<tr>
<td>110</td>
<td></td>
<td>Class C</td>
</tr>
<tr>
<td>1110</td>
<td></td>
<td>Class D</td>
</tr>
<tr>
<td>multicast address</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- originally, two-level hierarch $\rightarrow n$-level, changing
- an IP address identifies an *interface*, not a host!
- a host may have two or more addresses. Why?
Internet addresses

- (almost) every *interface* has one
- but may
  - change (dial-in)
  - have lots (WWW servers)
  - have none (some routers)
  - not be globally unique
- old: class-{A,B,C} ➔ 2-level addressing: network,host
- new: classless interdomain routing (CIDR) ➔ aggregation, route on prefix and mask
IP addresses

- dotted decimal notation: 4 decimal integers, each specifying one byte of IP address:
  - host name: lupus.fokus.gmd.de
  - 32-bit address: 1100 0000 0010 0011 1001 0101 0011 0100
  - dotted decimal: 192.35.149.52

- loopback: 127.0.0.1 (packets never appear on network)

- own network (broadcast): hostid = 0; own host: netid = 0

- directed broadcast: hostid = all ones

- local broadcast: 255.255.255.255
CIDR: Classless Interdomain Routing

- problem: too many networks ➔ routing table explosion
- problem: class C too small, class B too big (and scarce)
- discard class boundaries ➔ supernetting
- ISP assigns a contiguous group of $2^n$ class C blocks
- “longest match routing” on masked address; e.g. 192.175.132.0/22

<table>
<thead>
<tr>
<th>address/mask</th>
<th>next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.175.132.0/22</td>
<td>1</td>
</tr>
<tr>
<td>192.175.133.0/23</td>
<td>2</td>
</tr>
<tr>
<td>192.175.128.0/17</td>
<td>3</td>
</tr>
</tbody>
</table>

- e.g.,: all sites in Europe common prefix ➔ only single entry in most U.S. routers
Example: ifconfig

ifconfig -a
le0:  flags=863<UP,BROADCAST,NOTRAILERS,RUNNING>
      inet 192.35.149.117 netmask fffffff00
      broadcast 192.35.149.0
fa0:  flags=863<UP,BROADCAST,NOTRAILERS,RUNNING>
      inet 194.94.246.72 netmask fffffff00
      broadcast 194.94.246.0
qaa0: flags=61<UP,NOTRAILERS,RUNNING>
      inet 193.175.134.117 netmask fffffff00
qaa1: flags=61<UP,NOTRAILERS,RUNNING>
      inet 129.26.216.231 netmask ffff0000
qaa2: flags=60<NOTRAILERS,RUNNING>
qaa3: flags=60<NOTRAILERS,RUNNING>
lo0:  flags=849<UP,LOOPBACK,RUNNING>
      inet 127.0.0.1 netmask ff000000
IP address exhaustion

As of February 2000,

- 61.1% of available address space allocated
- 49.4% of allocated address space announced
- 30.2% of available address space announced

Routing table:

- 71,717 “autonomous system” (AS) entries
- 41,256 of which are /24
Routing Table Entries
Network Address (and Port) Translation (NA(P)T)

- most corporations use private address space, also residential
- 10/8, 172.16/12, 192.168/16
- NAT translates internal ⟷ external as needed
- works for outgoing TCP connections: POP, HTTP, SMTP, Telnet
- need application layer gateway (ALG) for out-of-band protocols (ftp, SIP, RTSP, H.323, …)
- problems:
  - controlled connections (ftp, Internet telephony, media-on-demand)
  - UDP services (streaming media)
  - security – rewriting breaks IPsec
- suggestion: Realm-Specific IP (RSIP) makes host aware of mapping
Problems with IP Addresses

- if a host moves from one network to another, its IP address changes
- currently, mostly assigned without regards to topology → too many networks
- CIDR
- limited space → IPv6
- class thresholds: class C net grows beyond 254 hosts
- hard to change: hidden in lots of places
- multihomed host: path taken to host depends on destination address
Multihoming

• = one “stub” network, multiple providers

• options:

  1. global prefix ➔ aggregation ↓
  2. divide network ➔ no redundancy
  3. multiple addresses ➔ applications need to try several, address space use ↑
Mobility and Renumbering

- renumber if immediate or up-stream provider changes
- mobility: change network attachment point
- mobility = renumbering: network “location” changes
- IP address as location ➔ keep address, break aggregation
- renumbering is hard: configuration files, transition
- IP address as identifier ➔ break connections
Subnetting

- large organizations: multiple LANs with single IP network address
- subdivide “host” part of network address ➔ subnetting

![Subnetting Diagram]

- Network 150.17.1.0 (mask fffffff0)
  - 254 nodes
  - 150.17.1.1
  - 150.17.1.2
  - 150.17.1.3

- Network 150.17.2.0 (mask fffffff0)
  - <256 subnets
  - 150.17.2.1
  - 150.17.2.2
  - 150.17.2.3

Internet

netmask: 0xff ff 00 00
How does a packet get to the server?

E.g., web page from http://www.cs.umass.edu:

• get host name www.columbia.edu from URL;

• DNS: translate to IP address 128.59.35.60

• is it on local network? no ➤ find local router

• local router sends to Internet

• Internet routes to Columbia network router (128.59.?.?)

• Columbia router routes to web server
Peeking inside a packet
IP Forwarding

get destination IP address D
if network(D) == directly attached network {
    ARP: D -> MAC address
    put in link layer frame
    forward
} else {
    foreach entry in routing table {
        if (D & subnet mask) == network(entry) {
            get next hop address N
            ARP: N -> MAC address
            put in link layer frame
            forward
        }
    }
}

IP source/destination remains same, MAC changes
IP Forwarding
128.10.0.0

193.175.132.0/16 193.175.132.1
default 193.175.132.2

193.175.132.17

193.175.132.1

193.175.132.25

193.175.132.0

anything else

128.10.0.0/16 193.175.132.1
default 193.175.132.2

Ethernet
GMD Fokus Network
Network Layer: IPv4 and IPv6

- unreliable datagram ➔ misorder, loose, duplicate
- 32-bit (IPv6: 128 bit) globally unique addresses
- no checksum on payload
- allow *fragmentation* of large packets into MTU-sized frames
- 20 (IPv6: 40) byte header
- IP multicast: receiver group with anonymous membership
IPv4
IPv4 Service Model

datagram: each packet is independent of all others

best effort: packet may arrive or not after some time
IPv4

- independent packets
- unreliable
- might be reordered (rare), delayed, duplicated, …
- but: minimal service on top of anything (see RFC 1149)
- only header checksum
IPv4 Header

RFC 791
A diagram of an IP packet header, showing the following fields:

- **Version**: 4 bits
- **Header length**: 4 bits
- **Type of service**: 8 bits
- **Total length**: 16 bits
- **Identification**: 16 bits
- **Flags**: 3 bits
- **Fragment offset**: 24 bits
- **Time-to-live**: 8 bits
- **Protocol identifier**: 8 bits
- **Header checksum**: 16 bits
- **Source IP address**: 32 bits
- **Destination IP address**: 32 bits
- **IP options**: 0 to 40 bytes
- **Data**: variable length

Legend:
- **DF**: Do Not Fragment
- **MF**: More Fragments

The diagram also indicates that fields may be modified by fragmentation and by router.
**IPv4**

**version:** always 4

**TOS (type of service):** precedence (3 bits) and “minimize delay”, “maximize throughput”, “maximize reliability”, “minimize cost” bits ➤ rarely used

**identifier:** identifier, different for each packet from host

**TTL:** time to live field; initialized to 64; decremented at each router ➤ drop if TTL = 0 (prevent loops!)

**protocol:** next higher protocol (TCP: 6, UDP: 17)

**header checksum:** add together 16-bit words using one’s complement ➤ optimized for software
IP Fragmentation and Reassembly

data link protocol may limit packets < 65,536 bytes \(\rightarrow\) transport layer packet may be too big to send in single IP packet

**ORIGINAL PACKET**

<table>
<thead>
<tr>
<th>id</th>
<th>flg</th>
<th>offset</th>
<th>src</th>
<th>dest</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>0</td>
<td>Y</td>
<td>Z</td>
<td>8K</td>
</tr>
</tbody>
</table>

**YIELDS TWO FRAGMENTS**

- **First Fragment**
<table>
<thead>
<tr>
<th>id</th>
<th>flg</th>
<th>offset</th>
<th>src</th>
<th>dest</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1</td>
<td>0</td>
<td>Y</td>
<td>Z</td>
<td>4K</td>
</tr>
</tbody>
</table>

- **Second Fragment**
<table>
<thead>
<tr>
<th>id</th>
<th>flg</th>
<th>offset</th>
<th>src</th>
<th>dest</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>4K</td>
<td>Y</td>
<td>Z</td>
<td>4K</td>
</tr>
</tbody>
</table>
IP Fragmentation and Reassembly

- split TPDU into *fragments*
  
  - each fragment becomes its own IP packet (routers don’t care)
  
  - each fragment has same identifier, source, destination address
  
  - fragment offset field gives offset of data from start of original packet
  
  - *more fragments* (MF) flag of 0 if last (or only) fragment of packet
  
  - fragments reassembled only at final destination
  
  - routers must handle at least 576 bytes
  
  - *do not fragment* bit prevents fragmentation => drop + error message
  
  - avoid multiple fragmentation (1500 → 620) => MTU discovery
IP Options

Extend functionality of IP without carrying useless information:

- security and handling restrictions for military
- determine route (source route)
- record route
- record route and timestamps

(rarely used ↔ not all routers support them)
**IP Record Route Option**

- source creates empty list of $\leq 9$ IP addresses
- option: length, pointer, list of IP addresses
- routers note outgoing interface in list
- ...and bump pointer
IP Source Route Option

• source determines path taken by packet (≤ 9 hops)
• *loose*: any number of hops in between
• *strict*: every hop; if not directly connected, discard
• same format as record route option
• router overwrites with address of outgoing interface
• must be copied to fragments
• destination should reverse route for return packets
• not too popular ➔ router performance ↓
ICMP

- used to communicate network-level error conditions and info to IP/TCP/UDP entities or user processes
- often considered part of the IP layer, but
  - IP demultiplexes up to ICMP using IP protocol field
  - ICMP messages sent within IP datagram
- ICMP contents always contain IP header and first 8 bytes of IP contents that caused ICMP error message to be generated

<p>| 20-byte standard IP header | 8 bit ICMP type | 8 bit ICMP code | 16-bit checksum | contents of ICMP msg |</p>
<table>
<thead>
<tr>
<th>type</th>
<th>code</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>echo reply (to a ping)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>destination network unreachable</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>destination host unreachable</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>destination protocol unreachable</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>destination port unreachable</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>fragmentation needed and DF set</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>destination network unknown</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>destination host unknown</td>
</tr>
<tr>
<td>3</td>
<td>...</td>
<td>other reasons</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>source quench (slow down)</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>redirect message to host</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>echo request (ping)</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>IS-ES router advertisement (new)</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>ES-IS router discovery (new)</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>time exceeded = TTL zero</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>IP header bad</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>address (subnet) mask request</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>address (subnet) mask reply</td>
</tr>
</tbody>
</table>
ping

- checks if host is reachable, alive
- uses ICMP echo request/reply
- copy packet data request → reply

```bash
ping -s gaia.cs.umass.edu
PING gaia.cs.umass.edu: 56 data bytes
64 bytes from gaia.cs.umass.edu (128.119.40.186): icmp_seq=0 time=276 ms
64 bytes from gaia.cs.umass.edu (128.119.40.186): icmp_seq=1 time=281 ms
64 bytes from gaia.cs.umass.edu (128.119.40.186): icmp_seq=2 time=276 ms
```

---gaia.cs.umass.edu PING Statistics---
4 packets transmitted, 3 packets received, 25% packet loss
round-trip (ms) min/avg/max = 276/277/281
**traceroute**

- allows to follow path taken by packet
- send UDP to unlikely port; ’time exceeded’ and ’port unreachable’ ICMP replies
- can use source route (-g), but often doesn’t work

```
$ traceroute gaia.cs.umass.edu
  1  gmdbgate (192.35.149.248)  6 ms  2 ms  2 ms
  2  188.1.132.142 (188.1.132.142)  263 ms  178 ms  188 ms
  3  gmdisgate.gmd.de (192.54.35.68)  153 ms  187 ms  151 ms
  4  icm-bonn-1.gmd.de (192.76.246.17)  226 ms  207 ms  242 ms
  5  icm-dc-1-S2/6-512k.icp.net (192.157.65.209)  320 ms  315 ms  393 ms
  6  icm-mae-e-H1/0-T3.icp.net (198.67.131.9)  372 ms  297 ms  354 ms
  7  mae-east (192.41.177.180)  456 ms  537 ms  401 ms
  8  borderx2-hssi2-0.Washington.mci.net (204.70.74.117)  529 ms  385 ms  340 ms
  9  core-fddi-1.Washington.mci.net (204.70.3.1)  437 ms  554 ms  581 ms
 10  core-hssi-3.NewYork.mci.net (204.70.1.6)  418 ms  547 ms  492 ms
 11  core-hssi-3.Boston.mci.net (204.70.1.2)  453 ms  595 ms  724 ms
 12  border1-fddi-0.Boston.mci.net (204.70.2.34)  789 ms  404 ms  354 ms
 13  nearenet.Boston.mci.net (204.70.20.6)  393 ms  323 ms  346 ms
 14  mit3-gw.near.net (192.233.33.10)  340 ms  465 ms  399 ms
 15  umass1-gw.near.net (199.94.201.66)  557 ms  316 ms  369 ms
 16  lgrc-gw.gw.umass.edu (192.80.83.1)  396 ms  309 ms  389 ms
 17  cs-gw.cs.umass.edu (128.119.44.1)  276 ms  490 ms  307 ms
 18  gaia.cs.umass.edu (128.119.40.186)  335 ms  317 ms  350 ms
```
ARP: IP address → MAC address

- for broadcast networks like Ethernet, token ring, ...
- if MAC address unknown, send ARP request and hold on to packet
- ARP request → broadcast: sender IP, MAC; target IP, MAC
- all machines update their cache → efficiency, allow change of interface
- ARP reply → requestor: reverse source/target; fill in source MAC
- directly on Ethernet, not IP!
- cache ARP replies; drop after 20 minutes
ARP example

```
arp -a
Net to Media Table
Device  IP Address    Mask    Flags    Phys Addr
------- ----------- --------------- ----- ---------------le0 hamlet 255.255.255.255 08:00:09:70:7d:16
le0 gaia 255.255.255.255 08:00:20:20:07:03
le0 pern 255.255.255.255 08:00:20:20:75:3c
le0 kite 255.255.255.255 08:00:09:92:0d:d1
le0 condor 255.255.255.255 08:00:20:1c:95:ed
```
RARP: MAC → IP address

- determine IP address at boot for diskless workstations
- remember: MAC address is unique and permanent
- host broadcasts RARP request (with its own MAC address)
- RARP server responds with reply
- allows third-party queries
- want several servers for reliability
Proxy ARP

- extend network: router fronts for H3, H4
- router answers ARP requests for H3, H4 from H1, H2 with its own hardware address
- assumes trusting relationship
- only needs to be added to single router
- only works for broadcast networks
Transport Layer: UDP and TCP

- UDP service = IP service + checksum + *ports*

- TCP service = UDP service + flow control + congestion control + sequenced, reliable byte stream

- TCP for multimedia:
  - loss recovery delay (RTT + $\epsilon$)
  - windowed flow/congestion control $\Rightarrow$ variable bandwidth
  - no multicast
Internet Domain Names
The Internet Domain Name System (DNS)

- hierarchical, dot-separated names
- multi-level delegation
- by country and by type of organization
- needs to be overhauled (59% of all domains = .com!)

Global top-level domains (gTLDs):

- 2 letters: countries
- 3 letters: independent of geography (except edu, gov, mil)
<table>
<thead>
<tr>
<th>domain</th>
<th>usage</th>
<th>example</th>
<th>hosts (7/97)</th>
</tr>
</thead>
<tbody>
<tr>
<td>com</td>
<td>business (global)</td>
<td>research.att.com</td>
<td>4501039</td>
</tr>
<tr>
<td>edu</td>
<td>U.S. 4 yr colleges</td>
<td>cs.columbia.edu</td>
<td>2942714</td>
</tr>
<tr>
<td>net</td>
<td>network provider</td>
<td>nis.nsf.net</td>
<td>2164815</td>
</tr>
<tr>
<td>mil</td>
<td>U.S. military</td>
<td>arpa.mil</td>
<td>542295</td>
</tr>
<tr>
<td>gov</td>
<td>U.S. non-military gov’t</td>
<td>whitehouse.gov</td>
<td>418576</td>
</tr>
<tr>
<td>org</td>
<td>non-profit orgs (global)</td>
<td><a href="http://www.ietf.org">www.ietf.org</a></td>
<td>327148</td>
</tr>
<tr>
<td>us</td>
<td>U.S. geographical</td>
<td>ietf.cnri.reston.va.us</td>
<td>825048</td>
</tr>
<tr>
<td>uk</td>
<td>United Kingdom</td>
<td>cs.ucl.ac.uk</td>
<td>878215</td>
</tr>
<tr>
<td>de</td>
<td>Germany</td>
<td>fokus.gmd.de</td>
<td>875631</td>
</tr>
</tbody>
</table>
Example

server 128.9.0.107
Default Server: b.root-servers.net
Address: 128.9.0.107

> erlang.cs.columbia.edu
Server: b.root-servers.net
Address: 128.9.0.107

Name: erlang.cs.columbia.edu
Served by:
- CUNIXD.CC.COLUMBIA.edu
  128.59.35.142
  COLUMBIA.edu
- DNS2.ITD.UMICH.edu
  141.211.125.17
  COLUMBIA.edu
New gTLDs

Proposed for 1998 by CORE, but fate uncertain:

- .firm for businesses, or firms
- .shop for businesses offering goods to purchase
- .web WWW activities
- .arts cultural and entertainment
- .rec recreation/entertainment
- .info information services
- .nom individual or personal nomenclature
DNS Issues

NTIA white paper, June 1998:

- no competition in registration cost
- registries vs. registrars
- additional TLDs?
- trademark disputes
- access to NSI database
- domain name speculators
- scarcity of good names
- IANA → non-profit oversight body

http://www.ntia.doc.gov/ntiahome/domainname/dnsdrft.htm
Domain Name Resolution

- hierarchy of redundant servers with time-limited cache
- each server knows the 13 root servers a.root-servers.net
- each root server knows gTLDs and refers queries to those
- each domain has $\geq 2$ servers, often widely distributed
- also: mailbox translation
- *almost* a distributed database
Internet Growth
Architectural Principles
Architectural principles of the Internet

RFC 1958:

- large existing infrastructure ➔ backward compatibility
- “The goal is connectivity, the tool is the Internet Protocol, and the intelligence is end to end rather than hidden in the network”
- End-to-end functions can best be realised by end-to-end protocols.
- Nobody owns the Internet, there is no centralized control, and nobody can turn it off.

Design principles:

- Heterogeneity is inevitable and must be supported by design.
- If there are several ways of doing the same thing, choose one.
• All designs must scale readily to very many nodes per site and to many millions of sites.

• Performance and cost must be considered as well as functionality.

• Keep it simple.

• In many cases it is better to adopt an almost complete solution now, rather than to wait until a perfect solution can be found.

• Avoid options and parameters whenever possible. Configure them automatically.

• Be strict when sending and tolerant when receiving.

• Be parsimonious with unsolicited packets.

• Circular dependencies must be avoided.

• Objects should be self describing (include type and size).

• All specifications should use the same terminology and notation, and the same bit- and byte-order convention.
- Nothing gets standardised until there are multiple instances of running code.

Names and addresses:

- No hardcoded addresses.
- Single naming structure.
- Names should be case-insensitive ASCII.
- Addresses must be unambiguous.
- Upper-layer protocols must be able to identify end points unambiguously.
Internet Design Principles

• small, single-function protocols ➞ re-invent similar functionality

• lower-layer protocols: attention to processing efficiency ➞ word alignment, fixed-length fields: CLNP vs. IPv4/IPv6

• incremental: can build small applications easily ➞ embedding of email, HTTP, whois, …
upper-layer protocols are text-based

+ “the universal telnet protocol simulator”
+ lots of tools (≈ Unix) such as Tcl, VisualBasic, Perl, . . .
  – space overhead
  – parsing costs, but usually insignificant
  – difficult to represent nested data structures
+/− typically ASCII ⇒ internationalization required

but: most upper layer protocols (including NFS, ftp, HTTP, SMTP, RTSP, SIP, . . .) are basically RPC
Soft state vs. hard state

Need to coordinate views of the world in a distributed system

**hard state:** traditional telecom state is precious:
- handshake, with retransmission on timeout
- explicit teardown of state

**soft state:** “optimistic” approach state is restorable
- send periodic messages to set up or refresh state
- no explicit teardown (except for efficiency) time-out state
- works even if routers, hosts fail or suffer amnesia
- sometimes allows selective state maintenance
- simpler? higher state overhead? recovery time?