Network Layer – Forwarding

Kurose & Ross, Chapter 4 (5th ed.)

Many slides adapted from:
J. Kurose & K. Ross
Computer Networking: A Top Down Approach (5th ed.)
Addison-Wesley, April 2009.
Copyright 1996-2010, J.F Kurose and K.W. Ross, All Rights Reserved.

Network Layer

- transport segment from sending to receiving host
- on sending side, encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in every host, router
- router examines header fields in all IP datagrams passing through it

Two Key Network-Layer Functions

- **forwarding**: move packets from router’s input to appropriate router output
- **routing**: determine route taken by packets from source to destination
  - **routers**: algorithms

Network service model

- **Q**: What service model for “channel” transporting datagrams from sender to receiver?
  - example services for individual datagrams:
    - guaranteed delivery
    - guaranteed delivery with less than 40 msec delay
  - example services for a flow of datagrams:
    - in-order datagram delivery
    - guaranteed minimum bandwidth to flow
    - restrictions on changes in inter-packet spacing

Network layer service models:

<table>
<thead>
<tr>
<th>Network Architecture</th>
<th>Service Model</th>
<th>Bandwidth</th>
<th>Loss</th>
<th>Order</th>
<th>Timing</th>
<th>Congestion feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet</td>
<td>best effort</td>
<td>none</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no (inferred via loss)</td>
</tr>
<tr>
<td>ATM</td>
<td>CBR</td>
<td>constant</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no congestion</td>
</tr>
<tr>
<td>ATM</td>
<td>VBR</td>
<td>guaranteed</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>no congestion</td>
</tr>
<tr>
<td>ATM</td>
<td>ABR</td>
<td>guaranteed minimum</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>ATM</td>
<td>UBR</td>
<td>none</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

IP provides best-effort service
Virtual circuits

"source-to-dest path behaves much like telephone circuit"
- performance-wise
- network actions along source-to-dest path

- call setup, teardown for each call before data can flow
- each packet carries VC identifier (not destination host address)
- every router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be allocated to VC (dedicated resources = predictable service)

VC implementation

a VC consists of:
1. path from source to destination
2. VC numbers, one number for each link along path
3. entries in forwarding tables in routers along path
- packet belonging to VC carries VC number (rather than dest address)
- VC number can be changed on each link.
  - New VC number comes from forwarding table

Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address
  - packets between same source-dest pair may take different paths

Datagram or VC network: why?

Internet (datagram)
- data exchange among computers
  - "elastic" service, no strict timing req.
- "smart" end systems (computers)
  - can adapt, perform control, error recovery
  - simple inside network, complexity at "edge"
- many link types
  - different characteristics
  - uniform service difficult

ATM (VC)
- evolved from telephony
- human conversation:
  - strict timing, reliability requirements
  - need for guaranteed service
- "dumb" end systems
- telephones
- complexity inside network

The Internet Network layer

Host, router network layer functions:

Transport layer: TCP, UDP
Routing protocols - path selection
Routing table
IP addressing conventions
Datagram format
Packet handling conventions

ICMP protocol - error reporting
OSPF, BGP - router "signaling"
Dynamic Host Configuration Protocol

- How can a newly connected host get an IP address?
  - (And other useful information: its network mask, an IP address of DNS server etc.)
  - Manual configuration by a system administrator
  - DHCP
    - Plug and play – host obtains this information automatically
    - Defined in RFCs 2131 & 2132.
    - Sent over UDP. Server’s port: 67, client’s port: 68.
    - As usual, we’ll give an overview and not go into all the technical details.

DHCP client-server scenario

- Arriving client sends a DHCP discover message
  - Src IP: 0.0.0.0 ("this host")
  - Dst IP: 255.255.255.255 ("broadcast")
  - Transaction ID: some number x.
  - Message is broadcast to all nodes on the subnet

DHCP client-server scenario 2

- Server replies with a DHCP offer message
  - Src IP: server’s IP
  - Dst IP: 255.255.255.255 ("broadcast")
  - Why? The client still can’t receive direct messages – it has no IP address.
  - Transaction ID: x
  - Message contains the proposed IP address

DHCP client-server scenario 3

- Why aren’t the previous messages enough?
  - The network might contain more than one DHCP server; a client may receive multiple DHCP offers.
  - Client sends a DHCP request message
    - Src IP: 0.0.0.0 ("this host")
    - Dst IP: 255.255.255.255 ("broadcast")
    - Why? To allow other servers know their offer was declined
    - DHCP Server ID: the IP of the server whose offer the client wish to accept
    - Transaction ID: some number y
DHCP client-server scenario 4
- Server responds with a DHCP ACK message
  - Src IP: Server’s IP
  - Dst IP: 255.255.255.255 ("broadcast")
  - The client still doesn’t have an IP address
  - Transaction ID: y

Lease times
- DHCP servers assigns a lease-time for each IP address allocation
  - A client may renew its allocation when it is about to expire
  - A client may relinquish its allocation

Internet Control Message Protocol
- ICMP – defined in RFC 792.
- Carried directly over IP
  - No transport protocol used
- Used by hosts and routers to communicate network layer information – usually report errors.

ICMP (cont.)
- For instance: “Destination unreachable”:
  - A router was unable to find a path to host B specified in host A’s request
  - The router sends an ICMP message (type 3) to A to indicate the error
- Another common use: ping
  - Host A sends an ICMP message (type 1) to host B
  - Host B sees this, and replies to A with another type 1 ICMP message.

ICMP: Internet Control Message Protocol
- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
  - network-layer "above" IP:
    - ICMP msgs carried in IP datagrams
    - ICMP message: type, code plus first 8 bytes of IP datagram causing error

<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 (ping)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>dest host unreachable</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>dest protocol unreachable</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>dest port unreachable</td>
</tr>
<tr>
<td>0</td>
<td>6</td>
<td>dest network unknown</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>dest host unknown</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>source quench (congestion control - not used)</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>echo request (ping)</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>route advertisement</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>router discovery</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>TTL expired</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>bad IP header</td>
</tr>
</tbody>
</table>

Traceroute and ICMP
- Source sends series of UDP segments to dest
  - First has TTL=1
  - Second has TTL=2, etc. unlikely port number
- When nth datagram arrives to nth router:
  - Router discards datagram and sends to source an ICMP message (type 11, code 0)
  - ICMP message includes name of router & IP address
- when ICMP message arrives, source calculates RTT
- Traceroute does this 3 times

Stopping criterion
- UDP segment eventually arrives at destination host
- Destination returns ICMP "port unreachable" packet (type 3, code 3)
- when source gets this ICMP, stops.
IP Addressing: introduction

- IP address: 32-bit identifier for host or router interface
- interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - a host has typically a single interface
- IP addresses associated with interface, not host, or router

<table>
<thead>
<tr>
<th>IP address</th>
<th>Binary form</th>
</tr>
</thead>
<tbody>
<tr>
<td>223.1.1.1</td>
<td>11001000 00010111 00000001 00000001</td>
</tr>
<tr>
<td>223.1.1.2</td>
<td>11001000 00010111 00000001 00000000</td>
</tr>
<tr>
<td>223.1.1.3</td>
<td>11001000 00010111 00000001 00000000</td>
</tr>
<tr>
<td>223.1.1.4</td>
<td>11001000 00010111 00000001 00000000</td>
</tr>
<tr>
<td>223.1.2.9</td>
<td>11001000 00010111 00000000 00000001</td>
</tr>
<tr>
<td>223.1.2.2</td>
<td>11001000 00010111 00000001 00000000</td>
</tr>
<tr>
<td>223.1.2.1</td>
<td>11001000 00010111 00000001 00000000</td>
</tr>
<tr>
<td>223.1.3.2</td>
<td>11001000 00010111 00000000 00000001</td>
</tr>
<tr>
<td>223.1.3.1</td>
<td>11001000 00010111 00000001 00000000</td>
</tr>
<tr>
<td>223.1.3.27</td>
<td>11001000 00010111 00000001 00000011</td>
</tr>
<tr>
<td>223.1.1.1</td>
<td>11011111 00000001 00000001 00000001</td>
</tr>
</tbody>
</table>

Q: What is the router's IP address in the drawing we see?

IP Addressing

- IP address is divided into two parts:
  - network prefix
    - high order bits
  - host number
    - remaining low order bits

- This partitioning of the address depends on the context network in which we see this NIC
  - networks are nested inside each other

What is a network in IP view?

IP network terminology:

- a Subnet is:
  - a set of devices that can physically reach each other without intervening router(s)
    - e.g., a LAN
- a Network is:
  - a subnet, or:
  - the union of several subnets that are interconnected by links

- three subnets (LANs) 223.1.1.*, 223.1.2.*, 223.1.3.*, together they form a larger network with prefix 223.1 (16 bits) (OR MORE bits?)

IP Addresses

given notion of "network", let's re-examine IP addresses:

"classful" addressing:
(does not need mask or /K indicator)

<table>
<thead>
<tr>
<th>Class</th>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.0.0.0</td>
<td>255.255.255.255</td>
</tr>
<tr>
<td>B</td>
<td>128.0.0.0</td>
<td>191.255.255.255</td>
</tr>
<tr>
<td>C</td>
<td>192.0.0.0</td>
<td>223.255.255.255</td>
</tr>
<tr>
<td>D</td>
<td>224.0.0.0</td>
<td>239.255.255.255</td>
</tr>
</tbody>
</table>

Given 32 bits

Q: What is the router's IP address in the drawing we see?

Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:

- Classless InterDomain Routing
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address
  - Requires inclusion of mask or /K in routing table
  - network part 11001000 00001110 00001000 00000000
  - host part 200.23.16.0/23

- CIDR: Classless InterDomain Routing
  - inefficient use of address space, address space exhaustion
  - e.g., class B net allocated enough addresses for 65k hosts, even if only 2k hosts in that network

- CIDR: Classless InterDomain Routing
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address
  - Requires inclusion of mask or /K in routing table
  - network part 11001000 00010111 00000001 00000000
  - host part 200.23.16.0/23

- CIDR: Classless InterDomain Routing
  - inefficiency use of address space, address space exhaustion
  - e.g., class B net allocated enough addresses for 65k hosts, even if only 2k hosts in that network

- CIDR: Classless InterDomain Routing
  - network portion of address of arbitrary length
  - address format: a.b.c.d/x, where x is # bits in network portion of address
  - Requires inclusion of mask or /K in routing table
  - network part 11001000 00010111 00000001 00000000
  - host part 200.23.16.0/23
Hierarchical addressing: more specific routes

ISP’s-R-Us has a more specific route to Organization 1

ISP’s-R-Us

Organization 0

200.23.16.0/23

Organization 2

200.23.20.0/23

Fly-By-Night-ISP

"Send me anything with addresses beginning 200.23.16.0/20"

Organization 7

200.23.30.0/23

ISP’s-R-Us

"Send me anything with addresses beginning 200.23.16.0/20 or 200.23.18.0/23"

Internet

Print the routing table

- Windows: "route print"
- Linux: "netstat -rn"

- On nova.cs.tau.ac.il (132.67.192.133) this gives:

  nova 2% netstat -rn

  Kernel IP routing table
  Destination Gateway Genmask Flags MSS Window Irtt Iface
  132.67.192.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
  0.0.0.0 132.67.192.1 0.0.0.0 UG 0 0 0 eth0

Exercise (Peterson & Davie, 5th ed.)

- Suppose A and B have been assigned the same IP address on the same Ethernet, on which ARP is used. B starts up after A.
  - What will happen to A’s existing connections?
  - Every device on the LAN which already has an ARP entry for A, upon receiving a packet from B, will update its ARP table and will now send to B.
  - For instance, if B transmits an ARP query (broadcast!) then all of A’s connections will be cut.

Exercise (Peterson & Davie, 5th ed.)

- Suppose A and B have been assigned the same IP address on the same Ethernet, on which ARP is used. B starts up after A.
  - How can A guard against this?
    - A might monitor for ARP broadcasts purportedly coming from itself.
    - A might even immediately follow such broadcasts with its own ARP broadcast in order to return its traffic to itself.

Exercise (Peterson & Davie, 5th ed.)

- Suppose A and B have been assigned the same IP address on the same Ethernet, on which ARP is used. B starts up after A.
  - Explain how “self-ARP” (querying the network on start-up for one’s own IP address) might help with this problem.
  - If B uses self-ARP on startup, it will receive a reply indicating that its IP address is already in use
  - This is a clear indication that B should not continue on the network until the issue is resolved.

Extra slides

Review of lecture, if time permits
**Interplay between routing and forwarding**

Value in arriving packet's header:

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initiate call</td>
</tr>
<tr>
<td>2</td>
<td>Incoming call</td>
</tr>
<tr>
<td>3</td>
<td>Accept call</td>
</tr>
<tr>
<td>4</td>
<td>Call connected</td>
</tr>
<tr>
<td>5</td>
<td>Data flow begins</td>
</tr>
<tr>
<td>6</td>
<td>Receive data</td>
</tr>
</tbody>
</table>

**Virtual circuits: signaling protocols**

- used to setup, maintain, teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet

**IP addresses: how to get one?**

**Q:** How does network get subnet part of IP addr?

**A:** gets allocated portion of its provider ISP's address space

<table>
<thead>
<tr>
<th>ISP's block</th>
<th>110010100 00010111 00010000 00000000</th>
<th>200.23.16.0/20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization 0</td>
<td>110010100 00010111 00010000 00000000</td>
<td>200.23.16.0/23</td>
</tr>
<tr>
<td>Organization 1</td>
<td>110010100 00010111 00010010 00000000</td>
<td>200.23.18.0/23</td>
</tr>
<tr>
<td>Organization 2</td>
<td>110010100 00010111 00010100 00000000</td>
<td>200.23.20.0/23</td>
</tr>
<tr>
<td>Organization 7</td>
<td>110010100 00010111 00011110 00000000</td>
<td>200.23.30.0/23</td>
</tr>
</tbody>
</table>

**IP addressing: the last word...**

**Q:** How does an ISP get block of addresses?

**A:** ICANN: Internet Corporation for Assigned Names and Numbers
- allocates addresses
- manages DNS
- assigns domain names, resolves disputes