SE 4C03 Winter 2003

04 Internet Protocol (IP)

Instructor: W. M. Farmer

Revised: 16 January 2003

Internet Protocol (IP)

- IP provides a connectionless packet delivery service between internet hosts
  - **Connectionless**: packets bounce across a sea of computers
  - **Best-effort delivery**: service is designed to deliver every packet
  - **Unreliable**: packet delivery is not guaranteed
- IP defines a mechanism consisting of:
  - A basic unit of data transfer called an **internet** or **IP datagram**
  - Software for routing datagrams
  - Rules for how hosts (and routers) should process datagrams

IP Addresses

- There are two Internet naming systems:
  1. The primary system is the **internet address system** which uses binary **IP addresses**
  2. The secondary system is the **domain name system (DNS)** which uses natural language **DNS names**

- IP addresses are 32-bit integers
  - Composed of four 8-bit octets
  - Represented as four integers, usually in base 2 or base 10, separated by dots

    base 2: 11000111.00010001.00101000.11010010

    base 10: 199.17.40.210

IP Address Assignment

- IP addresses are assigned to **network interfaces**, **not hosts**
  - A host with one network interface is assigned an IP address by default
- A network interface is normally assigned a unique IP address
  - In practice, an interface may be assigned more than one address
  - In some rare cases, an interface may be assigned no address at all
  - In certain cases, different interfaces may have the same IP address

Class Networks

- IP addresses are organized into **class networks** to facilitate address assignment and packet routing
  - Class A: 0nnnnnnn.iiiiiiii.iiiiiiii.iiiiiiii
  - Class B: 10nnnnnn.nnnnnnnn.iiiiiiii.iiiiiiii
  - Class C: 110nnnnn.nnnnnnnn.nnnnnnnn.iiiiiiii
  - Class D: 1110bbbb.bbbbbbbb.bbbbbbbb.bbbbbbbb

- A class network is really a set of IP addresses and not a network

- The Internet Corporation for Assigned Names and Numbers (ICANN) is responsible for assigning class A, B, and C networks to organizations

- Each address in a class A, B, or C network is a pair (N, I) where N is its **network identification** and I is its a **interface (or host) identification**. n and i denote bits in the network and interface identifications, respectively.

Class Networks (cont.)

- By convention, the address in a class A, B, or C network whose interface bits are all 0 (e.g., 199.17.40.0) is the **network address** for the class

- By convention, the address in a class A, B, or C network whose interface bits are all 1 (e.g. 199.17.40.255) is the **(direct) broadcast address** for the class

- The **limited broadcast address** is 255.255.255.255

- The addresses in a class D network are for multicasting

- Some addresses in the above classes and the addresses of the form 1111bbbb.bbbbbbbb.bbbbbbbb.bbbbbbbb are reserved

The Loopback

- Each host running TCP/IP has a virtual interface called the **loopback interface** which is the only interface on a virtual network called the **loopback network**

- The network and interface addresses of the loopback are 127.0.0.0 and 127.0.0.1, respectively

Weaknesses of IP Address System

- Some hosts (e.g., multi-homed host) have more than one IP address
- The class networks are too rigid
- There are not enough IP addresses for future expansion

Subnets

- IP addresses are also organized into **subnets** to facilitate address assignment, network organization, and routing

- Each subnet is a set of addresses determined by:
  1. A **subnet address** (e.g., 199.17.35.96)
  2. A **subnet mask** (e.g., 255.255.255.240)

- Each address in a subnet is pair (S, I) where S is its **subnet identification** and I is its **interface (or host) identification**

- Special cases:
  - Set of all IP addresses
  - Class A, B, and C networks
  - Individual interface (or host) IP address

Subnet Conventions

- Usually, but not necessarily, the subnet identification of a class A, B, or C address is an extension of the network identification of the address
- Usually, but not necessarily, the subnet mask consists of a block of 1s followed by a block of 0s
- By convention, there is one subnet corresponding to each SPN
  - Each interface on the SPN is assigned the same subnet address and subnet mask

Address Resolution Problem

- High-level IP addresses are used for communication across an internet and are assigned independently of physical hardware addresses

- Low-level physical addresses are needed for physically delivering a packet to an interface on a network

- How are IP addresses mapped to physical addresses?
  - A solution is a function f that maps each IP address i to a physical address $f(i)$
  - The function must be changed as the internet changes
  - The function must be represented efficiently

Address Resolution Solutions

1. Physical addresses are encoded in IP addresses
    - Possible for proNET networks
    - Not viable for Ethernet

2. Each machine contains a table that represents the local part of an address resolution function
    - Awkward for Ethernet because physical addresses change when a host or interface is replaced

3. IP addresses are bound to physical addresses dynamically

Address Resolution Protocol (ARP)

- Used for dynamically binding an IP address to a physical address (especially on Ethernet networks)

- ARP process:
  1. A host h_A broadcasts a request for the physical address which resolves an IP address i
  2. The host h_B with the network interface having the address i sends a reply to h_A containing the physical address of the interface

- The results of ARP queries are kept in a cache on each host

- When a sender requests a physical address, it can include its physical address in the reply

Complications

- Several packets may simultaneously need to know the same physical address

- The host of the requested physical address may be down

- The cache may contain out-of-date bindings
  - When a host boots it can send a broadcast message informing the other computers on the network of its physical address

- Hosts may provide bogus address bindings

- At boot-time a diskless host knows its physical address but not its IP address
  - The host must get its IP address from a server on another computer

Reverse Address Resolution Protocol (RARP)

- Used for obtaining the IP address that is bound to a physical address

- RARP process:
  1. A host h_A broadcasts a request for the IP address which reversely resolves a physical address p
  2. The RARP servers which receive the request send replies back to the h_A containing the requested IP address

- Some scheme is needed to keep all the RARP servers from sending replies at the same time (and causing collisions on an Ethernet network)

ARP and RARP Messages

- ARP and RARP messages are **encapsulated** in a physical frame
  - ARP and RARP share the same message format
  - Type field says the data is an ARP or RARP message
  - The message itself is held in the data portion of the frame
- Each message has the following address fields:
  - Sender IP address
  - Sender physical address
  - Target IP address
  - Target physical address

Internet Datagrams

- Similar to physical network frames
  - Have header and data areas
  - Header contains source and destination IP addresses
- Unlike frames, datagrams are generally manipulated by software, not hardware
- Datagrams are transferred across networks in the data area of a physical frame
- Ideally, the whole datagram is **encapsulated** in the physical frame, but this cannot always be done

Fragmentation

- Each network has a **maximum transfer unit (MTU)**, the limit on how much data can be transferred per frame
  - The MTU for Ethernet is 1500 octets
  - The MTU for FDDI is about 4500 octets

- The maximum size allowed for a datagram is $2^{16} = 65,535$ octets

- **Fragmentation** occurs when the length of a datagram is bigger than the MTU for the network on which it is to be transferred
  - The host or router forwarding a datagram divides the datagram into **fragments** which have the same format as a full datagram
  - The fragments are not **reassembled** until they arrive that their final destination
  - Reassembly fails if any fragments are lost

Fields in Datagram Header Area

- **Version**, the version of IP used to create the datagram
- **Header length**, the length of the header area
- **Service type** specifies how the datagram should be handled
- **Total length** of the datagram
- **Identification** number of the datagram which is used, for example, to identify the fragments of the same datagram
- **Flags** contain information for controlling fragmentation (**do not fragment** and **more fragments** bits)
- **Fragment offset** is used to reassemble fragments

Fields in Datagram Header Area (cont.)

- **Time to live** holds the maximum number of routers the datagram is allowed to visit
- **Protocol** holds the type of the datagram
- **Header checksum** is used for checking the integrity of the datagram's header
- **Source IP address**
- **Destination IP address**
- **IP options** is an optional field that may be used for holding testing information

IP Options

- The field contains a string of IP options each consisting of a single octet option code, a single octet length field, and a variable length data field

- Example IP options:
  - **Record route** holds the list of IP addresses that the datagram has visited
  - **Source route** prescribes a route (represented as a partial or total list of IP addresses) through the internet for the datagram to take
  - **Timestamp** holds the list of IP addresses that the datagram visited with each address timestamped with the Universal Time when the datagram was handled

IP Routing

- **IP routing** is the process of choosing a path across an internet for a datagram to travel

- Routing may also be used in individual physical networks

- IP routing is performed by internet routers as well as by each host on the internet

- IP routing can be both static and dynamic
  - **Static routing** is configured by hand by system administrators
  - **Dynamic routing** is configured automatically by routing protocols

Kinds of Datagram Delivery

- There are three kinds of datagram delivery:

  1. **Immediate**: The datagram is delivered to the host that is processing the datagram
  2. **Direct**: The datagram is transmitted via a directly connected SPN to the destination host
  3. **Indirect**: The datagram is transmitted via a directly connected SPN to a "next hop" router which will forward the datagram

- For both direct and indirect delivery, the router needs to determine:

  1. The IP address of the next host h that is to receive the datagram
  2. The interface to the physical network on which h resides

Routing Tables 1

- Each host and router h contains an IP routing table
- Routing for direct and indirect delivery is usually done on the basis of the **network portion** of the datagram's destination address
- Each entry in the table for **direct delivery** is of the form (a, i) where:
  1. a is an IP network address of an SPN N directly connected to h
  2. i is the network interface that connects h to N

Routing Tables 2

- Each entry in the table for **indirect delivery** is of the form (a, r, i) where:
  1. a is an IP network address of an SPN N directly connected to h
  2. r is the IP address of the **next hop router** on N
  3. i is the network interface that connects h to N

- The table may contain a **default route** of the form $(*, r, i)$ where:
  1. $*$ matches any network address
  2. r is the IP address of the **default router** on an SPN N directly connected to h
  3. i is the network interface that connects h to N

Routing Tables 3

- The table may contain entries for **host-specific routes** of the form (a, r, i) where:
  1. a is a host IP address
  2. r is the IP address of the next hop router on an SPN N directly connected to h
  3. i is the network interface that connects h to N
- Notice that the table contains no information about SPNs (such as physical addresses) except for IP addresses and network interfaces

Basic Routing Algorithm

1. Extract destination IP address d from datagram

2. Deliver datagram to the host if d matches one of the IP addresses of the host (for incoming datagrams only)

3. Otherwise extract the destination network address d' from d

4. Forward the datagram as specified by the first entry in the host's routing table that matches d or d'

5. Otherwise declare a routing error

Special Cases

- Routing in single-homed hosts
  - Need to route outgoing datagrams
  - Usually should not route incoming datagrams
- Sending a datagram to the source host itself
  - Route the datagram to the loopback interface (which will cause the datagram to be added to the incoming datagram queue)
  - Route the datagram for direct delivery to one of the other local SPNs (which will cause the datagram to be redirected to the loopback interface)

Class Network Problem for Routing

- **Underlying assumption**: There is a one-to-one mapping between SPNs and class networks such that, if SPN N is mapped to class network C, then the address of each interface on N is a member of C

- This assumption is problematic because class networks are too rigid and too few

- Need a way of sharing a single class network of addresses among several SPNs

Solution 1: CPNs

- Use special routers to combine one or more SPNs into a **compound physical network (CPN)** that behaves like a SPN

- Transparent router scheme
  - Transparent routers manipulate IP datagrams
  - They lack the full status of an IP router

- Proxy ARP scheme
  - Proxy ARP routers manipulate physical frames
  - They allow ARP requests and replies to be sent from one SPN to another

Solution 2: Anonymous Networks

- The interfaces on a point-to-point network are not assigned IP addresses

- The interface hardware does not use a next hop address so it can be whatever one wants

Solution 3: Subnetting

- Divide a class network into several subnets
  - Called **subnetting** or **subnet addressing**

- **New underlying assumption**: There is a one-to-one mapping between SPNs and subnets such that, if SPN N is mapped to subnet S, then the address of each interface on N is a member of S

- Subnetting should be kept simple within an organization:
  - All subnet masks should be contiguous (i.e., a string of 1s followed by a string of 0s)
  - All the subnets of the organization should have the same mask
  - All hosts in the organization should participate in subnetting

Solution 4: Supernetting

- Combine a range of class networks into a subnet
  - Called **supernetting**, **supernet addressing**, or **classless addressing**

- Benefits:
  - Several Class C networks can be used instead of a class B network
  - Routing tables are smaller
  - Internet Service Providers (ISPs) can manage a collection of class C networks

- Routing is complicated because an address does not self-identify the subnet it belongs to

Subnet Routing

Each host or router h contains a routing table with entries of the form (a, m, r, i) where:

1. a is the subnet address of an SPN N directly connected to h
2. m is the subnet mask of N
3. r is the IP address of the next hop router on N or $*$ (which signifies that the next hop is the destination address of the datagram)
4. i is the network interface that connects h to N

Special cases

- A class A network route has the form $(a, 255.0.0.0, r, i)$ where a is the network address of the class
- A class B network route has the form $(a, 255.255.0.0, r, i)$ where a is the network address of the class
- A class C network route has the form $(a, 255.255.255.0, r, i)$ where a is the network address of the class
- A host-specific route has the form $(a, 255.255.255.255, r, i)$ where a is the address of the host
- A default route has the form $(0.0.0.0, 0.0.0.0, r, i)$

Unified Routing Algorithm

1. Extract the destination IP address d from datagram

2. Deliver the datagram to the host if d matches one of the IP addresses of the host (for incoming datagrams only)

3. Otherwise forward the datagram as specified by the first entry (a, m, r, i) in the host's routing table such that

$$d \text{ bitwise-and } m = a$$

4. Otherwise declare a routing error

Delivery Failure

- The delivery of an IP datagram may fail because:
  - Networking hardware and software are not functioning correctly
  - The destination host or intermediate routers are down
  - The routing tables of the source host or intermediate routers are misconfigured
  - The routing path is too long (and therefore the time-to-live limit is surpassed)
  - Datagram traffic is too congested

- There needs to be a mechanism for reporting network failures
  - Cannot be implemented in hardware
  - Must use the IP protocol

Internet Control Message Protocol (ICMP)

- ICMP is for:
  - Reporting network failures
  - Controlling network traffic

- ICMP reports but does not correct errors
  - Errors are reported only to the source address of the IP datagram that could not be delivered
  - Fixing errors requires cooperation between host administrators and network administrators
  - ICMP messages are encapsulated in IP datagrams
    - The protocol field of the IP datagram is set to 1 (for ICMP)
    - The ICMP message is held in the IP datagram's data area

When ICMP is Not Used

ICMP messages are not sent in response to:

- An ICMP message
- A datagram with a broadcast destination address
- A datagram with a source address that does not define a single host (i.e., zero address, loopback address, broadcast address, or multicast address)
- A noninitial IP datagram fragment

Format of an ICMP Message

- Header
  - **Type** (8 bits) identifies type of message
  - **Code** (8 bits) identifies subtype of message
  - **Checksum** (16 bits) holds checksum of entire message
- Data area
  - Header of the failed IP datagram
  - First 64 data bits of failed IP datagram

Destination Unreachable Messages

- **Destination unreachable messages** have type 3, code 0–12

- Means router cannot forward or deliver IP datagram
  - Message is sent to the datagram's source address
  - Router **drops** the datagram

- **Network unreachable message** (code 0) usually means there is a routing error

- **Host unreachable message** (code 1) means that the datagram could not be directly delivered

- **Port unreachable message** (code 3) means that no server is listening at the requested port

Source Quench Messages

- **Source quench messages** have type 4, code 0
- Means a router has to drop a message due to traffic congestion
- Types of congestion:
  - Too many datagrams coming from one host
  - Too many datagrams coming from several hosts together

Redirect Messages

- **Redirect message** have type 5, code 0–3
- Used by a router to tell a host to change one of its routes
  - Router and host must be on the same SPN
  - Does not solve the general problem of propagating routes
- Allows a host to boot with minimal routing information

Ping Service

- The **ping service** uses **echo request** (type 8, code 0) and **echo reply** (type 0, code 0) to test if a specified destination IP address is reachable

- A successful request/reply shows:
  - Source host has IP working and can route IP datagrams
  - Intermediate routers can route IP datagrams to the destination correctly
  - Destination host is running, has IP working, can route IP datagrams, and has ICMP working

- Sophisticated versions of ping will provide statistics about datagram loss and response times

- Ping can be used by hackers to probe networks

Miscellaneous Messages

- **Time exceeded message** (type 11, code 0–1)
  - For code 0, means time-to-live limit was exceeded
  - For code 1, means fragment assembly time limit was exceeded
- **Parameter problem message** (type 12, code 0–1)
  - Usually means format of datagram's header is wrong
- Clock synchronization service
  - Uses **timestamp request** (type 13, code 0) and **timestamp reply** (type 14, code 0) to ask another machine for the time
- Subnet mask determination service
  - Uses **subnet mask request** (type 17, code 0) and **subnet mask reply** (type 18, code 0) to ask another machine for the subnet mask of the local network