Network Layer Protocols:

IPv4 (Internet Protocol Version 4)

Internet Protocol Version 4 (IPv4)

- Network layer protocol in TCP/IP suite
- Defined originally in RFC 791 (*what is a RFC?)*
- Connectionless (i.e., no explicit connection setup/termination phase before/after data transfer), datagram-oriented
- Message broken up into packets, packets switched between routers. Header attached to each packet (IP header).
- Main issues handled:
  - Routing
  - Fragmentation and reassembly
- Unreliable, best-effort service. Packets can be lost, duplicated, received out-of-sequence.
- Encapsulated in data link layer frame (for ex. Ethernet)

IP Address

- Required to specify source and destination of packet
- Each host connected to the Internet is identified by a unique IP address (actually, each network card on a host has an IP address)
- An IP address is a 32-bit quantity.
  - Expressed as W.X.Y.Z, where W, X, Y, Z are the four octets
  - Consists of two logical parts:
    - A prefix for network id
    - A suffix for host id within that network
- Important to remember that an IP address is actually assigned to an interface (NIC), not a machine
  - A machine may have more than one IP addresses if it has more than one NIC
    - Ex. – your laptop (the machine) can have one IP address assigned to the LAN interface and another to the wireless interface
  - Though the term host, machine, device, …used interchangeably and informally very commonly when talking about IP address assignment

Classful Addressing

- All addresses are to fall in five well-defined IP address classes

Class A	UNICAST
Class B	UNICAST
Class C	UNICAST
Class D	MULTICAST
Class E	RESERVED

- Each class has a pre-defined number of bits for the network part

Class	Address Range	High-order bits	Network bits	Host bits
A	0.0.0.0 – 127.255.255.255	0	7	**24**
B	128.0.0.0 – 191.255.255.255	10	14	**16**
C	192.0.0.0 – 223.255.255.255	110	21	**8**
D	224.0.0.0 – 239.255.255.255	1110		
E	240.0.0.0 – 255.255.255.255	1111		

CLASS A

0 || Network	Host	Host	Host

No. of networks: $2^{7} – 1 = 127$

No. of hosts/network: $2^{24} – 2 = 16,777,214$

CLASS B

10 || Network	Network	Host	Host

No. of networks: $2^{14} – 1 = 16,383$

No. of hosts/network: $2^{16} – 2 = 65,534$

CLASS C

110 || Network	Network	Network	Host

No. of networks: $2^{21} – 1 = 2,097,151$

No. of hosts/network: $2^{8} – 2 = 254$

Network Specification

- Specified as W.X.Y.Z/p where
  - W.X.Y.Z has the same form as an IP address
  - p is an integer saying first p bits of the address identify the network, the remaining (32 – p) bits can be used for assigning host addresses in that network
- For class A, B, C, p is 8, 16, 24 respectively
  - 64.0.0.0/8 is a Class A network
  - 129.100.0.0/16 is a Class B network
  - 223.200.100.0/24 is a Class C network

Some Special IP Addresses

- Loopback address
  - Loopback address - allows applications on same host to communicate using TCP/IP
  - Anything starting with 127., usually 127.0.0.1
- Net-directed Broadcast (to *netid*)
  - NETID = *netid,* HostID = all 1's
  - Not assigned to any host
- Network address
  - NETID = *netid,* HostID = all 0's
  - Not assigned to any host
- Private IP addresses
  - 10.0.0.0/8, 172.16.0.0-172.31.255.255, 192.168.0.0/16
  - Should be used only internally in an organization, should never go on the internet
  - No central allocation, anyone can use it

Problems with Classful Addressing

- IP block allocation was done by class
  - An organization gets only multiples of Class A, B, and C
  - Wastage of IP addresses
    - Organizations may not need a full class
      - Especially for Class A and B
    - Big problem, as IP address space is bounded
      - May run out of IP addresses for new machines eventually

Classless Addressing

- No well-defined class
  - No fixed size prefix for network part
  - Choose the prefix size as per need
- Allocate only what is needed
  - Value of p (number of network bits) can be arbitrary
  - Ex: An organization needs only 56 IP addresses
    - Classful addressing option: Allocate a whole Class C, say, 220.100.200.0/24 (24 network bits, 8 host bits), wastes around 200 addresses
    - With classless addressing, you can allocate 220.100.200.0/26 (256 network bits, 6 host bits)
    - Allows the allocation of 220.100.200.64/26, 220.100.200.128/26, and 220.100.200.192/28 to 3 other organizations with similar needs (needs to connect < 64 machines)

IP Address Allocation

- Done centrally by IANA (www.iana.org)
- IANA assigns unallocated IP blocks to RIRS (Regional Internet Registries)
  - AFRINIC, APNIC, LACNIC, ARIN, RIPE NCC
- RIRs allocates to ISPs and other users in their region
- Defined policies exist for allocation
- Region/ISP based allocations keep IP address blocks in a region mostly contiguous
  - Important for keeping routing tables small (we will see)

IP Subnetting

- Subnet (or subnetwork)
  - A subdivision of a network
  - Breaks host part further into subnet part and host part
- Allows better network administration and management
- Reduces route table size in non-local routers (will see later how)

- Uses network masks (subnet mask)
  - In binary, the mask is a series of contiguous 1's followed by a series of contiguous 0's.
  - The 1's portion identifies the network portion of the address (see example)
  - The 0's portion identifies the host portion of the address in the original
  - To check if two IP addresses belong to the same subnet or not, check if the bit-wise AND of the two addresses with the netmask is the same or not.

An Example

- Network mask 255.255.255.240 is applied to a class C network 195.16.100.0
- Mask = 11111111 11111111 11111111 11110000
- Address of 1st host on this subnet = 195.16.100.1 (0 = 0000 is special)
- Address of last host on this network = 195.16.100.14 (15 = 1111 is special)
- Next subnet will start from 195.16.100.16
- Addresses 195.16.100.3 and 195.16.100.12 are in the same subnet by the previous rule
- Addresses 195.16.100.3 and 195.16.100.19 are in different subnets by the previous rule

Note: original subnet specs (RFC 950) do not allow subnets with all 0's and all 1's (so 195.16.100.0 and 195.16.100.240 subnets cannot be used). Current systems allow with suitable configuration.

Subnetting Example

- We want to break 195.6.100.0/24 into 2 subnets with 64 addresses each, 4 subnets with 32 addresses each
- We first get 4 no. 64 address subnets
    - 64 addresses needs 6-bit host address
    - So network part is 32 – 6 = 26
        - Top 2-bits of the last 8 bits (the host part in the original network) are to be included in network part of the subnets
    - Get the 4 subnets by varying the 7th and 8th bit (from right) from 00 to 11
        - 195.6.100 part of the address remains the same
    - The four subnets are
        - 195.6.100.0/26 (Subnet 1, bit value = 00)
            - Addresses 195.6.100.0 to 195.6.100.63
        - 195.6.100.64/26 (Subnet 2. bit value = 01)
            - Addresses 195.6.100.64 to 195.6.100.127

- 195.6.100.128/26 (Subnet 3, bit value = 10)
  - Addresses 195.6.100.128 to 195.6.100.191
- 195.6.100.192/26 (Subnet 4, bit value = 11)
  - Addresses 195.6.100.192 to 195.6.100.255

- Now we break Subnet 3 into 2 no. 32 address subnets
  - 32 addresses need 5-bit host part
  - So network part is 32 – 5 = 27 bits
    - Top 1-bit of the last 6 bits (the host part in Subnet 3) is to be included in network part of the new subnets
  - Get the 2 subnets by varying the 6th bit (from right) from 0 to 1
    - Remaining bits of Subnet 3 remains the same
  - The two subnets are
    - 195.6.100.128.0/27 (Subnet 3(a), 6th bit value = 0)
      - Addresses 195.6.100.128 to 195.6.100.159
    - 195.6.100.160/27 (Subnet 3(b), 6th bit value = 1)
      - Addresses 195.6.100.160 to 195.6.100.191
- Break Subnet 4 similarly (work it out writing out the bit patterns)

Natural Masks

- Class A, B and C addresses each have natural masks, which gets defined from the definition of the classes themselves.
  - Class A :: natural mask is 255.0.0.0
  - Class B :: natural mask is 255.255.0.0
  - Class C :: natural mask is 255.255.255.0
- Network address can be specified by either specifying the netmask explicitly, or by the / notation

Routing IP Packets

- The process of finding a path from a source to a destination through intermediate nodes
- A route – a path from the source to the destination
- Router – the intermediate nodes that forward the packet towards its destination
- Routing Table
  - Kept at each node
  - Contains the route to each destination node reachable from that node
  - A routing protocol fills up and maintains the routing table at the nodes
    - Updates routes if existing routes change or new better routes found
  - Entries can also be added/deleted manually

How is a route stored?

- Basic fields of a route
  - Destination
    - Route to which destination?
  - Next hop
    - To reach that destination, what is the next node the packet should be forwarded to
      - The next node will then use its next node field to go one more hop and so on till the destination is reached
  - Cost
    - What is the cost of this route?
      - Definition of cost can vary. One common metric is the number of hops (links) to the destination
- There are usually other fields, we will see, but these three are the most important and what you need now to understand routing

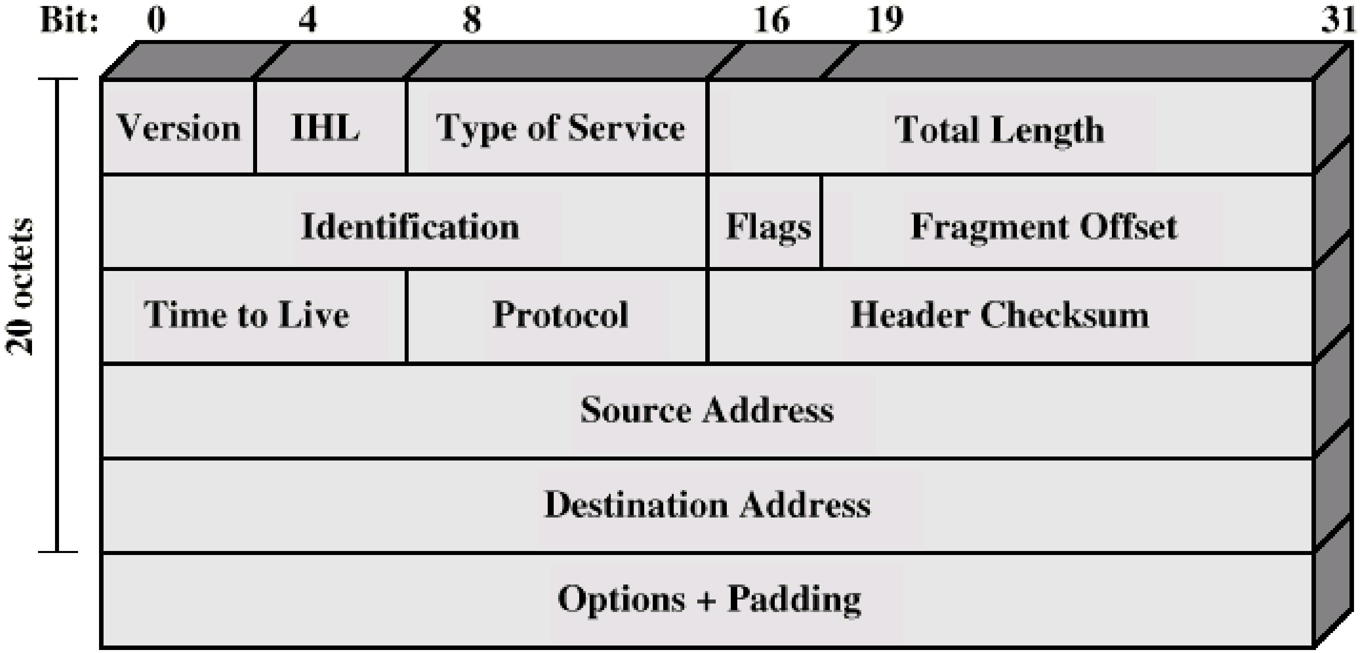
Routing vs. Forwarding

- Routing is the process of finding the routes
  - Routing protocols can run even if no packets need to be sent
    - Keeps the table ready to forward a packet
- Forwarding is the process of using a route to send a packet towards the destination
  - Done only when a packet arrives at the node whose final destination is not this node
  - Done in real time
- Sometimes, routing is loosely used to refer to both
  - "route a packet"
- We will study routing protocols to create the routing table separately, right now just assume that the table exists

Packet Forwarding

- Each machine has a routing table
- Any IP packet received (from the Ethernet layer) is checked against the routing table
- Forwarded to higher layer (TCP/UDP) if this machine is the destination
- Forwarded to next hop using the interface specified in the entry
- But first lets see what is in the IP header

IP Header



Bit: 0 4 8 16 19 31

20 octets

Version | IHL | Type of Service | Total Length

Identification | Flags | Fragment Offset

Time to Live | Protocol | Header Checksum

Source Address

Destination Address

Options + Padding

Figure 15.6 IPv4 Header

What’s Stored in an IPv4 Header?

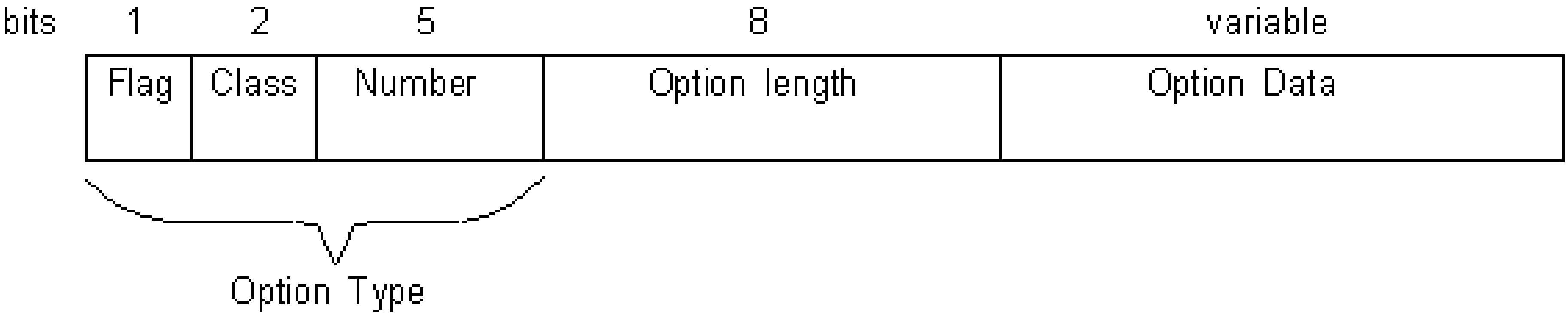
- Version: 4 bit field specifying the IP version (4)
- Header length: specified in 32 bit words. Range is 5..15 words, or 20..60 bytes
- Type of Service (8 bits)
  - 3 bit precedence field, one “must be zero” field
  - 4-bit field specifying desired service qualities.
    - Minimize Delay
    - Maximum Throughput
    - Maximize Reliability
    - Minimize Monetary Cost
  - Only one bit can be set. None set is “normal service”
  - Largely ignored by routers & IP implementations

IPv4 Header (contd.)

- Datagram length: header + data, in bytes
- Identification
  - Unique value, used with flags & fragment offset if a message must be fragmented
- Flags, Fragment offset – discussed later
- Time to live field - upper limit on the number of "hops" a message can go before being dropped
- Protocol: identifies higher layer protocol like TCP, UDP *etc.*
- Header checksum: checksum of just the header
- Source address
- Destination address
- Options

Options Field

- Can specify a variable number of options, each of the form



bits	1	2	5	8	variable
	Flag	Class	Number	Option length	Option Data

Option Type

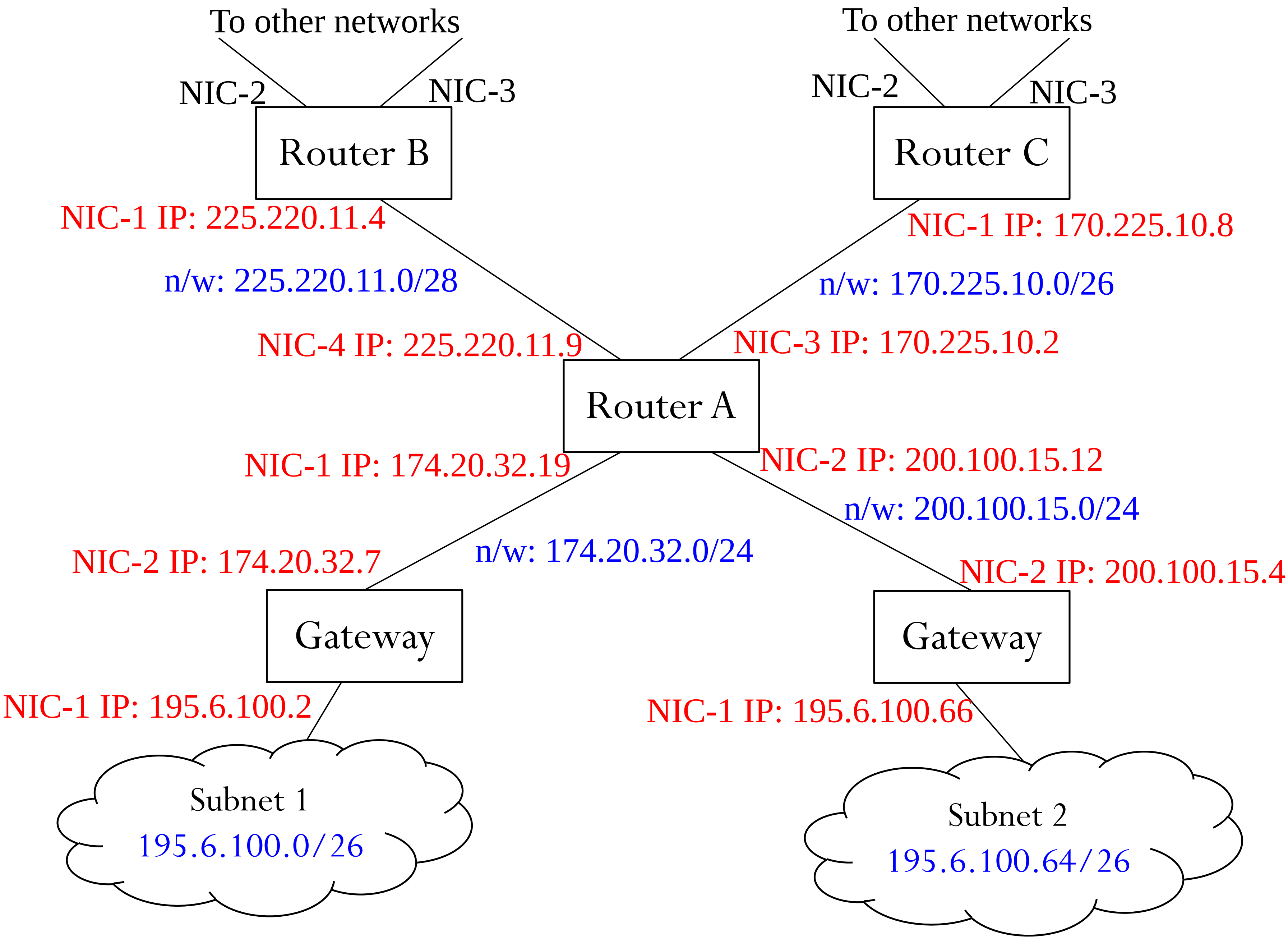
- Flag – 0 means the option is NOT to be copied to each fragment if the datagram is fragmented, 1 means to be copied
- Class – 0 (Normal), 2 (debugging)

- . Option Number
  - 0 - the end of the option list,
  - 1 - No Operation
  - 2 - Security
  - 3 - Loose Source Routing
  - 4 - Internet Timestamp
  - 7 - Record Route
  - 8 - Stream ID
  - 9 - Strict Source Routing
- Option-Length - variable and not present for the NOP and the end of Option List
- Option-Data - variable and not present for the NOP and the end of Option List. See RFC 791 for the detail on the data content for each of the options if you want
- Overall, options are not much used for internet communication

Back to Packet forwarding

- As mentioned, forwarding will check destination IP (from header) against routing table entries
- Two types of machines
  - End devices that are sources and destinations of data (ex. Your PCs/Laptops etc.)
    - Belongs to some subnet
    - Can have only one NIC with an IP address assigned from that subnet
      - May have more, ex. a wired and a wireless interface
    - Routing tables are usually static
      - Built by system during network configuration
      - Routes can be added manually also
      - No routing protocol runs to dynamically update routes
      - Small and simple tables
    - Each subnet will have a gateway (router) that connects it to the rest of the larger network

- Routers
  - Routes packets, not source or destination
  - Connects more than one network
    - Connected to the gateway/router of each network
    - So must have more than one NIC, one for connecting to each router
  - Each NIC has an IP address assigned from the network it is connected to
  - Runs some routing protocols to build the routing table dynamically
  - Static routes can also be added in addition
  - Larger size, more complex



To other networks

NIC-2

NIC-3

Router B

NIC-1 IP: 225.220.11.4

n/w: 225.220.11.0/28

NIC-4 IP: 225.220.11.9

To other networks

NIC-2

NIC-3

Router C

NIC-1 IP: 170.225.10.8

n/w: 170.225.10.0/26

NIC-3 IP: 170.225.10.2

Router A

NIC-1 IP: 174.20.32.19

n/w: 174.20.32.0/24

NIC-2 IP: 174.20.32.7

NIC-2 IP: 200.100.15.12

n/w: 200.100.15.0/24

NIC-2 IP: 200.100.15.4

Gateway

NIC-1 IP: 195.6.100.2

Subnet 1

195.6.100.0/26

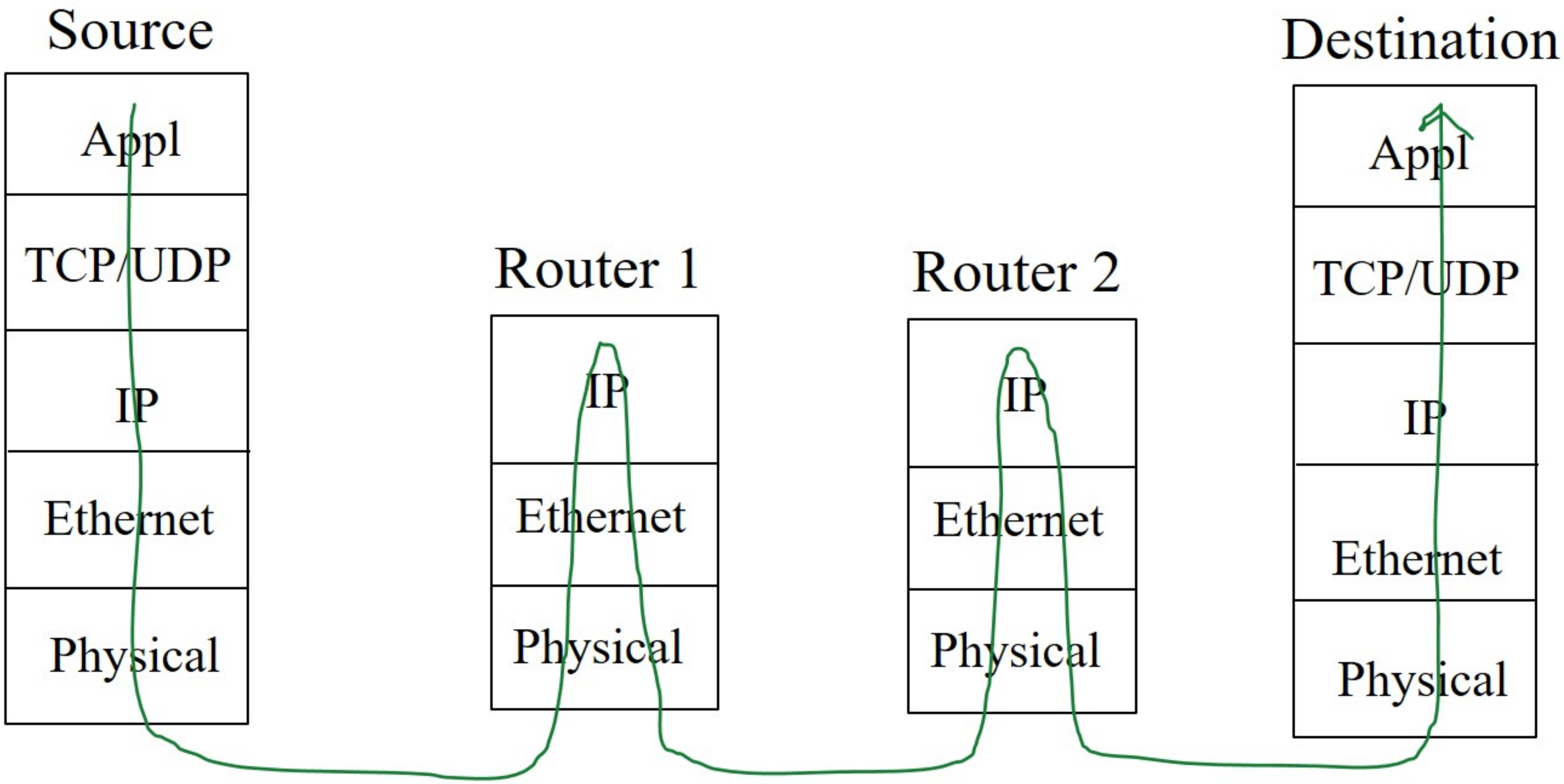
Gateway

NIC-1 IP: 195.6.100.66

Subnet 2

195.6.100.64/26

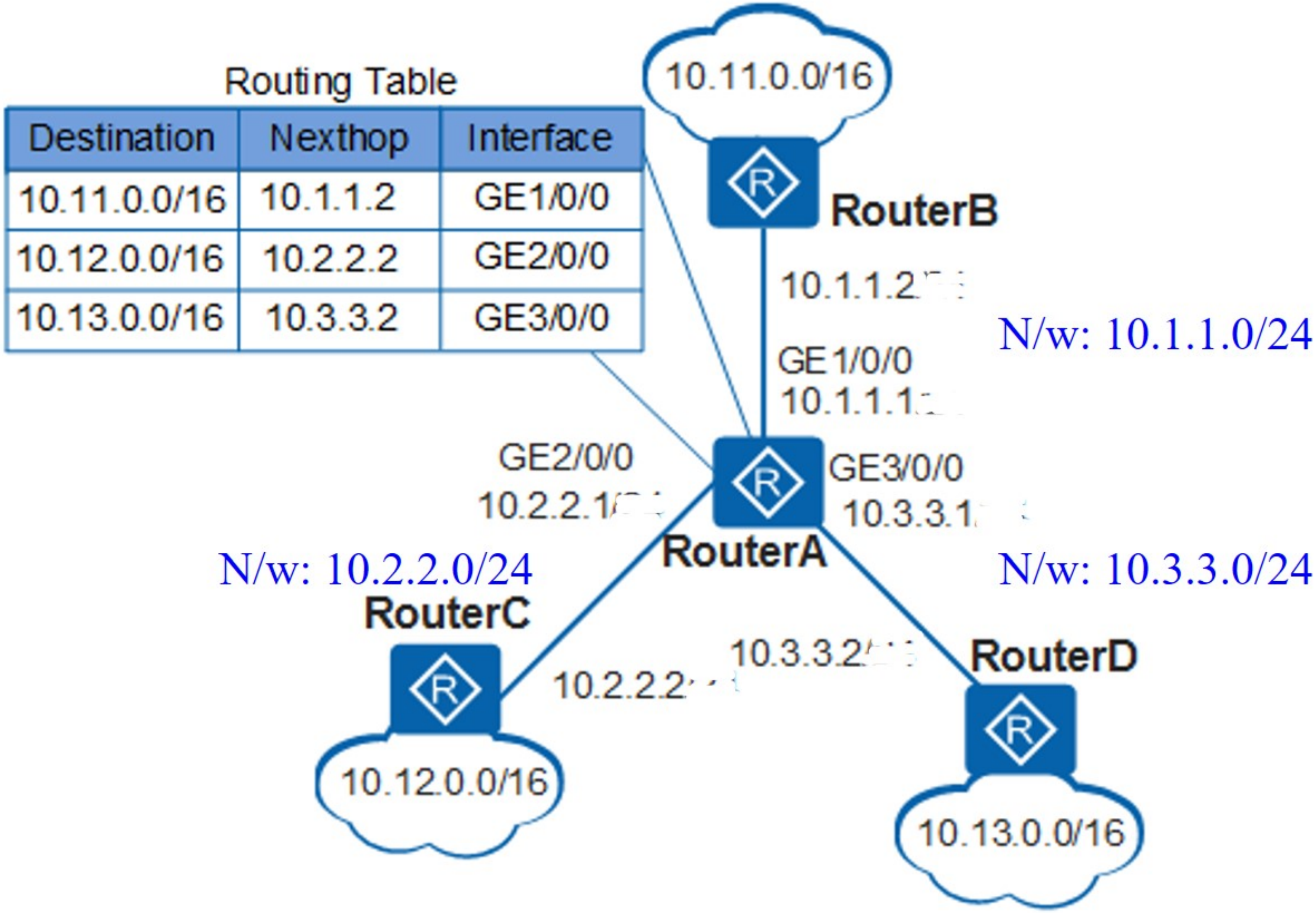
Layers that an IP packet flows through from Source to Destination



A router has only till IP layer. No need for higher layers.

Routing by the Network

- When IP packet comes, destination address checked with routing table to find next hop's IP address
- So should each router keep one entry for each host address existing in the network?
  - Will explode the routing table at higher level routers
    - Think of the no. of hosts they will have to keep track of
    - Increases both storage and search time
- Solution: Route by Network
    - Routing entries specify networks, not hosts
    - In IP routing, the "destination" field in each routing entry is a network address, not a host address
      - Look at the IP address in it in conjunction with the subnet mask or the / part



Routing Table

Destination	Nexthop	Interface
10.11.0.0/16	10.1.1.2	GE1/0/0
10.12.0.0/16	10.2.2.2	GE2/0/0
10.13.0.0/16	10.3.3.2	GE3/0/0

- Routing by network will still need one entry per subnet
  - Still very large
- CIDR (Classless Inter Domain Routing)
  - Allocate IPs in contiguous blocks using classless addressing
  - Combine contiguous networks into a larger network for the purpose
  - Saves IP space and reduces routing table size
  - Also called Supernetting or Route Aggregation

Example

- Suppose company X needs 1000 IP addresses from its ISP Y
- ISP Y allocates the network 192.60.128.0/22
  - A contiguous block of 1024 addresses
- Company X now subnets these into 8 subnets with 128 addresses each
  - 192.60.128.0/25 (Subnet 1)
  - 192.60.128.128/25 (Subnet 2)
  - 192.60.129.0/25 (Subnet 3)
  - 192.60.129.128/25 (Subnet 3)

  .
  .
  .

  - 192.60.131.128/25 (Subnet 8)

- Router of X, R_X, will have 8 entries in its routing table, one for each subnet
  - Each subnet will have its own gateway
  - A subnet's entry in R_X will have the subnet's gateway as its next hop
- Router of Y, R_Y, will have only one entry, for the "network" 192.60.128.0/22 in its routing table
  - The next hop for that entry will be R_X
- Any higher level router that R_Y connects to will also have just one entry for the 8 subnets, with R_Y as the next hop

- Possible only due to contiguous allocation, so that higher level routers can just send it to lower level routers (in this case company A's router) using one entry only for all the subnets. Lower level router will distinguish.
  - If the 8 subnets assigned were not contiguous, R_Y would need 8 entries also
- Possible because of classless addressing
  - Note that the 1024 address block is nothing but 4 contiguous Class C blocks
  - But looked at as classless, using subnet mask to distinguish

- Routing table at all higher level routers:
  - 192.60.128.0/22 - send to R_Y (next hop on way to Company X's router R_X)
- Routing table at R_X :
  - 192.60.128.0/25 – send to router of first subnet
  - 192.60.128.128/25 – send to router of second subnet
  - 192.60.129.0/25 – send to router of third subnet
  - 192.60.129.1/25 – send to router of fourth subnet
  - ….

- So routing table will contain networks (in prefix form or explicit net/mask form) and maybe some hosts (32 bit prefix), and a next hop address for each of them, plus some other information
- Routers always do longest prefix match. If two entries match, longest match is taken.
  - Example: two entries in table: one for 192.65.0.0/16 and one for 192.65.128.0/24. If address is 192.65.128.4, second entry will be used even though it matches both.
- Usually a routing cache is also there. Cache contains recent routing decisions. Destination address first looked up in cache. If not found, longest-prefix match done in routing table.

Routing Table Example (PC that is source and destination of data)

```
========================================================================
Active Routes:
Network Destination        Netmask                    Gateway              Interface              Metric
       0.0.0.0               0.0.0.0                   10.124.52.2          10.124.52.149          35
     10.124.52.0             255.255.255.0            On-link               10.124.52.149          291
    10.124.52.149           255.255.255.255           On-link              10.124.52.149           291
    10.124.52.255           255.255.255.255           On-link              10.124.52.149          291
      127.0.0.0              255.0.0.0                On-link               127.0.0.1              331
      127.0.0.1             255.255.255.255           On-link               127.0.0.1              331
  127.255.255.255           255.255.255.255            On-link              127.0.0.1             331

=======================================================================
```

Basic Packet Forwarding Rule

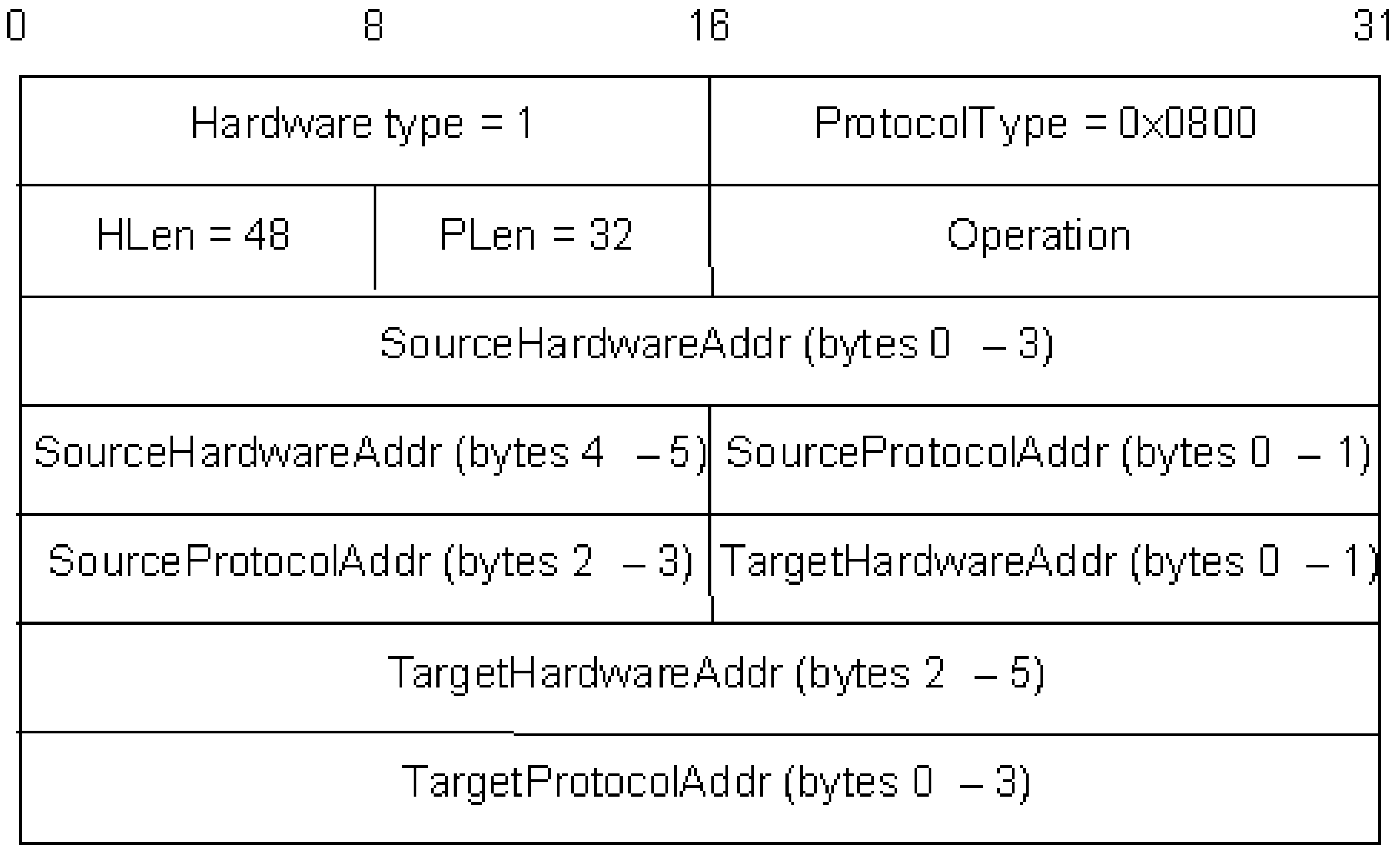
- Match the destination IP for network part against routing table entries
  - For each routing entry, bitwise AND the subnet mask to the destination IP, and see if the result is equal to the network part in the routing entry. If yes, match found
  - If multiple matches found, choose the longest prefix match
- Check the gateway (next hop) part of the matched entry
  - If local ("On-Link" in our earlier example, can be specified in other ways like IP of this machine etc.), the destination IP is in the same network, no need for any further routing.
    - Send to destination IP (how?)
  - If some other IP, send to it for further routing (how?)

- Sending to destination m/c in local network
  - If this m/c is the destination, send up to TCP/UDP layer
  - Otherwise,
    - Can send using broadcast MAC address in Ethernet frame
      - All nodes on network receive it, pass it up till IP layer, only destination m/c accepts (based on destination IP), others drop it
    - Can send with destination m/c MAC address in Ethernet frame if known
      - Only the destination passes it up to the IP layer, rest all drops it at Ethernet layer itself
      - MAC can be known by ARP (see later)
- Sending to next hop through interface specified
  - Same as above
  - Note that the next hop must have at least 2 network interfaces in this case, one in this network, one to some external network (its next hop towards the final destination)
  - One of them must be in the same network as this machine

ARP (Address Resolution Protocol)

- Provides IP to Hardware address mapping
- ARP packet encapsulated in Ethernet
- Type field in frame specifies it is an ARP packet
- Broadcast IP for which hardware address is needed to all nodes, destination picks it up and replies
- ARP cache maintained for faster mapping

ARP Packet Format



<table>
<tr><th>0</th><th>8</th><th>16</th><th>31</th></tr>
<tr><td colspan="2">Hardware type = 1</td><td colspan="2">ProtocolType = 0x0800</td></tr>
<tr><td>HLen = 48</td><td>PLen = 32</td><td colspan="2">Operation</td></tr>
<tr><td colspan="4">SourceHardwareAddr (bytes 0 – 3)</td></tr>
<tr><td colspan="2">SourceHardwareAddr (bytes 4 – 5)</td><td colspan="2">SourceProtocolAddr (bytes 0 – 1)</td></tr>
<tr><td colspan="2">SourceProtocolAddr (bytes 2 – 3)</td><td colspan="2">TargetHardwareAddr (bytes 0 – 1)</td></tr>
<tr><td colspan="4">TargetHardwareAddr (bytes 2 – 5)</td></tr>
<tr><td colspan="4">TargetProtocolAddr (bytes 0 – 3)</td></tr>
</table>

RARP

- Opposite of ARP
- Gets IP address given Hardware Address
- Used during bootup by diskless workstations etc. to initialize IP
- No role in packet forwarding

Example

- IP packet sent from A=144.16.192.55 to B=192.15.32.3 through gateway C=144.16.192.2
- Since A sends to C first, should A change the destination address in the packet to 144.16.192.2 (address of C)?

NO!!!

- IP address is for end-to-end communication, C will need it to know that the packet is for B
  - Replacing the destination IP field from B to anything else loses the final destination permanently
- A will look in ARP cache to see if C's MAC address is already there
- If found,
  - Decrement TTL field in IP header by 1 if not the source
  - Recompute checksum in header
  - Send the IP packet to Ethernet along with MAC address of C
  - Ethernet layer will put its own header with destination MAC = C's MAC address and send

- If not found,
  - Create an ARP packet with target protocol address = C's IP address (144.16.192.2)
  - Send to Ethernet with broadcast MAC address
  - Ethernet puts its own header with destination MAC = all 1's and sends
  - C receives the ARP and sends an ARP response with its MAC address
  - ARP response received by Ethernet passed back to ARP processing software
  - ARP protocol adds this to the ARP cache, and also gives to the waiting packet
  - Rest of the steps is same as the earlier case (entry found in ARP cache) for sending the packet
- C will then follow the same steps to send towards B
  - Look up routing table, find next hop etc.

- Questions
  - What happens if the user gives a wrong destination IP address?
  - What happens when the TTL becomes 0?
  - What happens if the gateway/next hop is dead?
- We have so far looked at forwarding using the routing table
- Next lets see how the routing tables are built