COMPUTER NETWORKS
CS 45201
CS 55201

CHAPTER 4
Internetworking

H. Peyravi and P. Farrell
Department of Computer Science
Kent State University
Kent, Ohio 44242
farrell@cs.kent.edu
http://www.cs.kent.edu/~farrell

Fall 2001
Contents

- Internetworking
- Global Internet
- Route Propagation
- Next Generation IP (IPv6)
- Internet Multicasting
- Host Names (DNS)
Internetworking

Definition: Logical network built from multiple physical networks.

Hn = Host
Rn = Router

Network 1 (Ethernet)

Network 2 (Ethernet)

Network 3 (Token Ring)

Network 4 (point-to-point)
IP Service Model

- Packet Delivery Model
- Global Addressing Scheme

IP Datagram Packet Delivery Model

- Connection-less (datagram-based)
- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time
- TCP/IP standards given by RFCs (Request-for-Comments)
### Datagram format

- **Version (4):** currently 4
- **Hlen (4):** number of 32-bit words in header
- **TOS (8):** type of service (not widely used)
- **Length (16):** number of bytes in this datagram
- **Ident (16):** used by fragmentation
- **Flags/Offset (16):** used by fragmentation
- **TTL (8):** number of hops this datagram has traveled
- **Protocol (8):** demux key (TCP=6, UDP=17)
- **Checksum (16):** of the header only
- **DestAddr & SrcAddr (32)**

---

<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>19</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>HLen</td>
<td>TOS</td>
<td>Length</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identi</td>
<td>Flags</td>
<td>Offset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td>Protocol</td>
<td>Checksum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SourceAddr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DestinationAddr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options (variable)</td>
<td>Pad (variable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fragmentation and Reassembly

- Each network has some MTU (Max Transmission Unit)
- Strategy
  - fragment when necessary (MTU < Datagram)
  - IP packet needs to fit in payload part of frame
  - use CS-PDU (not cells) for ATM $\rightarrow$ CS: convergence sublayer
  - try to avoid fragmentation at source host
  - refragmentation is possible at routers etc.
  - fragments are self-contained datagrams
  - all fragments contain same value in ident field
  - each fragment is re-encapsulated in frame
  - delay reassembly until destination host
  - do not recover from lost fragments
- Example
### Unfragmented Packet

<table>
<thead>
<tr>
<th>Start of header</th>
<th>ident = x</th>
<th>0</th>
<th>offset = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest of header</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1400 data bytes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### First Fragment

<table>
<thead>
<tr>
<th>Start of header</th>
<th>ident = x</th>
<th>1</th>
<th>offset = 512</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest of header</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512 data bytes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Second Fragment

<table>
<thead>
<tr>
<th>Start of header</th>
<th>ident = x</th>
<th>0</th>
<th>offset = 1024</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest of header</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>376 data bytes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Last Fragment

<table>
<thead>
<tr>
<th>Start of header</th>
<th>ident = x</th>
<th>0</th>
<th>offset = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest of header</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>512 data bytes</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Global Addresses

Properties

- globally unique
- hierarchical: network + host

Format

- **Class A**
  - 7 bits for network
  - 24 bits for host

- **Class B**
  - 14 bits for network
  - 16 bits for host

- **Class C**
  - 21 bits for network
  - 8 bits for host

Dot notation

- 10.3.2.4
- 128.96.33.81
- 192.12.69.77
# Datagram Forwarding

## Strategy

- every datagram contains destination's address
- if directly connected to destination network, then forward to host
- if not directly connected to destination network, then forward to some router
- forwarding table maps network number into next hop
- each host has a default router
- each router maintains a forwarding table

### Example (router R2)

<table>
<thead>
<tr>
<th>Network Number</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R3</td>
</tr>
<tr>
<td>2</td>
<td>R1</td>
</tr>
<tr>
<td>3</td>
<td>interface 1</td>
</tr>
<tr>
<td>4</td>
<td>interface 0</td>
</tr>
</tbody>
</table>
**Address Translation**

- Map IP addresses into physical addresses
  - destination host
  - next hop router
- Techniques
  - encode physical address in host part of IP address
  - table-based
- ARP (Address Resolution Protocol)
  - table (cache) of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed
- Notes
  - table entries timeout in about 10 minutes
  - update table with source when you are the target
  - refresh table if already have an entry
  - do not refresh table entries upon reference
  - do not add to table if not there and not target
Request format

<table>
<thead>
<tr>
<th>HardwareType=1</th>
<th>ProtocolType=0x0800</th>
</tr>
</thead>
<tbody>
<tr>
<td>HLEN=48</td>
<td>PLEN=32</td>
</tr>
<tr>
<td>SourceHardwareAddr</td>
<td>SourceProtocolAddr</td>
</tr>
<tr>
<td>SourceProtocolAddr</td>
<td>TargetHardwareAddr</td>
</tr>
<tr>
<td>TargetHardwareAddr</td>
<td>TargetProtocolAddr</td>
</tr>
</tbody>
</table>

- **HardwareType**: type of physical network (e.g., Ethernet)
- **ProtocolType**: type of higher layer protocol (e.g., IP)
- **HLEN & PLEN**: length of physical and protocol addresses
- **Operation**: request or response
- **Source/Target Physical/Protocol addresses**
Chapter 4: Internetworking

The Address Resolution Protocol, ARP

- Every machine on the Internet has one (or more) IP addresses.

- These addresses cannot be used for sending packets because data link layer does not understand IP addresses.

- LAN addresses are 48-bit and they know nothing about 32-bit IP addresses.

- How do IP addresses get mapped onto the data link layer address.
  - The upper layer software builds a packet with 192.31.65.5 as destination address and gives it to IP software to transmit.
  - The IP software can look at the address to see whether it is local, but it needs the corresponding Ethernet address.
  - One solution is to have a configuration file to map IP addresses to LAN addresses, but it is not practical for large number of hosts.
  - Another solution is to broadcast a packet onto the LAN asking who own the address 192.31.65.5?
  - The destination responds with its LANS address and the source learns the destination LAN address. This is called ARP.

- At this point the IP software builds an Ethernet frame addressed to the destination, puts the IP packet in the payload field and dumps it on to the Ethernet. The destination detects the frame, reorganize it. The Ethernet driver extract the IP packet from the payload and passes it to the IP software.
One optimization is to cache the address in case it will be needed soon.

Another optimization is to have every machine broadcast its mapping when it boots.

For remote LAN access, ARP fails, there are two solutions:

1. The local router could be configured to respond to ARP request to other LANs. This is called *proxy ARP*.
2. Have the source send all nonlocal traffic to a default LAN when ARP fails locally.
The Reverse Address Resolution Protocol, RARP

- ARP solves the problem of finding an Ethernet address from an IP address.
- Sometime the reverse problem has to solved; finding the IP address from the Ethernet address (booting a diskless workstation).
- This protocol allows a newly-booted workstation to broadcast its 48-bit Ethernet address and ask: "Does anyone out there know my IP address?"
- The RARP server sees this request, looks up the Ethernet address in its configuration file and sends back the corresponding IP address.
- A RARP server is needed on each network.
- An alternative protocol is called BOOTP (bootstrap) protocol that uses UDP (user data protocol) messages which are forwarded over routers.
- It also provide a diskless workstation with additional information, including the IP address of the file server holding the memory image, the IP address of the default router.
## Internet Control Message Protocol

- **Echo (ping)**
- **Redirect (from router to source host)**
- **Destination unreachable (protocol, port, or host)**
- **TTL exceeded (so datagrams don’t cycle forever)**
- **Checksum failed**
- **Reassembly failed**
- **Cannot fragment**

The Internet operations is closely monitored by the routers.

All unexpected events are reported by the ICMP (Internet Control Message Protocol), which is used to test the Internet.

About a dozen types of ICMP messages are defined.

### The principal ICMP message types

<table>
<thead>
<tr>
<th>Message type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination unreachable</td>
<td>Packet could not be delivered</td>
</tr>
<tr>
<td>Time exceeded</td>
<td>Time to live hit 0</td>
</tr>
<tr>
<td>Parameter problem</td>
<td>Invalid header field</td>
</tr>
<tr>
<td>Source quench</td>
<td>Choke packets</td>
</tr>
<tr>
<td>Redirect</td>
<td>Teach a router about geography</td>
</tr>
<tr>
<td>Echo request</td>
<td>Ask a machine if it is alive</td>
</tr>
<tr>
<td>Echo reply</td>
<td>Yes, I am alive</td>
</tr>
<tr>
<td>Timestamp request</td>
<td>Same as Echo request, but with timestamp</td>
</tr>
<tr>
<td>Timestamp reply</td>
<td>Same as Echo reply, but with timestamp</td>
</tr>
</tbody>
</table>
Dynamic Host Configuration Protocol (DHCP)

- Allow automatic configuration from DHCP server of:
  - IP address (can be static in table indexed by hardware address OR dynamically assigned from pool (periodic renewal required))
  - default router

- DHCP server identification
  - host broadcasts DHCP DISCOVER message
  - carried in UDP packet
  - if DHCP server local replies
  - otherwise relayed by relay agent
**Virtual Networks & Tunnels**

- implement VPN using IP
- allow only members of VPN to access
- tunnel is established between routers
- router encapsulates packet in IP datagram with destination address of router at other end of tunnel

![Diagram of network with routers R1 and R2 connected through an internetwork](image)

<table>
<thead>
<tr>
<th>Network Number</th>
<th>NextHop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Interface 0</td>
</tr>
<tr>
<td>2</td>
<td>Interface 0</td>
</tr>
<tr>
<td>Default</td>
<td>Interface 1</td>
</tr>
</tbody>
</table>

**Why?**
- Security
- connect routers with enhanced capability (multicast MBone)
- carry non-IP packets (IPX)
- make machines at both ends seem like on same network
Chapter 4: Internetworking

Global Internet

Internet Structure

Recent Past

Stanford
BARNET Regional
Berkeley
PARC
NCAR
Westnet Regional
UA
UNM
UNL
KU
ISU
NSFNET Backbone

Today

Service Provided Backbone

Stanford
BARNET Regional
Berkeley
PARC
NCAR
Westnet Regional
UA
UNM
UNL
KU
ISU
MidNet Regional
Scalability Issues

IP “hides” hosts in address hierarchy, but...

- Inefficient use of address space
  - class C network with 2 hosts \((2/255 = 0.78\% \text{ efficient})\)
  - class B network with 256 hosts \((256/65535 = 0.39\% \text{ efficient})\)

- Too many networks
  - today’s Internet has tens of thousands of networks
  - routing tables do not scale
  - route propagation protocols do not scale
Subnetting

- Add another level to address/routing hierarchy: subnet
- Subnet masks define variable partition of host part
- Subnets visible only within site

<table>
<thead>
<tr>
<th>Network Number</th>
<th>Host Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Class B address

| 111111111111111111111111 | 00000000 |

Subnet Mask (255.255.255.0)

<table>
<thead>
<tr>
<th>Network Number</th>
<th>Subnet ID</th>
<th>Host ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Subnetted Address
**Subnet Example**

Consider the following configuration:

- **Subnet Mask**: 255.255.255.128
- **Subnet number**: 128.96.34.0

Subnet Mask: 255.255.255.128
Subnet number: 128.96.34.0

Subnet Mask: 255.255.255.0
Subnet number: 128.96.33.0

**with the following routing table for R1**

<table>
<thead>
<tr>
<th>Subnet Number</th>
<th>Subnet Mask</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.96.34.0</td>
<td>255.255.255.128</td>
<td>interface 0</td>
</tr>
<tr>
<td>128.96.34.128</td>
<td>255.255.255.128</td>
<td>interface 1</td>
</tr>
<tr>
<td>128.96.33.0</td>
<td>255.255.255.0</td>
<td>R2</td>
</tr>
</tbody>
</table>
How to Send a Packet

- The host performs bitwise & between its subnet mask and the destination IP address
  - If the result is its subnet number \( \Rightarrow \) local subnet
  - If not, the packet is sent to a router
- Suppose H1 is sending to H2 where H1 and H2 belong to the same subnet
  
  \[
  H2 = 10000000 01100000 00100010 10001011 = 128.96.34.139 \\
  M1 = 11111111 11111111 11111111 10000000 = 255.255.255.128 \\
  \]

  \[
  D1 = 10000000 01100000 00100010 10000000 = 128.96.34.128 \\
  \]

  This does not match the subnet number (128.96.34.128) for H1 (128.96.34.0)
- H1 knows that H2 is on a different network \( \Rightarrow \) it sends it to R1
- R1 has more work to do when Subnetting
- Table entries now (subnet number, subnet mask, next hop)
- R1 ANDs the H2 IP address (28.96.34.139) with the subnet mask for each entry e.g. (255.255.255.128)
  - The result (128.96.34.128) is compared with the network entry (128.96.34.0) \( \Rightarrow \) no match
  - But it is matched with the second entry entry of the table
  - Forwarded to the interface 1 connected to R1
### Forwarding Algorithm

D = destination IP address
for each entry \(\langle \text{SubnetNum}, \text{SubnetMask}, \text{NextHop}\rangle\)
  D1 = SubnetMask & D
  if D1 = SubnetNum
    if NextHop is an interface
      deliver datagram directly to destination
    else
      deliver datagram to NextHop (a router)

Notes
- Would use a default router if nothing matches
- Not necessary for all ones in subnet mask to be contiguous
- Can put multiple subnets on one physical network
- Subnets not visible from the rest of the Internet
Classless InterDomain Routing, CIDR: Superneting

- IP has been extremely successful with its exponential growth, but it is running out of address space.
- In principle, over 2 billion addresses exist, but in practice millions of them are wasted by classes.
- For most organizations, class A with 16 million addresses is too big and class C with 256 addresses is too small.
- A class B network with 65,536, is just right.
- Studies have shown that more than half of all class have fewer than 50 hosts.
- Another problem is table explosion. Routers do not have to know about all the host, but they know about other networks.
- Having half a million class C networks, every router would require a table with half million entries.
- Another problem is the complexity of various algorithms for management of the tables that grows faster than linear.
- Design choices made a decade ago with only 1000 network is far from optimal.
- The routing table problem can be solved by going to a deeper hierarchy (like telephone), but it requires more than 32-bit for IP addresses.
- Most solutions solve one problem but create new ones.
- One solution currently being implemented is CIDR (Classless InterDomain Routing)
The basic idea behind CIDR is to allocate the remaining class C networks (almost 2 million) in variable-sized blocks.

- If a site needs 2000 addresses, it is given 2048 addresses (8 contiguous class C networks), and not a full class B address.

In addition to using contiguous blocks, the allocation rule were also changed. The world was partitioned into four zones.

- Europe: 194.0.0.0 to 195.255.255.255
- North America: 198.0.0.0 to 199.255.255.255
- Central and South America: 200.0.0.0 to 201.255.255.255
- Asia and Pacific: 202.0.0.0 to 203.255.255.255

Each region was given 32 million addresses, with another 320 million class C addresses from 204.255.255.255 to 223.255.255.255 reserved for future use.

Within each block allocate sub-block to ISP. ISP then allocates to customers.

- Restrict block sizes to powers of 2
- Use a bit mask (CIDR mask) to identify block size
- All routers must understand CIDR addressing
Route Propagation

- Impose a second hierarchy on the network that limits what routers talk to each other.
- The first hierarchy is the address hierarchy that governs how packets are forwarded.
- Idea: Know a smarter router
  - hosts know a local router
  - local routers know site routers
  - site routers know core router
  - core routers know everything

- Autonomous System (AS)
  - corresponds to an administrative domain
  - examples: University, company, backbone network
  - assign each AS a 16-bit number

- Two-level route propagation hierarchy
  - interior gateway protocol (each AS selects its own)
  - exterior gateway protocol (Internet-wide standard)
Popular Interior Gateway Protocols

- **RIP: Route Information Protocol**
  - developed for XNS
  - distributed with Unix
  - distance-vector algorithm
  - based on hop-count

- **OSPF: Open Shortest Path First**
  - recent Internet standard
  - uses link-state algorithm
  - supports load balancing
  - supports authentication
EGP: Exterior Gateway Protocol

Overview

- designed for tree-structured Internet
- concerned with reachability, not optimal routes

Protocol messages

- neighbor acquisition: one router requests that another be its peer; peers exchange reachability information
- neighbor reachability: one router periodically tests to see if the other router is still reachable; exchange HELLO/ACK messages;
- routing updates: peers periodically exchange their routing tables (distance-vector)
**BGP-4: Border Gateway Protocol**

Assumes the Internet is an arbitrarily interconnected set of AS's. Define *local traffic* as traffic that originates at or terminates on nodes within an AS, and *transit traffic* as traffic that passes through an AS, we can classify AS's into three types:

- **Stub AS**: an AS that has only a single connection to one other AS; such an AS will only carry local traffic.
- **Multihomed AS**: an AS that has connections to more than one other AS, but refuses to carry transit traffic.
- **Transit AS**: an AS that has connections to more than one other AS, and is designed to carry both transit and local traffic.

Each AS has:

- One or more border routers (gateways)
- One BGP *speaker* that advertises:
  - local networks
  - other reachable networks (transit AS only)
  - gives *path* information
- BGP advertises complete paths as enumerated lists of ASs to reach network
Speaker for AS 2 advertises reachability to P and Q

Network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS 2.

Speaker for backbone network then advertises

Networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path ⟨AS 1, AS 2⟩.

Speaker can also cancel previously advertised paths

BGP prevents looping by carrying complete path

As numbers unique 16 bit numbers assigned by a central authority
Next Generation IP (IPv6)

- CIDR may last for a few years, but everyone realizes that the days of IP in its current form (IPv4) are numbered.
- Besides growing number of Mobile stations, it may not be long before every TV set is an Internet node, producing billion machines.
- IETF has started working on a new version of IP, which would never run out of addresses, and solving a variety of other problems.

Its major goals were to

- Supports billions of hosts, even with inefficient address space allocation.
- Reduce the size of the routing tables.
- Simplify the protocol to allow routers to process packets faster.
- Provide better security (authentication and privacy) than current IP.
- Pay more attention to type of service, particularly for real-time data.
- Aid multicasting by allowing scopes to be specified.
- Make it possible for a host to roam without changing its address.
- Allow the protocol to evolve in the future.
- Permit the old and new protocols to coexist for years.
Chapter 4: Internetworking

Next Generation IP (IPv6)

Major Features

- 128-bit addresses ($3.4 \times 10^{38}$ addresses)
- Autoconfiguration
- Multicast
- Real-time service
- Authentication and security
- Auto-configuration
- End-to-end fragmentation
- Protocol extensions

The Main IPv6 Header

<table>
<thead>
<tr>
<th>Version</th>
<th>Priority</th>
<th>Flow label</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Payload length</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Source Address</td>
</tr>
</tbody>
</table>

|         |          | Destination Address | 16 bytes |          |

The IPv6 fixed header
■ 40-byte "base" header
■ Extension headers (fixed order, mostly fixed length)
  ► fragmentation
  ► source routing
  ► authentication and security
  ► other options
■ The Version field is always 6 for IPv6 and 4 for IPv4.
■ The Priority field is used to distinguish between packets whose
  sources can be flow controlled and those that cannot.
  ► 0-7 are capable of slowing down in the event of congestion. lower
    numbers are less important (1 for news, 4 for FTP, 6 for
    Telnet, .. ).
  ► 8-15 are for real-time traffic(audio, video).
■ The Flow label is still experimental, but will be used to allow to set
  up particular properties between source and destination.
■ The Payload length field tells how many bytes follow the 40-byte
  header
■ The Next header field tells which of the (currently) six extension
  headers follows this one. If this header is last IP header the next
  header would be transport layer (TCP, UDP, etc.)
■ The Hop limit field is used to keep packets from living forever. In
  IPv4 this was time.
■ The Source and Destination fields are fixed-length 16-byte
  addresses.
IPv6 has introduced the concept of an (optional) extension header that can be used to provide extra information.

<table>
<thead>
<tr>
<th>Extension header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hop-by-hop options</td>
<td>Miscellaneous information for routers</td>
</tr>
<tr>
<td>Routing</td>
<td>Full or partial route to follow</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>Management of datagram fragments</td>
</tr>
<tr>
<td>Authentication</td>
<td>Verification of the sender’s identity</td>
</tr>
<tr>
<td>Encrypted security payload</td>
<td>Information about encrypted contents</td>
</tr>
<tr>
<td>Destination options</td>
<td>Additional information for the destination</td>
</tr>
</tbody>
</table>
IPv6 Addresses

- Classless addressing/routing (similar to CIDR)
- Notation: x:x:x:x:x:x:x:x (x = 16-bit hex number)
  - one set of contiguous 0s are compressed: 47CD::A456:0124
  - Addresses with 80 0s are reserved for IPv4.
  - IPv4 compatible IPv6 address: ::128.42.1.87
  - IPv4 mapped IPv6 address: ::00FF:128.42.1.87

- Address assignment - a number of techniques
  - Aggregable Global Unicast Addresses - like classless IPv4
  - provider-based: Separate prefixes are assigned to different network providers.
  - Problems: changing providers, two providers
  - geographic
  - Autoconfiguration option - use interface ID that is unique on link + address prefix (from router?)
### Address prefixes

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 0000</td>
<td>Reserved</td>
</tr>
<tr>
<td>0000 0001</td>
<td>Reserved for NSAP (OSI)</td>
</tr>
<tr>
<td>0000 001</td>
<td>Reserved for IPX</td>
</tr>
<tr>
<td>001</td>
<td>Aggregateable Global Unicats Addresses</td>
</tr>
<tr>
<td>1111 1110 10</td>
<td>Link local use addresses</td>
</tr>
<tr>
<td>1111 1110 11</td>
<td>Site local use addresses</td>
</tr>
<tr>
<td>1111 1111</td>
<td>Multicast addresses</td>
</tr>
<tr>
<td>others</td>
<td>Unassigned (see Table 4.11)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>m</th>
<th>n</th>
<th>o</th>
<th>p</th>
<th>125−m−n−o−p</th>
</tr>
</thead>
<tbody>
<tr>
<td>010</td>
<td>Registry ID</td>
<td>Provider ID</td>
<td>Subscriber ID</td>
<td>Subnet ID</td>
<td>Interface ID</td>
</tr>
</tbody>
</table>
Internet Multicasting

- IP-V4 supports multicasting using class D addresses
- demonstrated with MBone
- 28 bits are available for identifying groups, so over 250 million groups can exist at the same time.
- problem is making it scale
- When a process sends a packet to a class D address, it is delivered to all the members of the group addressed, but no guarantees are given
- Two kinds of group addresses are supported
  - Permanent addresses: are always there; does not have to be set up
  - Temporary addresses: must be created before they can be used
- Multicasting is implemented by special multicast routers
- Once a minute, each multicast router sends a hardware multicast to the hosts on its LAN asking them to report back on the group their processes currently belong to
- Each host sends back responses for all the class D addresses it is interested in
- The query and response packets use a protocol called IGMP (Internet group management protocol)
- Router to host handled by network level multicast
Link-State Multicast

- Routers know entire topology
- Each host on a LAN periodically announces the groups it belongs to (IGMP).
- Augment update message (or Link state packet LSP) includes set of groups that have members on a particular LAN.
- Each router uses Dijkstra’s algorithm to compute shortest-path multicast spanning tree for each source/group pair.
- Each router caches tree for currently active source/group pairs.
Example
Distance-Vector Multicast

Reverse Path Broadcast (RPB)

- Each router already knows that shortest path to destination S goes through router N.
- When a router receives multicast packet from S, forwards it on all outgoing links (except the one on which the packet arrived), iff packet arrived from N.
- Eliminate duplicate broadcast packets by only letting “parent” for LAN (relative to S) forward
  - shortest path to S (learn via distance vector)
  - smallest address to break ties

Reverse Path Multicast (RPM)

- Goal: Prune networks that have no hosts in group G
- Step 1: Determine if LAN is a leaf with no members in G
  - leaf if parent is only router on the LAN
  - determine if any hosts are members of G using IGMP
- Step 2: Propagate “no members of G here” information
  - augment ⟨Destination, Cost⟩ update sent to neighbors with set of groups for which this network is interested in receiving multicast packets.
  - only happens when multicast address becomes active.
Protocol Independent Multicast (PIM)

- Solve scaling problems
- Sparse mode: few receivers
- Dense mode: many receivers

Sparse Mode PIM
- Routers explicitly Join/Prune (leave)
- Send to Rendezvous Points (RPs)
- Create Multicast Forwarding Tree
  - shared (per group) (see a, b)
  - source specific (per sender - only created if data rate warrants) (see c, d)
(a) R4 sends Join to RP and joins shared tree
(b) R5 joins shared tree
(c) RP builds source specific tree to R1 by sending Join to R1
(d) R4 and R5 build source specific tree to R1 by sending Joins to R1

RP = Rendezvous point
- - - - Source-specific tree for source R1
Internet Multicasting

R1 → R2 → R3 → RP → R4

Host → R1 → R3

G

RP

R2

R3

R4

R5

CS 4/55201: Computer Networks Fall 2001
Host Names (DNS)

**Overview**

- **Names versus Addresses**
  - names are variable length, mnemonic, easy for humans to remember
  - addresses are fixed length, tied to routing, and easy for computers to process

- **Name Space**
  - defines set of possible names
  - flat versus hierarchical
  - consists of a set of name to value *bindings*

**Example hierarchy**

```
Name Server (cs.arizona.edu: 192.12.69.5)
  ↕
  ↕
User (user@cs.arizona.edu)
  ↕
Mail Program
  ↕
TCP
  ↕
IP
```

```
1. User requests mail
2. User sends request to Name Server
3. Name Server responds with IP address
4. TCP establishes connection
5. Data is sent over TCP/IP
```
Example name: cheltenham.cs.arizona.edu

Name Servers

Partition hierarchy into zones

Each zone implemented by two or more name servers, a primary name server and one or more secondary name servers
Each name server maintains a collection of resource records \( \langle \text{Name}, \text{Value}, \text{Type}, \text{Class}, \text{TTL} \rangle \)

- **Name/Value**: not necessarily host names to IP addresses

- **Type**
  - NS: the Value field gives the domain name for a host running a name server that knows how to resolve names within the specified domain.
  - CNAME: the Value field gives the canonical name for a particular host; it is used to define aliases.
  - MX: the Value field gives the domain name for a host running a mail server that accepts messages for the specified domain.

- **Class**: allow other entities to define types

- **TTL**: how long the resource record is valid
Strategies

- forward
- iterative
- recursive

Clients know local name server

Local server queries other servers

Authoritative and non-authoritative answers

- need to know root at only one place (not each host)
- site-wide cache
Many Internet users have portable computers

Every IP address contains 3 fields: class, network number, and host number

Given the class and network number, routers all over the world know how to get to the LAN

By moving a machine to another LAN, the routers would still send packets to the old destination

Giving a new IP address for a new location is not realistic since a large number of people, programs, databases should be informed about this change

Another solution is to use the complete IP address (rather than class and network number) which requires routers to have millions of table entries (astronomical cost to the Internet)

IETF formulated a number of goals:

- Each mobile host must be able to use its home IP address anywhere
- Software change to the fixed hosts were not permitted
- Changes to router software and tables were permitted
- Most packets for mobile host should not make detours
- No overhead should be incurred when a mobile host is at home.

The solution was chosen foreign and home agents.
Home Agent (HA) - router on home network
- Mobile Host (M) registers initially with it

Foreign Agent (FA)
- M registers with it when moved
- gives home IP address and hardware address
- FA contacts HA giving care-of address

Issues to be resolved

- How does HA intercept packets for M?
  - uses proxy ARP

- How does HA deliver to FA?
  - use tunneling

- How does FA deliver to M?
  - using hardware address

- M can be its own FA if it can dynamically acquire an IP address (e.g. using DHCP)