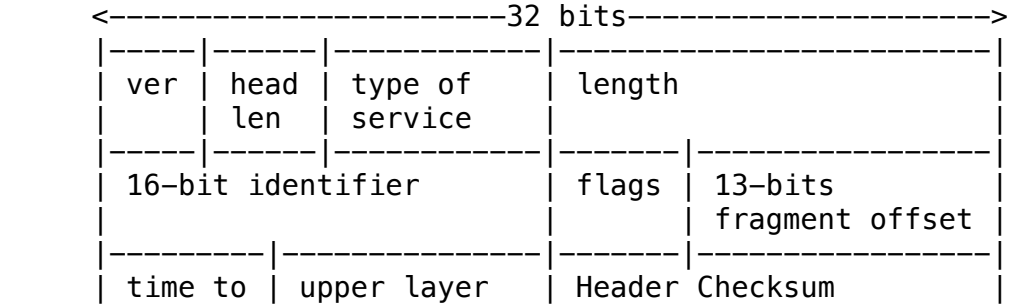
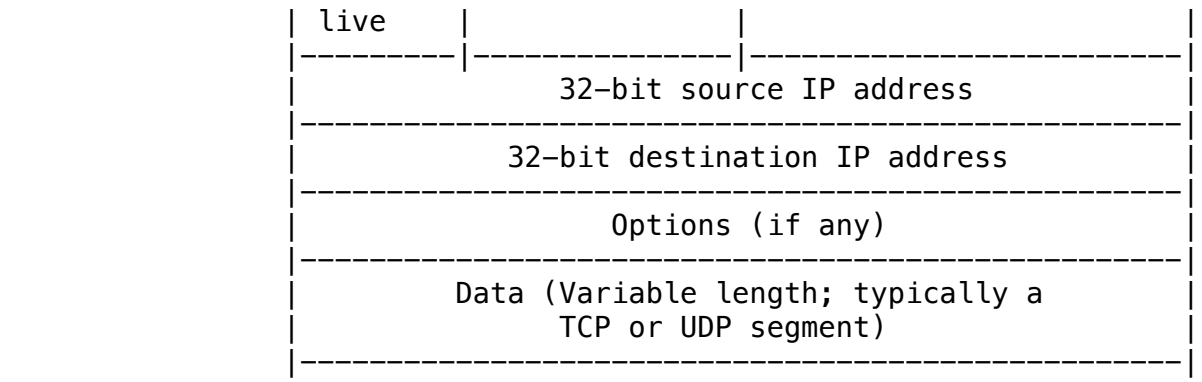
Week.8.txt

- March 1st, 2021
    - IP Layer
        - Also referred to as network layer, internet layer
        - Last class, we discussed:
            - Basic concepts
            - Key functions in the network layer
            - IP address
                - Is a 32-bit number associated with a host or a router interface
                - Typically represented in dotted-quad notation
            - The functions inside a router
                - Routing protocols
                    - These exchange information across routers to populate the forwarding table
                        - Forwarding tables are utilized, on the fly, to decide where a packet needs to be forwarded to
                            - i.e. Which outgoing interface?
                - IP protocols
                    - Specify address conventions, datagram format, and packet handling conventions
                - ICMP protocol
                    - Used for error reporting and signaling among routers
            - Details of the internals of a router
                - How are packets looked up via the information from forwarding tables
                - The key concept is that forwarding tables will store entries, corresponding blocks of addresses, and their corresponding outgoing interface that a particular packet that needs to forwarded to
                - Upon reception of a packet, routers lookup the forwarding table, utilizing a mechanism called: Longest prefix match
                    - Whichever block range has the longest prefix match is the outgoing interface the packet is forwarded to
    - IP Datagram Format
        - i.e. Diagram of IP Datagram





- 'ver' = IP protocol version number
- 'head len' = Header length (in bytes)
- 'type of service' = Type of data
- 'time to live' = Max number of remaining hops
  - Decremented at each router
- 'upper layer' = Upper layer protocol to deliver payload to
- 'length' = Total datagram length (in bytes)
- The '16-bit identifier', 'flags', and '13-bits fragment offset' are used for fragmentation/ reassembly
- Options include, but not limited to:
  - Timestamp
  - Record route taken
  - Specify list of routers to visit

- The information bits are divided into rows of 32-bits in the IP datagram
  - Each row corresponds to 4 bytes
  - Similar format to TCP and UDP
- There are several fields in the IP header:
  - IP version number
    - i.e. IPv4, IPv6, Etc.
  - Header length
    - Corresponds to the length of the IP header
      - There are optional fields in the IP datagram header that may make the IP header longer than the default value
    - Can be used to indicate how many options are present in the optional fields
  - Type of service
    - Also called TOS
    - Can be used as a way to specify the type of IP packet, and the corresponding services the packet needs to be received
      - i.e. Preferential treatment of certain IP packets, compared to other packets; in terms of scheduling decisions
      - At the outgoing interface there is a packet queue, and different packets may receive

        different scheduling decisions based on information carried in the TOS field
      - However, 'type of service' is not commonly used
- Length
  - This is the total length of the datagram
    - Includes the header as well as the payload in the IP datagram
  - Measured in bytes
- 16 bit identifier, flags, 13-bits fragment offset
  - Used for fragmentation and reassembling the IP packet
    - In other words, large segments are split into smaller segments that can be processed by the network layer
  - The maximum segment length needs to be set based on the maximum accommodated transfer unit along each link from source to destination
    - Depends on access network technology used
      - i.e. Ethernet's default is 1460
      - i.e. Wi-Fi has no limitation on maximum segment length of TCP segments
    - If the TCP segment length is larger than the maximum transfer unit of a particular link, then fragmentation is utilized at the network layer to break down a large TCP segment into fragments
      - However, the end-host will still need to reassemble the fragments into the original IP packet, and then deliver it to the transport layer
  - The identifier is used utilized to indicate which of the original IP packet those fragments belong to
  - The flag field is used to indicate whether there are further fragments for the original IP packet
  - The offset field indicates where does the fragment sit in the original IP packet
    - It is the offset of the packet
  - Often times, because of the value of the MTU discovery, the maximum segment length of TCP will be set to the value such that there is no need to perform further fragmentation at the network layer
    - However, there can be a discrepancy between the maximum segment size in the transport layer, and the fragments of IP packets that can be delivered along a particular link from source to destination
- Time to live
  - Also referred to as 'TTL'
  - Useful in implementing `traceroute`
    - `traceroute` can be used to find out the hops along the path from source to destination

    - This field allows a packet to set which hop it has reached
    - You can specify the value of this field in the IP datagram to be transmitted from your host
        - `traceroute` can be implemented via UDP or TCP, but it requires the modification of the IP header; specifically the 'time to live' field
            - It will limit how many hops a particular datagram can visit
    - The maximum number of hops in the Internet, from end-to-end, is (typically limited to) 32 hops
    - In `traceroute` you want to artificially limit how many hops a datagram can visit
        - i.e. For the initial probe in `traceroute`, you want to get a response from the first router/hop
            - To do this, set the 'TTL' field to 1, so the first hop/router is forced to generate an ICMP message
                - For the `N'th` router along a path, the 'TTL' field is set to 'N'
                    - This is how `traceroute` discovers routers along the path
    - For every router a packet is forwarded to, the router decreases the 'TTL' value by 1
        - If the 'TTL' field in a packet is reduced to 0, then the router will not forward the packet
            - At 0, the router will drop the packet, generate an ICMP message, and send it to the source-host as an IP datagram
                - The router uses the "32 bit source IP address" in the datagram to determine the source's IP address
                - `traceroute` uses the information in an ICMP message to calculate information such as 'RTT'
                - End hosts have no knowledge about the route, a packet takes, until it receives an ICMP (error) message from the router
    - In short, the 'time to live' field dictates how many hops a packet can survive in the Internet
        - When 'TTL' is reduced to 0, the router drops the packet, generate an ICMP error message, and send it to the source as an IP datagram
- Upper layer
    - Stores information about which upper layer protocol the payload needs to be delivered to
        - i.e. UPD, TCP, etc.
- Header checksum
    - Used to compute the checksum of the IP header

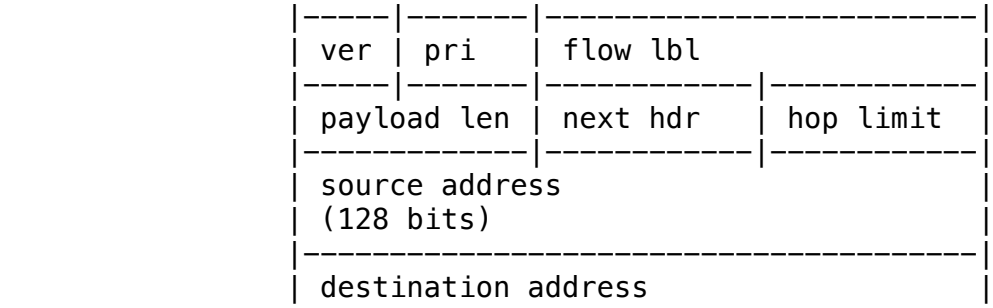
        - Helps detect if there is any bit error in the header
        - Note: TCP and UDP have their own checksum, which is based on its own payload and information in the IP header
            - This is another level of error detection, that is employed by the transport layer
    - 32 bit source IP address
        - IP address of the source host
            - "Who sent the packet, initially?"
    - 32 bit destination IP address
        - IP address of the destination host
            - "Who is the packet ultimately for?"
    - Options
- The minimum size of an IP header, without options, is:
  5 * 32 bits = 160 bits
    - Without options, there are 5 rows in the IP header and each row is 32-bits long
    - 160 bits = 20 bytes
- For a TCP segment, the minimum overhead is:
  20 bytes of TCP header + 20 bytes of IP header + Payload
    - This does not include options in the TCP/IP headers
    - Application layer overhead is variable
    - If you need to send a small chunk of data, this is a lot of overhead
        - The IP header is unavoidable, and must be included
            - TCP can be swapped for UDP
        - For instant messaging, it may be better to use UDP instead of TCP
- IP: How To Handle Packet
    - In the IP header, there is an upper layer field that:
        - Identifies the higher level protocol
            - Is 8 bits in length
        - Important for demultiplexing at receiving host
        - Most common examples:
            - 6 for TCP, 17 for UDP
- IP: Header: Checksum, TTL
    - Checksum (16 bits)
        - Particular form of checksum computed over packet header
        - Used by routers to detect bit errors
            - If the checksum in the header does not match the calculated checksum, then the packet is discarded
        - The checksum is re-computed at every router, because the 'TTL' field is decremented by 1
            - Thus, the header has changed and so has the checksum
            - In addition to packet forwarding, routers need to compare checksums, and compute checksums
    - Time to live (TTL) field (8 bits)
        - Decremented by 1 at each hop/router
            - Packet discarded if 'TTL' reaches 0
- IPv4 Address Exhaustion

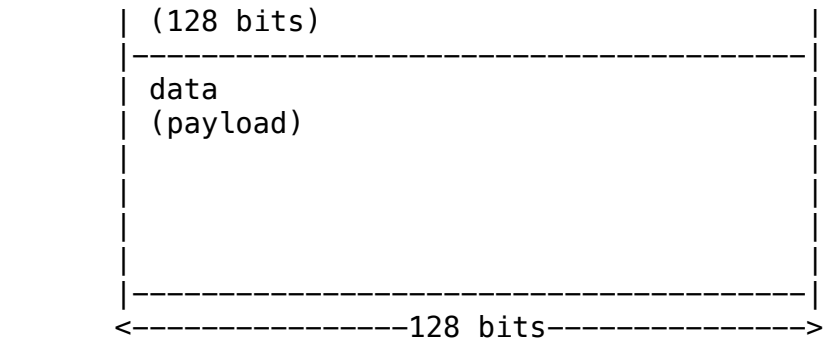
    - Only 32 bits are utilized to store the source and destination IP addresses
        - 32 bits is not a large number, it is roughly 4 billion
            - With the rapid increase in network devices, this is not sufficient
                - This problem is called IPv4 address exhaustion
    - Overtime, there are fewer and fewer public IP addresses that are available and can be allocated
        - There are two approaches to deal with this problem:
            - Make the IPv4 header bigger to accommodate IP address sizes greater than 32-bits
            - Some devices do not need to have a public IP address
                - i.e. In your home, each devices shares the same public IP address. Each device is assigned a private IP address. Network address translation is used to pair public and private IP addresses
                - This is the dominating solution to the IPv4 address exhaustion
    - IANA Unallocated address pool exhaustion: 03-Feb-2011
    - i.e. Table of Projected RIR Address Pool Exhaustion Dates

RIR	Projected Exhaustion Date	Remaining Addresses In RIR Pool (/8s)
APNIC	19-Apr-2011 (actual)	0.7331
RIPE NCC	14-Sep-2012 (actual)	0.9634
LACNIC	10-Jun-2014 (actual)	0.1950
ARIN	16-May-2015 (actual)	0.3064
AFRINIC	09-Jan-2019	2.6121

    - The 32 bit address space is soon to be completely allocated
- IPv6
    - The most straight-forward solution to the IPv4 address exhaustion problem
    - Is a different version of IP address
        - The format is entirely different; compared to IPv4
        - The most significant difference is the source and destination address fields
            - Instead of 32 bits, they are now 128 bits
        - Other differences include, but not limited to:

        - Flow label
        - Next header
- Initially motivated by IPv4 address exhaustion
    - Address length is 4x as big as IPv4
        - i.e. 128 bits VS. 32 bits
    - 2^128 is a VERY large number
        - "If you give an IPv6 address to every piece of sand, it will still not be exhausted..."
        - Giving an IPv6 address to every single device on Earth will not be a problem
- Fixes problems in IPv4, such as:
    - Header format helps speed processing/forwarding
    - Header changes to facilitate QoS
    - IPv6 datagram format:
        - Fixed length 40 byte header
        - No fragmentation allowed
            - It is assumed that the transport layer will do its job to discover the maximum segment size along the path
            - IPv4 supports fragmentation; at the expense of the end-host to reassemble the fragments into the original packet, and deliver it to the transport layer
            - Since fragmentation is not allowed, the header length is reduced, because the header does not need to store information associated with the identifier, fragment offset, and flags
        - No checksum
            - IPv6 leaves checksum-ing to the transport layer
            - The checksum isn't re-computed every single time the packet is forwarded
    - IPv6 has a priority field to specify the priority among datagrams in flow
    - In IPv6, you can associate a collection of datagrams to a flow
        - i.e. All packets corresponding to video streaming
            - These packets will be differently treated at the router
        - In IPv4, there is no notion of flow; every single packet is treated as a separate entity
- i.e. Datagram Format of IPv6 Header



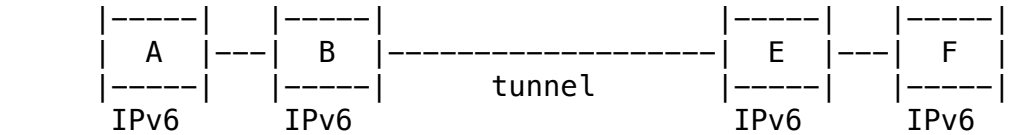


- 'pri' = 'priority'
  - Identify priority among datagrams in flow
  - Similar to 'type of service' in IPv4
- 'flow lbl' = 'Flow label'
  - Flow label identifies datagrams in "flow"
    - Concept of "flow" is not well defined
  - Allows the association of datagrams
    - i.e. All packets corresponding to streaming
- 'next hdr' = 'next header'
  - Identify upper layer protocol for data

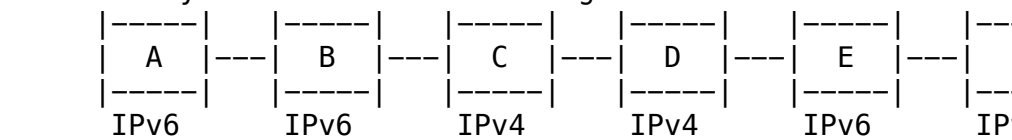
- Other Changes From IPv4
  - Checksum: Removed entirely to reduce processing time at each hop
  - Options: Allowed, but outside of header, indicated by "Next header" field
    - Instead of being part of the IP header, the optional field outside the header is indicated by the next header field
      - It is kind of a pointer to point where the optional fields belong/are
      - Allows routers to process headers faster, because they do not need to take account of those optional fields, unless it is actually necessary
  - ICMPv6: New version of ICMP
    - IPv6 comes with its own error messaging and signalling mechanism
      - It is called ICMPv6 protocol
    - Adds additional message/error types
      - i.e. "Packet too big"
    - Multicast group management functions
- Transition From IPv4 To IPv6
  - IPv6 is not widely adopted in the internet, because:
    - Not all routers can be upgraded simultaneously
      - If IPV6 is adopted overnight, then every single house needs to be upgraded, because IPv4 and IPv6 are not compatible
        - The headers for IPv4 and IPv6 are entirely different, making it impossible for them to communicate with each other
        - The IP address format, and size, for IPv4 and

IPv6 are completely different
- Getting people to update their routers, devices, etc. is nearly impossible
- How will the network operate with mixed IPv4 and IPv6 routers?
- New networks can take advantage of IPv6, by implementing it inside their local networks
  - i.e. Enterprises, educational institutions, etc.
- One way to facilitate compatibility between IPv4 and IPv6 is through tunneling
  - In tunneling, IPv6 is carried as a payload in an IPv4 datagram among IPv4 routers

- Tunneling Example
  - i.e. Logical View of Tunneling

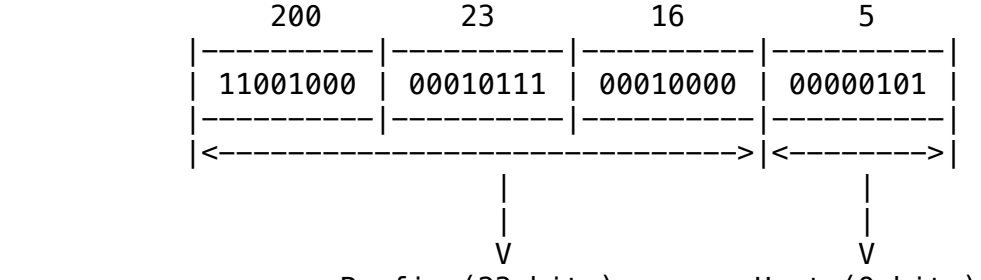


    - Assume that 'A' & 'B', are networks on their own island, and 'E' & 'F' are networks on their own island
    - If 'A' and 'B' want to communicate with each other, OR, 'E' and 'F' want to communicate with each other, then there are no problems, because all networks communicate in IPv6
    - Problems arise when processes on 'A' want to send data to processes on 'F'
      - If 'A' sends an IPv6 packet to 'F', then when the data arrives at the gateway between 'B' and 'E', the packet is no longer recognizable by the router
        - This is because the internal routers speak IPv4
  - i.e. Physical View of Tunneling



    - Assume that 'A', 'B', 'E', & 'F' are IPv6 networks, and 'C' & 'D' are IPv4 networks
      - 'B' and 'E' are gateway networks
        - They are IPv6 networks that interface with IPv4 networks
    - If 'A' wants to send data to 'F', then tunneling needs to be used, because the intermediate networks, 'C' & 'D', do not communicate in IPv6
      - When an IPv6 packet reaches the gateway router, 'B', it will be wrapped inside an IPv4 packet, and become the payload for that IPv4 packet
        - 'B' uses its own IP as the source IP address
        - The source host is now 'B', and the packet is of type IPv4
          - Regardless, the destination host will still

                    be able to realize the real sender of the packet
                - Upon reception at border gateway router 'E', it will retrieve the payload, and forward the original IPv6 packet to 'F', utilizing the IPv6 network
    - Tunneling is not efficient, because putting an IPv6 packet inside an IPv4 packet, and then taking it out requires computational resources
        - Routing IPv6 packets through an IPv4 network creates additional overhead
    - Tunneling is an inefficient solution to the IPv4 address exhaustion problem
        - Migrating the Internet from IPv6 to IPv4 is cumbersome
    - Another solution to the IPv4 address exhaustion problem is utilizing network address translation (NAT)
        - The success of NAT has made it harder for IPv6 to be adopted globally
            - However, IPv6 has been adopted locally
                - i.e. Education networks, corporate networks, etc.
    - Even though IPv6 is a very good idea, there are a lot of challenges in achieving global adoption
- IPv4 Addressing Recap
    - Note: Future lectures will stick to IPv4; IPv6 will no longer be discussed
    - IPv4 Address: Unique 32-bit number associated with a host/router interface
    - Represented via dotted-quad notation
        - Separate 32 bits into 4 bytes
            - Every byte represents a decimal number
            - Bytes are separated by a '.' (dot)
            - i.e. 200.23.16.5
        - This notation makes it easier for us (humans) to recognize and remember IP addresses
        - i.e. Diagram of IPv4 Address



    - Subnets
        - The 32 bit number can be divided into 2 parts:
            - The most significant bits correspond to the network address
            - The lower bits correspond to the host address
        - The division of network address and host address depends

  on where it is utilized
  - i.e. Breakdown of Subnet Addresses

11111111	11111111	11111110	00000000
255	255	254	0

- IP Addressing: Classful Addressing
  - In classful addressing, the division between the network address portion and the host portion is rigid
    - There is a total of 5 classes: 'A', 'B', 'C', 'D', 'E'
      - The classes differ based on the size, number of bits in the network address portion, and range of addresses covered
  - Class 'D' is used for multicast
  - Broadcoast address: 255.255.255.255
  - 127.0.0.1 is the loopback address
  - Problem: Class 'C' is too small, and class 'B' is too big
  - i.e. Table of Classful Addressing

Class	Left-most Bits	Start Address	Finish Address	Size of Network Number Bit Field
A	0xxx	0.0.0.0	127.255.255.255	8
B	10xx	128.0.0.0	191.255.255.255	16
C	110x	192.0.0.0	223.255.255.255	24
D	1110	224.0.0.0	239.255.255.255	N/A
E	1111	240.0.0.0	255.255.255.255	N/A

    - Class 'A':
      - The top 8 bits, out of 32 bits, indicate the network address portion
      - Starts with the leftmost bit at 0, and covers the range of addresses from 0.0.0.0 to 127.255.255.255
      - Since the host portion is `32 - 8`, the remaining 24 bits corresponds to the host field
        - Note: 8 is used to indicate which class 'A' network it is
        - There are `2^24` hosts that are possible within this network
      - Allocated to large corporations (i.e. Microsoft, Intel, etc.) in the early days of the Internet
        - Large corporations have large number of network hosts; they all share the same network address,

but differ in terms of host address
- Class 'B' and 'C' are used for smaller corporations that do not require a large number of network hosts
  - Compared to class 'A', class 'B' & 'C' have a smaller number of hosts within the same network
    - This is because of the number of bits in the 'network number' portion
    - Class 'C' has a smaller number of hosts than class 'B'
  - Class 'B' utilizes 16 bits to indicate the network portion
    - Thus, the total number of hosts within the same network is `2^16`
    - Class 'B' addresses start at 128.0.0.0, and end at 191.255.255.255
  - Class 'C' utilizes 24 bits in the network portion
    - The remaining 8 bits are utilized to identify the host within the same network
    - Addresses start at 192.0.0.0, and end at 223.255.255.255
  - Class 'C' is primarily used for home/residential networks
- Class 'D' is utilized for multicasting
- Class 'E' is reserved

- Classful addressing has pitfalls
  - A strict division of different classes of addresses is a bad idea, because lots of IP addresses may not be utilized by an entity
    - i.e. If class 'A' address range is assigned to Microsoft, and if Microsoft only uses 2^20 (1 million) IP addresses, then there are 15 million unutilized IP addresses; (15 million = 2^24 - 2^20)

- IP Addressing: Private Addressing
  - i.e. Table of Private Addresses

Class	Start Address	Finish Address	Blocks
A	10.0.0	10.255.255.255	10.0.0.0/8
B	172.16.0.0	172.31.255.255	172.16.0.0/12
C	192.168.0.0	192.168.255.255	192.168.0.0/16

    - Class 'A' allows a total of 16,777,216 hosts
      - Addresses in class 'A' start from 10.0.0 and end at 10.255.255.255
    - Class 'B' allows a total of 1,048,576 hosts
      - Addresses in class 'B' start from 172.16.0.0 and

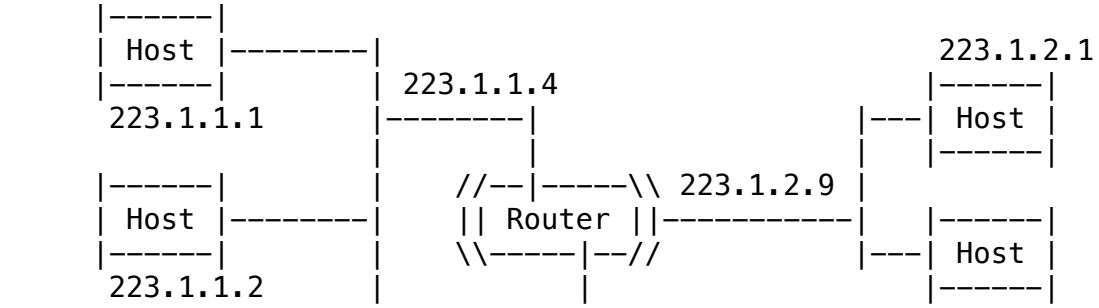
        end at 172.31.255.255
      - Class 'C' allows a total of 65,536 hosts
        - Addresses in class 'C' start from 192.168.0.0 and end at 192.168.255.255
  - Private addresses are free to use internally
    - They are utilized for private networks, and not for public networks
  - If you host a web server on your local computer, the private address that you allocated to the hosting machine cannot be used publicly for other users to communicate with
    - You have to use some tricks like network address translation (NAT) to map your private IP address, and port, to the public IP address that you obtain from an ISP, in order to host the service you want others to be able to reach
  - Private networks can have the same IP addresses without any conflict, because hosts on a private network can only talk to other hosts on the same private network
    - Machines on one private network cannot talk to machines on another private network
  - Private IP addresses are assigned by the network administrator or router
    - Depending on how many different types of hosts you want to have, you can have different classes
- IP Addressing: CIDR
  - Classes of addresses are no longer widely used
    - i.e. Class 'A', 'B', 'C', etc. are obsolete
  - CIDR: Classless InterDomain Routing
    - In CIDR, 32 bits are divided based on the actual number of hosts that need to be accommodated
      - The network address portion is decided in the 32 bits
    - Is widely adopted
  - Idea: Flexible division between prefix and host addresses
    - The idea behind CIDR is to break the barrier imposed by different classes
    - We want to have the flexibility of differing lengths in the network address field, and host address field
  - Intention: Offer a better tradeoff between size of the routing table and efficient use of the IP address space
  - Problem: The flexibility introduced by CIDR leads to variable length network address portions, which makes routing more complicated
    - Fortunately, there is a solution to this problem; it is called 'longest prefix match'
      - Even though the length of the network address field may differ, we treat the addresses as a block of address ranges, and only store 1 entry per block in the forwarding table
        - The destination port is also stored in the

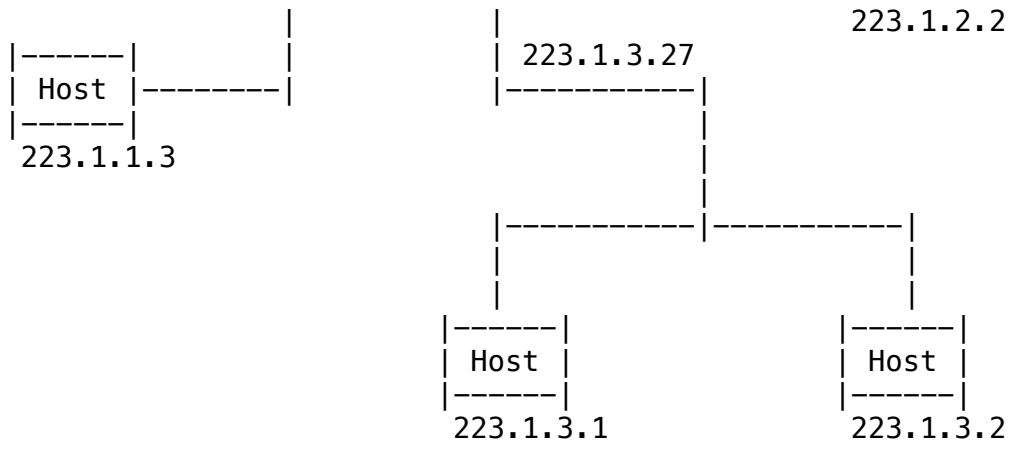
forwarding table
    - In contrast, routing is much simpler in classful addressing
        - i.e. If you want to reach Microsoft, all you need is a particular entry corresponding to one network that corresponds to Microsoft. Once the packet reaches the gateway router at Microsoft, it will reach the correct destination
- IP Addressing: CIDR Example
    - Fundamentally, there is a connection between how the addresses are allocated, and how routing is done; at least the complexity of routing
    - Using classful addressing for address allocation makes routing simpler, because all the addresses are already (kind of) aggregated
        - Aggregation is done through default ranges
    - In CIDR, routing is more complicated, because of variable length network addresses
    - Suppose a network has 90 computers. What is the most efficient number of bits for prefix part and host parts?
        - The goal is to decide how many bits to use for the network portion, and how many bits to use for the host portion
        - Note: The prefix part refers to the network portion
        - The answer is 7 bits need to be used for the host part
            - Because 2^6 = 64, and 2^7 is 128
                - 64 is not big enough
                - 2^7 allows for 128 computers/IP addresses
        - The remaining 25 bits are used for the network prefix
            - `25 = 32 - 7`
        - The assumption is that the computers will share the same prefix, but they will differ in the last 7 bits
            - The computers will share the same network prefix to facilitate aggregating those range of IP addresses in the router
            - As long as the computers belong to the same network, they share the same prefix and only 1 entry is required in the forwarding table, because the computers share the same gateway router with the Internet
        - Example of an IP address for this network: 223.1.1.0/25
            - The network address is: `223.1.1.0`
                - All hosts, in this network, share this prefix of 25 bits
                    - i.e. Possible host IP addresses:
                        - 223.1.1.0.00000000
                        - 223.1.1.0.00000001
                        - 223.1.1.0.10100100
                        - 223.1.1.0.11111111
            - The number of bits in the network prefix is: `25`

                - The number after the slash indicates how many bits are in the network prefix
                    - The slash notation is compact, and is widely utilized in network administration

- March 3rd, 2021
    - Recap From Last Class
        - There are 2 types of addressing mechanisms:
            1. Classful Addressing
                - Consists of address ranges of different sizes
                    - i.e. Class 'A', 'B', & 'C'
                    - i.e. Class 'D' is utilized for multicast
                    - i.e. Class 'E' is specially reserved
                - Was used in the early stage of the Internet, where IP addresses were abundant and hosts were scarce
                - Problem with classful addressing:
                    - When you allocate a large chunk of IP addresses for a corporation, like class 'A', there is a good chance that the addresses will be under-utilized by the organization
                        - The remaining addresses cannot be utilized by other people outside the organization
            2. CIDR: Classless InterDomain Routing
                - Idea: Flexible division between prefix and host addresses
                - Intention: Offer a better tradeoff between size of the routing table and efficient use of the IP address space
                - Instead of having a rigorous partition of network addresses into 8, 16, or 24 bit prefix portion, and the host portion, the bits are divided in an arbitrary way
                    - Allows for a larger number of prefix bits, so that the corresponding number of hosts in the network can be significantly less than 2^24, or 2^16, or 2^8
                - Provides a tradeoff between not having to allocate large blocks, but still have some notion of aggregation
        - There is an intrinsic relationship between the addressing mechanism and longest prefix matching in routers
        - In the IP forwarding table, the forwarding entries store information in blocks of IP addresses, and corresponding output port for any IP packet with destination address for that range
        - If you can allocate IP addresses in a block manner (i.e. All McMaster network hosts have a public address in a continuous range), and facilitate some kind of aggregation of IP addresses, you can use some long prefix to represent the block of network hosts within McMaster

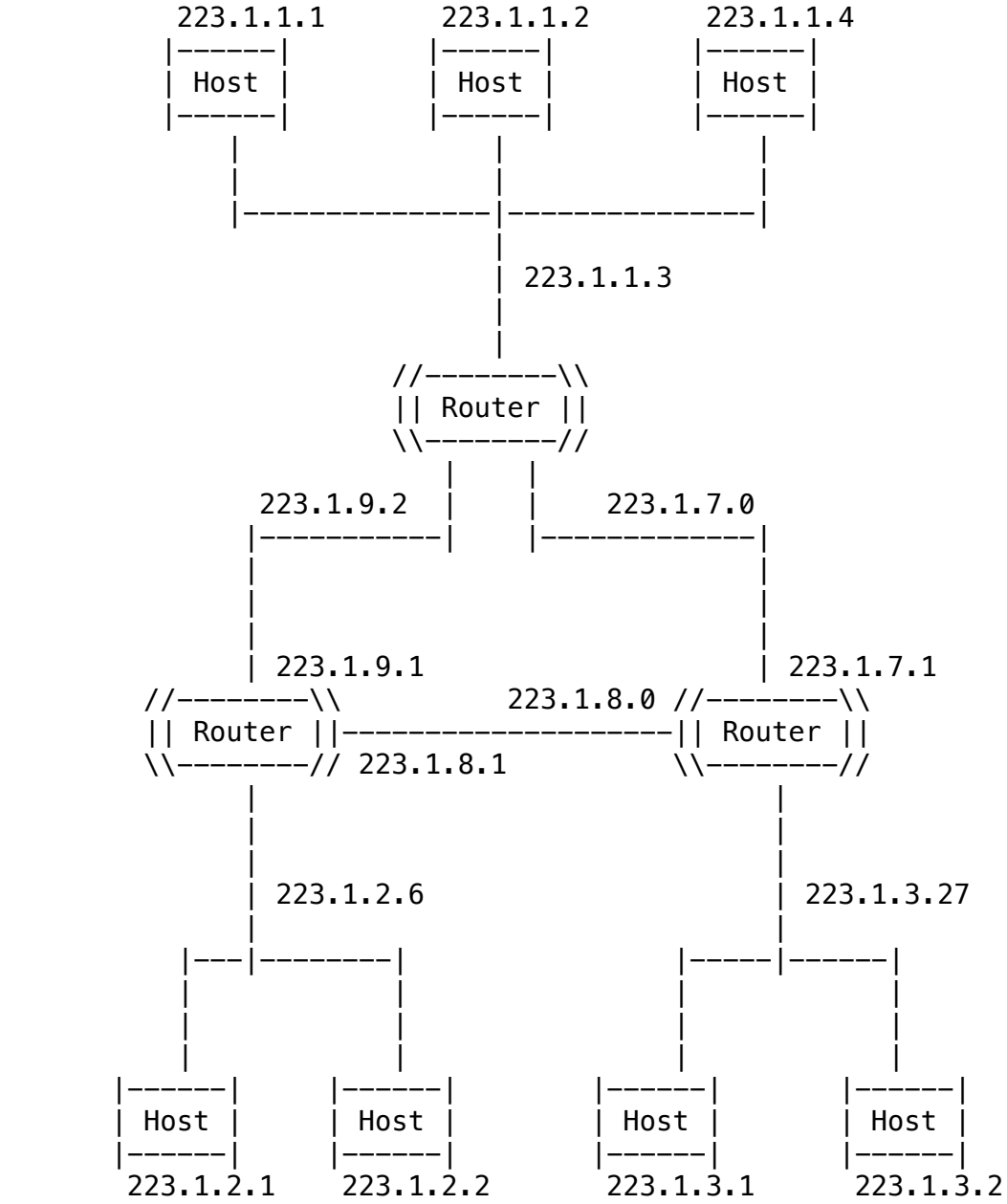
        - Since McMaster's hosts are likely to reside in the same physical location, the routes from remote hosts tend to be the same until you reach the campus' network
            - Thus, if you can allocate addresses in a block manner to physical hosts that are in physical proximity, then you can facilitate address aggregation in the router so that there are fewer routing table entries
                - This makes IP lookups faster
    - CIDR is one way to not have to allocate large blocks, but still have some notion of aggregation
        - The 32 bit IP addresses are divided into network address and host address
        - The network hosts within an organization tend to share the same network address portion, or the same prefix
            - But, they differ in the host address portion
    - Routing tables and forwarding tables are synonymous
- IP Addressing: CIDR Example
    - Suppose a network has 90 computers. What is the most efficient number of bits for the prefix and host part?
        - Allocate 7 bits for host addresses
            - Since `2^6 < 90 < 2^7`
        - Remaining `32 - 7 = 25` bits as network prefix
            - If the network only has 90 computers, then it is a good idea to reserve the top 25 bits for the network prefix, and host addresses will use 7 bits
        - i.e. 223.1.1.0/25
- Subnets Example 1
    - Subnets are important when you want to broadcast information within your local area network (LAN)
        - Dynamic host address allocation allows you to reach all network interfaces in the local area network (LAN)
            - In a family organization LAN the devices tend to reside on the same local area network
        - Virtual local area networks provide a logical partition of the LAN
            - But from a (pure) physical connectivity point-of-view, the host residing on the same LAN are connected by switches and hubs
                - They connect to other networks through routers
    - i.e. Diagram of Subnets





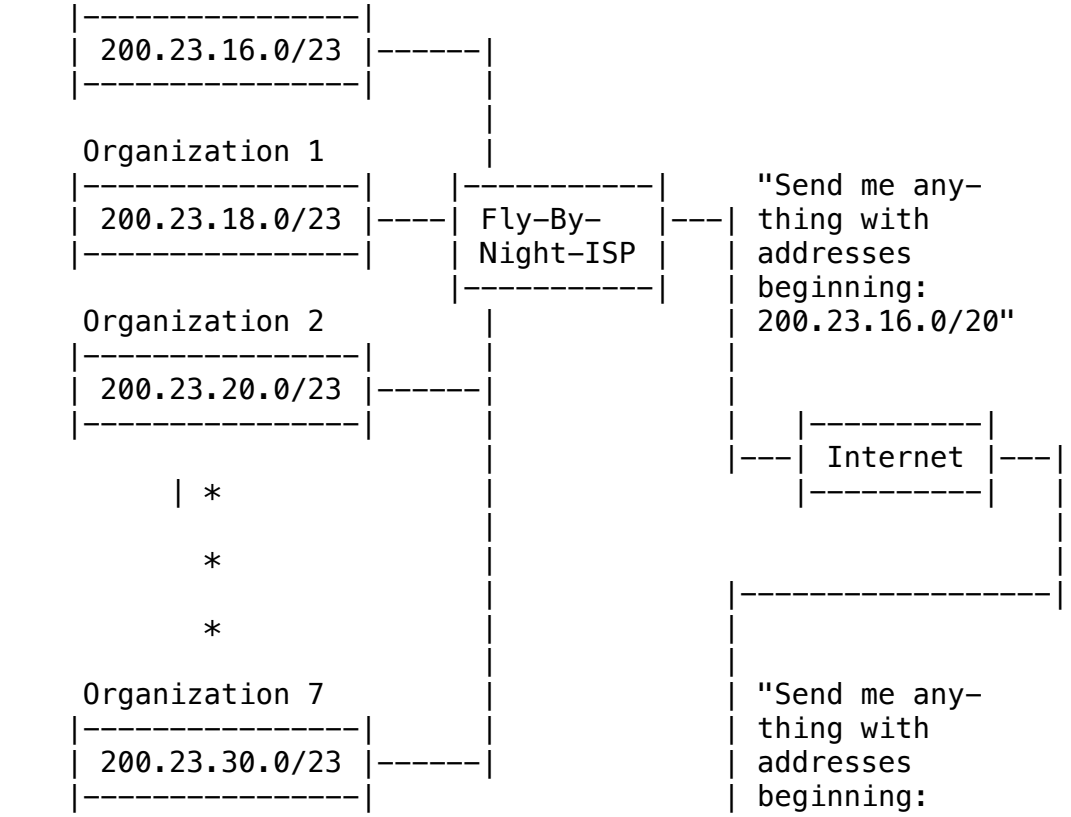
- The subnets have a 24 bit network prefix
- In this subnet, there's a collection of network hosts that are connected together, and form some kind of LAN
- Different LANs may be connected through the router
- The router has 3 interfaces
  - Each interface connects to a local area network
- Local area networks (LANs) are referred to as subnets
- A subnet is a unit of a collection of network hosts that can be reached through broadcasting
  - If any host sends a broadcast message, it will be received by all the other hosts within the same subnet
    - However, the broadcast message will not be forwarded by the router that connects multiple subnets
  - Subnets are the logical units of the broadcast domain
- Question: How many subnets are in the diagram above?
  - Answer: There are a total of 3 subnets
    - To distinguish subnets, look at where the interface ends at the router
    - Each interface connects to a single subnet
- Question: What is the IP address notation of each subnet, in the diagram above?
  - Given the IP addresses of hosts within each of the subnets, each block of IP addresses can be represented via slash notation
  - Answer:
    - The network address portion of all subnets, in the diagram above, are 24 bits long
      - This means that the hosts in this subnet share a common prefix of 24 bits, starting from the most significant bit
      - The slash notation is a compact notation that can be used to represent a range of IP addresses
    - The subnet on the LEFT has 3 hosts

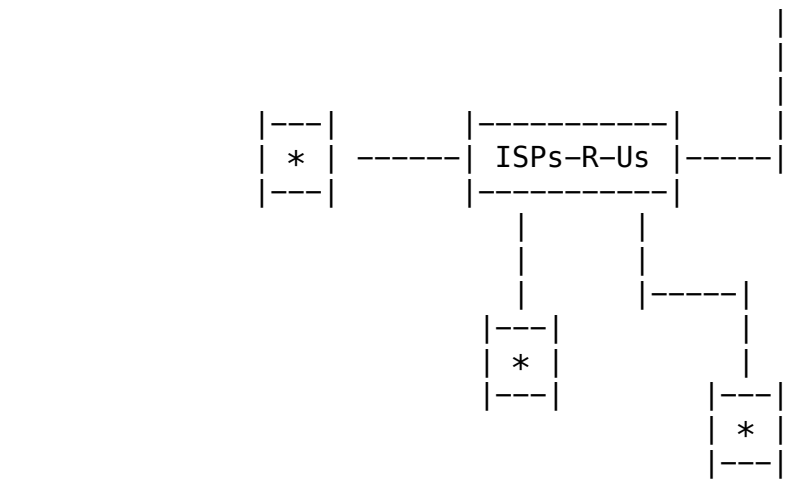
        - It can be represented via the following slash notation: 223.1.1.0/24
          - `223.1.1.0` is the network address
          - `24` indicates the number of bits in the network address portion
      - The subnet on the RIGHT has 2 hosts
        - It can be represented via the following slash notation: 223.1.2.0/24
      - The subnet on the BOTTOM has 2 hosts
        - It can be represented via the following slash notation: 223.1.3.0/24
- Subnets Example 2
  - i.e. Diagram of Subnets



    - In this diagram, multiple subnets are interconnected through different routers
    - Question: How many subnets are in the diagram above?
        - Answer: In total, there are 6 subnets. They are:
            - 3 subnets correspond to the LANs that consist of network hosts - they are:
                1. Hosts connected to the top router
                2. Hosts connected to the bottom-left router
                3. Hosts connected to the bottom-right router
            - 3 subnets correspond to the link that connects the interfaces of the 2 routers - they are:
                1. The link between the top and bottom-left router interfaces
                2. The link between the top and bottom-right router interfaces
                3. The link between the bottom-left and bottom-right router interfaces
                - Note: These are considered subnets because if you send a broadcoast from one network interface to another, the corresponding router will receive it, but it will not forward it to other routers/hosts
        - The general rule is: To determine the number of subnets, you take out the router and look at how many disconnected components are left in the network
    - Question: What is the IP address notation of each subnet, in the diagram above?
        - Answer: The slash notation for each subnet is:
            - For the top-most subnet, consisting of hosts, it is: 223.1.1.0/24
            - For the bottom-left subnet, consisting of hosts, it is: 223.1.2.0/24
            - For the bottom-right subnet, consisting of hosts, it is: 223.1.3.0/24
            - For the link between the top-most and bottom-left router interfaces, it is: 223.1.9.0/24
            - For the link between the top-most and bottom-right router interfaces, it is: 223.1.7.0/24
            - For the link between the bottom-left and bottom-right router interfaces, it is: 223.1.8.0/24
        - Note: The network address portion corresponds to the top, most significant, 24 bits
        - Note: It is acceptable to not put the last zero at the end of the slash notation. For the subnet, we only care about the network address portion
- Hierarchical Addressing: Allocation
    - Public IP addresses are assigned in a hierarchical manner
        - The ICANN is at the very top
    - Internet corporation for assigned names and numbers (ICANN) allocates chunks of IP address to:

    - Regional Internet Registries (RIR)
  - Regional Internet Registries (RIR) assign block of addresses to:
    - Large institutions (ISPs)
      - i.e. Rogers, Bell, Cogeco, etc. may receive a big chunk of IP addresses that they further assign to their hosts
  - ISPs assign addresses to:
    - Hosts
  - From ICANN's point of view it does not care how RIR allocates their block of IP addresses
    - Similarly, RIR does not care how ISPs assign IP addresses
  - Hierarchical addresses makes it easier for aggregation of IP addresses, because the IP addresses for each organization, or institution, falls into a continuous range
    - This allows the IP addresses to share common prefixes
      - As a result, routers only need a single entry in their forwarding table
- Hierarchical Addressing: Route Aggregation
  - Hierarchical addressing allows efficient advertisement of routing information
    - If the block of addresses aggregate nicely, then we can use a simple representation in the forwarding table
  - i.e. Diagram of Route Aggregation





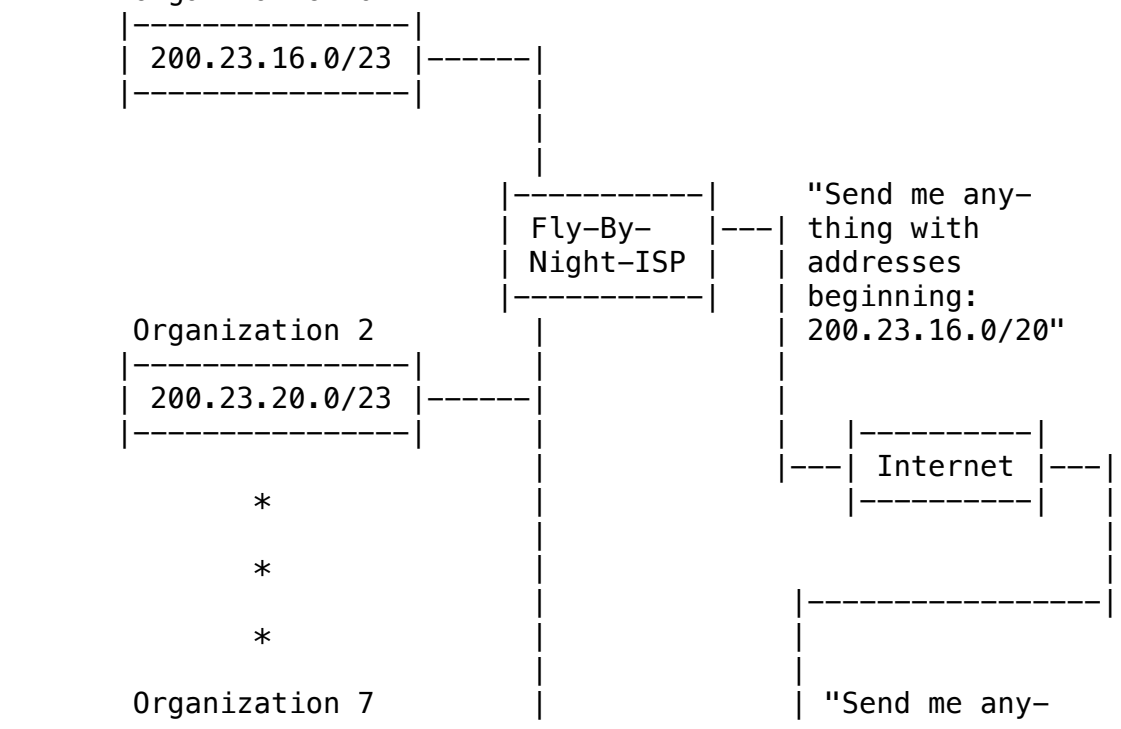
- There are 2 ISPs:
  - "Fly-By-Night-ISP"
  - "ISPs-R-Us"
- 'Fly-By-Night' provides services to 8 organizations
  - Each organization has its own block of IP address with a different network address portion
    - The prefix length is 23; this means that the top 23 bits are in common
    - Only the top 7 bits matter for the representation of the subnet address
  - Note: An asterisk ('*') represents a customer
- If a customer is on a different ISP, and wants to connect to one of the organizations, then the routing table on the customer's ISP will have a single entry that aggregates blocks of addresses for all of the (different) organizations
  - Depending on the organization that the customer wants to reach, the packet may take a different route, but each route will share the same gateway router that contains a single entry that aggregates the blocks of addresses that correspond to the (different) organizations
- From a routing point of view, if a customer is on the same ISP as the organization, then all packets will be routed through the common ISP
  - The routing table entries in the customer's network will have a single entry that aggregates all the blocks of addresses for different organizations
    - The routing table does NOT need to have an entry for each host inside an organization, because all hosts within an organization are likely to follow the same path
      - Thus, they will share a single IP forwarding table entry within their provider ISP
- From the customer's point of view, regardless of which host/organization it wants to address, you can aggregate

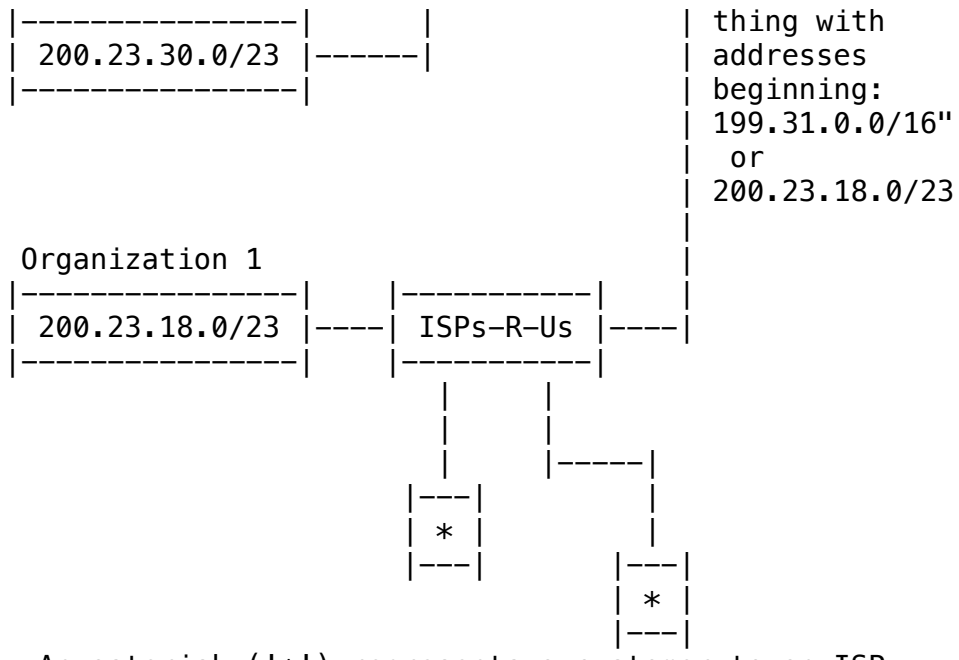
them together into a single entry
  - To aggregate them, simply take the longest common prefix across the entire address block
    - i.e. In the diagram above, the longest common prefix for all organizations is '... 0001'. This is because:
      X.X.16 = ... 0001 0000
      X.X.18 = ... 0001 0010
      ...
      X.X.30 = ... 0001 1110
      The first couple of bits are the same, and the remaining 4 bits are different. Since the LAST 4 bits are different, the network address portion is 20
  - The collection of organizations in the diagram above can be represened by the following slash notation: 200.23.16.0/20
    - The organizations share the top 20 bits, but differ for the remaining 12 bits
    - 20 bits is used instead of 23 bits
- From the (Internet) network backbone's point-of-view, the organizations will share the same routes, because the organizations are subscribed to the same ISP
  - As a result, 'Fly-By-Night' only needs to broadcast the aggregated address range, which is: 200.23.16.0/20
  - There is no need to specify the IP address for each organization
    - This is another level of aggregation, at the core routers, which reduces the number of routing table entries
- The provider 'ISPs-R-Us' has its own customer's organizations, and is capable of aggregating the IP address ranges for its customer network
  - The ISP will broadcast a larger block to the rest of the network; it will tell the other routers and ISPs that, "You can send me anything with the address range: 199.31.0.0/16"
    - The longest prefix is 16
- The address blocks, '200.23.16.0/20' && '199.31.0.0/16', become separate entries in the forwarding table in the core routers
- If the ISPs, 'Fly-By-Night' && 'ISPs-R-Us' have a similar address range, due to geographical proximity, then their IP address range can be aggregated and stored as a single entry in the next level/tier ISP's forwarding table
  - The tier 2 ISP will broadcast an address block to its peers and providers
  - This is another level of aggregation, which leads to

fewer entries in the forwarding table
- Note: The internet is organized in a hierarchical manner, which leads to address allocation also being done in a hierarchical way
- Address allocation is not done arbitrarily
- This facilitates the possibility of being able to aggregate many-many IP addresses together, and only have a unique forwarding table entry associated with them

- The purpose of aggregation is to reduce the number of routing table entries in the forwarding table
- i.e. All hosts within an organization do NOT need their IP address within a forwarding table. Only 1 host needs to have its IP in the table. Subsequent hosts can be discovered from the former hosts' IP address
- The rule for aggregation is to take the longest common prefix of all IP addresses in a block/organization
- Grouping IP addresses makes routing more efficient, and occupies less space in the routing table
- i.e. In the diagram above, the organization's ISP only needs to maintain 1 IP address for each organization, instead of the hosts within their organization. This reduces the number of routing table entries

- Hierarchical Addressing: More Specific Routes
- The following example assumes that an organization decides to switch ISP providers due to pricing issues
- i.e. Diagram of Route Aggregation





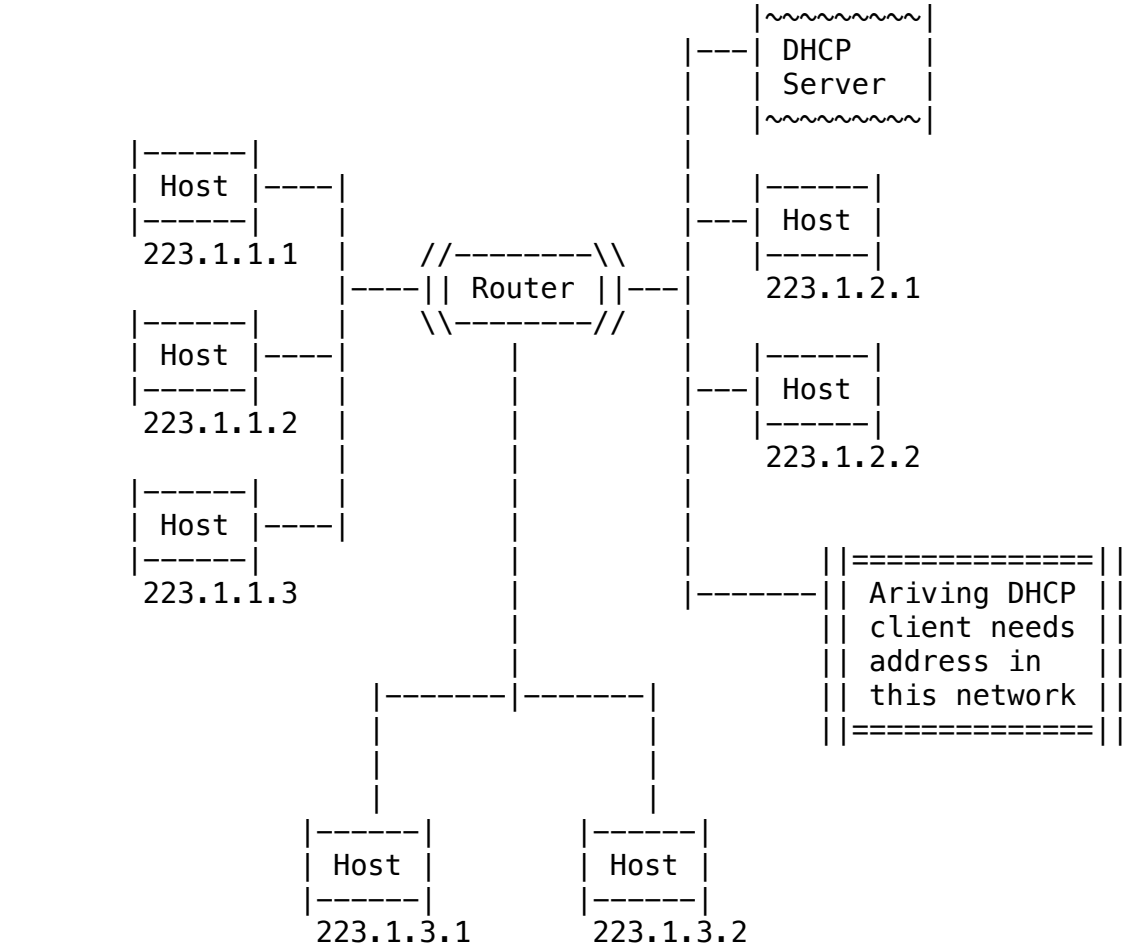
- An asterisk ('∗') represents a customer to an ISP
- Organization #1 decides to switch ISP providers from 'Fly-By-Night' to 'ISPs-R-Us', because 'Fly-By-Night' charges too much money
- After organization #1 switches ISP providers, it continues to maintain its own block of IP addresses
  - There is no reallocation of IP addresses
- The block of aggregated IP addresses that are stored in the forwarding table for 'Fly-By-Night' continues to remain the same
  - It sends '200.23.16.0/23' to higher tier ISPs
    - However, there is a hole in this aggregated block of IP addresses, because part of this block contains the range of IP addresses that corresponds to organization #1, which is no longer a customer of 'Fly-By-Night-ISP'
- Organization #1 will have its own entry in the forwarding table for 'ISPs-R-Us'
  - All packets headed for organization #1 will be routed through 'ISPs-R-Us'
- Even though there is a hole in the aggregated block of IP addresses for 'Fly-By-Night', we do not need to divide the address range into each organization's IP address
  - If we add a specific entry for organization #1 in the routing table for 'ISPs-R-Us', then we can use longest prefix match to properly direct packets headed for/to organization #1
    - The specific entry in the router table makes it impossible for packets headed to organization #1 be directed to 'Fly-By-Night', because the

longest prefix match will always direct the packet to 'ISPs-R-Us'
- The routers in the backbone network will have aggregated IP addresses from 'Fly-By-Night-ISP' and 'ISPs-R-Us' in their forwarding table
  - By the principle of longest prefix match:
    - A packet that is headed toward organization 0, or 2 - 7, will be directed to 'Fly-By-Night-ISP'
    - A packet destined for organization #1, will be sent to 'ISPs-R-Us', because the routers will contain a specific entry for organization #1 in their forwarding tables
      - i.e. 23 bits in VS. 20 bits (in the prefix)
- To summarize, as long as 'ISPs-R-Us' contains a specific entry in its routing table for organization #1, then the aggregated block of addresses in the forwarding table for 'Fly-By-Night-ISP' does not need to be altered due to the principle of longest prefix match
  - On the other hand, if there are a lot of discontinuous blocks of IP addresses, then each IP address needs to have its own entry in the routing/ forwarding table
    - This lowers the efficiency by increasing storage and computational complexity

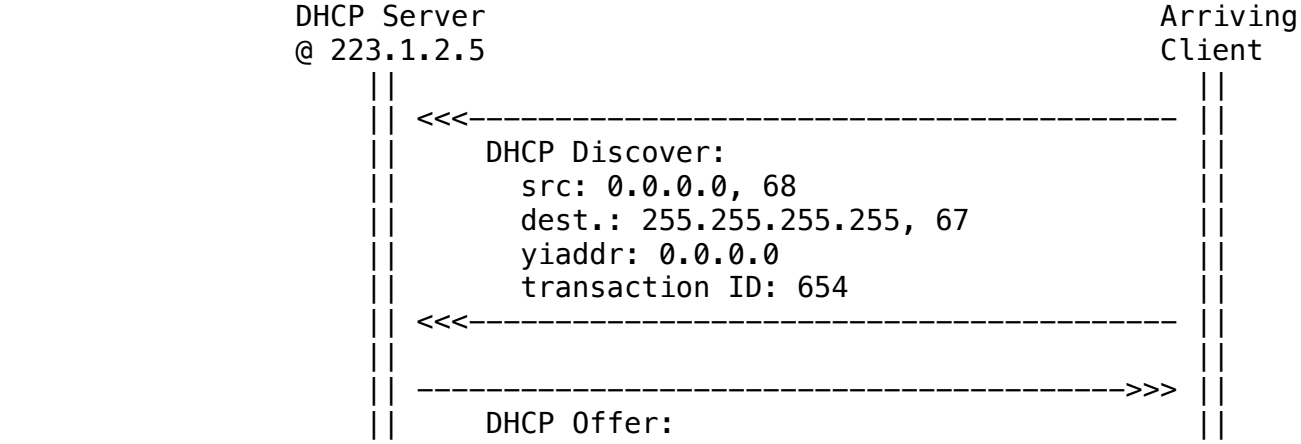
- IP Addresses: How To Get One?
  - Question: How are IP address allocated to hosts?
    - Answer: The IP address of the host can be set once the IP address has been allocated to you
      - You can manually configure the network interface
    - Answer: Hard-coded by system admin in a file
      - Windows: Control Panel -> Network -> Configuration -> TCP/IP -> Properties
      - UNIX: /etc/rc.config
  - Another way to get an IP address is by utilizing a protocol called Dynamic Host Configuration Protocol (DHCP)
    - DHCP is responsible for allocating private addresses to network hosts, in a dynamic manner
      - This is done in a completely automatic way
        - "Plug and play" approach
    - When a host joins a network, it will be able to get an IP address via DHCP
      - DHCP dynamically gets an IP address from the server
        - If the host leaves the network, the IP address can be reused by other hosts
- DHCP: Dynamic Host Configuration Protocol
  - DHCP utilizes UDP/IP for sending and receiving
    - DHCP messages are encapsulated in UDP, and the resulting datagrams are encapsulated in IP
    - Note: Since DHCP uses UDP, there is a chance that the datagrams get lost in the network. In this case,

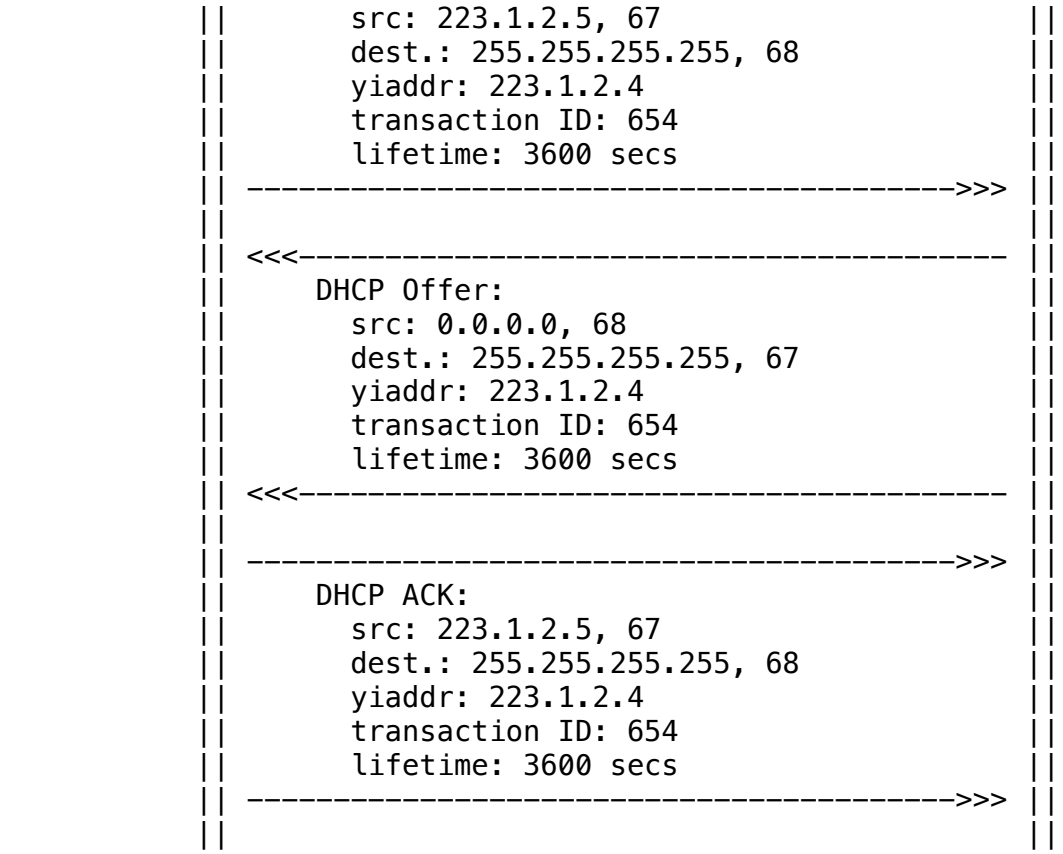
the host will have to retransmit its request 'msg'
- Goal: Allow hosts to dynamically obtain IP addresses from network server when they join the network
  - The DHCP server is responsible for allocating IP addresses, and keeping track of the usage of the corresponding IP addresses
  - IP addresses are leased to hosts
    - Hosts do not get permanent IP addresses allocated to them
    - Each allocated address has an expiration time
      - Upon expiration, a host can renew its lease, and continue using the address
      - If the host does not renew its lease, then the corresponding IP address will be released, and it can be allocated to other hosts
  - Allows reuse of addresses
    - Only hold addresses while connected, or "on"
  - Support for mobile users who want to join network
    - More shortly
  - Consists of a number of message exchanges
    - The first 2 messages are optional
    - The remaining 2 messages are mandatory for DHCP to work properly
- DHCP messages overview:
  - Host broadcasts "DHCP discover" msg[optional]
    - This is the first message, and is sent by the host
      - i.e. When a computer first boots up, and attempts to connect to the LAN, the host will broadcast the DHCP discovery message
  - DHCP server responds with "DHCP offer" msg[optional]
    - This is the second message, in response to the first message, and is sent by the server
  - Host requests IP address: "DHCP request" msg
  - DHCP server sends address: "DHCP ack" msg
    - Upon reception of this message, the host will use the allocated IP address, and configure it to its network interface
- DHCP Client Server Scenario (1)
  - DHCP is necessary because hosts need to have an IP address allocated to them
    - Without an IP address, the host will not be able to communicate with other hosts
      - i.e. If you don't have an address, how can other people address you?
  - The IP protocol is used to exchange information between the host that requests an IP address with the DHCP server
    - This is accomplished via broadcast messages
    - Before a particular host is assigned an IP address, for the purpose of delivering data, a broadcast address can still be utilized

    - i.e. Diagram of Network & DHCP Server



        - The slash notation of:
            - The left subnet is: 223.1.1.0/24
            - The right subnet is: 223.1.2.0/24
            - The bottom subnet is: 223.1.3.0/24
        - The DHCP server is running on a particular LAN
        - An arriving DHCP client needs to request an IP address
- DHCP Client Server Scenario (2)
    - i.e. Diagram of Client Connecting To DHCP Server





- 'yiaddr' = Your IP Address
- The DHCP server operates on the IP address: 223.1.2.5
- The first 2 DHCP messages are optional
  - The DHCP Discover & DHCP Offer messages are a pair
- The last 2 DHCP messages are mandatory
  - The DHCP Request & DHCP ACK messages are a pair
- Initially, the client has no information about where the DHCP server is, and what its address is
  - Thus, the initial message would have to utilize the broadcast address
    - Any packet with a broadcast address as its IP destination address will be received by all the hosts in the local area network (LAN)
- In the IP header of the discover message, sent by the host to the DHCP server, the source IP address is (initially): 0.0.0.0
  - Even though the IP address is all zeros, it corresponds to the IP address of the host
    - Currently, the host has no allocated IP address
  - The port number is 68
    - This is used by the client
  - Note: The discover message is a UDP datagram
- In the IP header of the discover message, the destination address is the broadcast address
  - This is all 1's in the 32-bit IP address

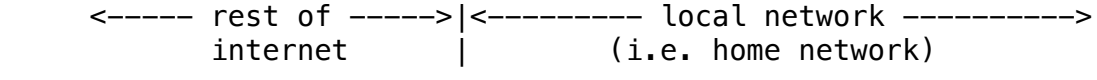
    - The port number is 67
        - This is used by the DHCP server
- In all DHCP messages, there is a field called 'yiaddr'
    - This corresponds to the host's IP address
    - Eventually it is utilized to store the address that has been assigned to the client
- The transaction ID is used to keep track of the discovery and response messages
- Upon reception of the DHCP Discover message, the DHCP server will respond with a DHCP Offer message
    - The DHCP Offer message needs to be broadcasted because the requesting host does not have an IP address assigned to it
- The DHCP Offer message contains:
    - IP address of the DHCP server
    - The 'yiaddr' field
        - This is the IP address that is allocated by the DHCP server, and assigned to the arriving client
    - An expiration timer, called 'lifetime', that terminates the offer if it is not taken within the time frame
- The DHCP Offer message is not sufficient; it is just an offer
    - The client needs to send a confirmation to the DHCP server, indicating whether it has chosen to accept or decline the offer
        - Thus, subsequent messages need to be exchanged
- The DHCP Request and DHCP ACK messages are mandatory
    - A client can start from a DHCP Request message
        - i.e. In renewal messages where the client already has an IP address allocated to it
- In the DHCP Request message, the client will take the IP address in the 'yiaddr' field
    - Since the client hasn't been allocated an IP address by the DHCP server, the IP address of the client is still all zeros: `0.0.0.0`
- The DHCP Request message increments the transaction ID
    - Each pair of messages share the same transaction ID
        - DHCP Discover and DHCP Offer are a pair
        - DHCP Request and DHCP ACK are a pair
- The DHCP Request message allows the client to explicitly request the IP address that was indicated by the server in the 'yiaddr' field
- The DHCP ACK message is sent by the server to the client to acknowledge the DHCP request message
    - The 'yiaddr' field remains the same
    - Broadcast address is used as the destination
    - The source address is the IP address of the DHCP server
    - The 'lifetime' field specifies the length of the

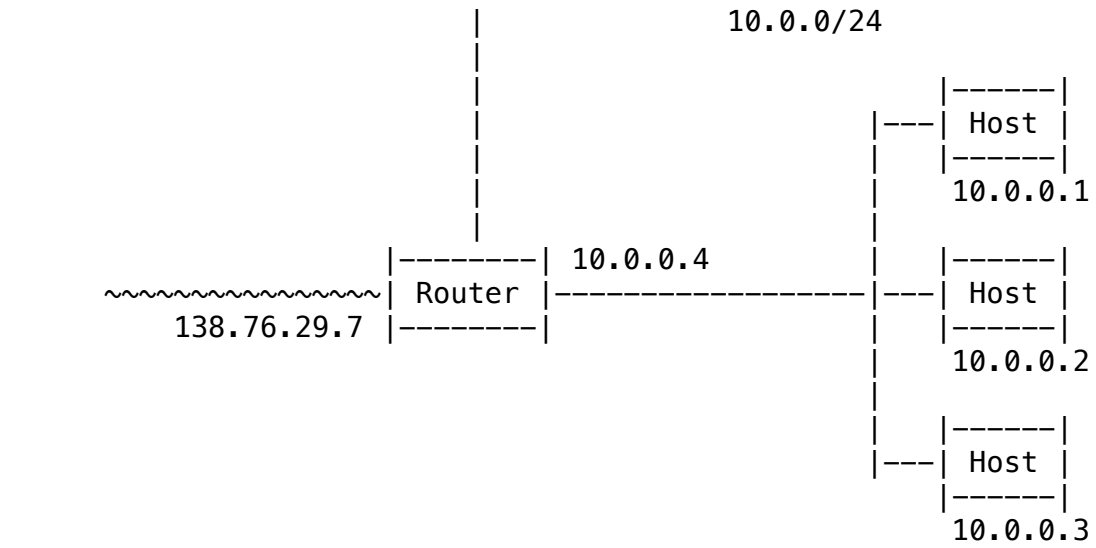
lease for the IP address
- i.e. How long the client can hold on to the IP address, before it expires?
- When the client receives the DHCP ACK message, it can now safely configure 'yiaddr' as its own IP address, and proceed with communication
- Most clients, especially mobile clients, need to go through this DHCP message exchange process when they first join a local area network (LAN)
- DHCP messages are necessary because there might be multiple DHCP servers in the local area network (LAN), and each server can make an offer to the client
- In this case, the client has to decide which offer to take, and must specify it in a subsequent message
- Thus, DHCP Offers are not sufficient for the successful allocation of IP addresses

- DHCP: More Than IP Addresses
- In addition to assigning IP addresses and returning allocated IP addresses on a subnet, DHCP can do other things such as:
- Indicate what is the IP address of the gateway router
- This is the first hop router for the client
- This allows the client to configure its own routing table for subsequent communication
- Forwarding tables on network hosts can only have 2 situations
- The host can send something within the LAN
OR
- The host can send something out of the LAN
- The subnet mask indicates what hosts can be directly communicated within the LAN, without going through a router
- If the destination IP address is outside the subnet, then the next hop the packet is forwarded to is the gateway router
- Provide information about the local DNS server
- Its name, IP address, and subnet mask
- The subnet mask separates which hosts belong in the same subnet, and which are outside
- Network mask
- Indicates the network versus host portion of address
- Typically, for a local area network (LAN), the router runs the DHCP server

- March 5th, 2021
- DHCP: More Than IP Addresses
- The DHCP protocol is utilized to dynamically allocate addresses to network hosts
- In addition to address allocation, it also provides:
- Information about the IP address of the first hop router

for the client
        - The name and IP addresses of the DNS servers
        - A network mask, which is utilized by the network host to determine whether a particular destination host is within the same local area network or not
- NAT: Network Address Translation (1)
    - NAT is implemented in your home gateway router
    - Motivation: A private (home) network uses just 1 IP address as far as outside world is concerned
        - Network address translation (NAT) helps solve the IPv4 address exhaustion problem
            - Every host on a network does not need its own public IP address
            - No need to be allocated range of addresses from ISP
        - Home networks utilize private IP addresses for the network host
            - Hosts are still able to access the Internet
        - From a cost reduction point-of-view, NAT is useful because you do not need to pay for every public IP address that you utilize
        - Can change addresses of devices in a private network without notifying outside world
            - If you have multiple devices and a single IP address, NAT allows you to allocate private addresses to individual network hosts without bothering your ISP
            - Depends on the reachability of the devices
            - Overtime, different devices can be assigned different IP addresses
                - Typically, this is accomplished through DHCP
        - Can change ISP without changing addresses of devices in a private network
            - If you change your ISP the private IP address will remain the same, and you can internally utilize it
                - However, the public IP address will change
        - Devices inside not explicitly addressable by or visible to outside world
            - From a security point-of-view, because the devices are not explicitly addressable from the outside world, the network is more secure because external devices cannot directly connect to internal devices
            - This is a benefit in terms of security
- NAT: Network Address Translation (2)
    - Question: How can internal devices communicate with outside networks?
        - Answer: Network address translation (NAT)
    - i.e. Diagram of Private Network Interacting With Public Network

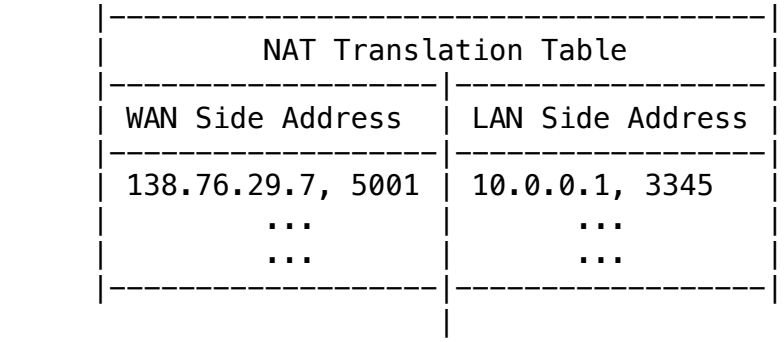


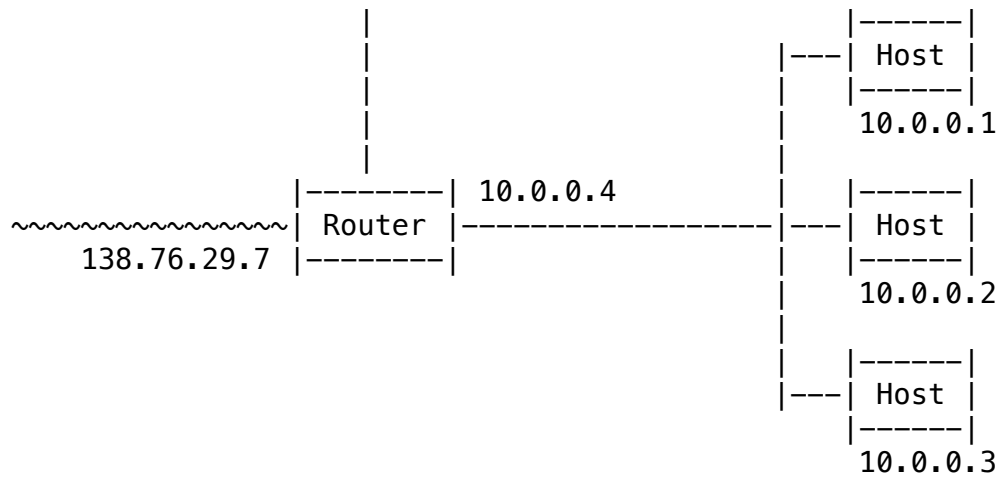


- This diagram depicts a private network
  - It is a local area network that shares a (very large) private address block of 10.0.0/24
    - `10.0.0/24` is a class 'A' address range
    - Internally, all of the devices utilize some address in this range
  - In this network, all the devices will share the following common public IP address: 138.76.29.7
    - This is the ONLY public IP address
- The network address translator (NAT) sits on the router, and translates between the private network address and the public IP address
- All datagrams that leave the local area network (LAN) are going to use the same public IP address as their source address
  - Although, they will have different port numbers
- From the Internet's point-of-view, all packets coming into the local area network (LAN) are addressed to the same public IP address that is used by all hosts
  - However, when the packets arrive at the LAN, the NAT will map the single public IP address to its respective private host in the LAN
- What if I have more hosts than available public IP addresses?

- NAT: Network Address Translation (3)
  - Assume you have a datagram that originates from one of the hosts in your local area network (LAN)
    - At the NAT gateway, it will replace the tuple in the IP and transport layer header that corresponds to source IP address and port number with a different tuple that corresponds to your public IP address and a port number that is uniquely allocated by the NAT
      - The source IP address and port number are recorded in the NAT's table, before they are replaced

        - Upon reception of an incoming datagram (i.e. An acknowledgement to a TCP segment), the NAT will replace the NAT IP address and new port number with the original source IP address and port number, that is stored in the NAT's table
            - For incoming datagrams, the new port number becomes a destination port number
        - Hosts on the LAN are completely oblivious to the fact that they are utilizing a private IP address that cannot be used to communicate directly with outside hosts
            - The NAT performs the translation to make this happen
    - Not only does the NAT modify the IP layer's header, it also modifies the transport layer's header
        - It maps a tuple of source IP and port number to the IP address that is allocated to your whole private network and a new port number
            - All mappings are remembered in the NAT's table
    - To summarize, the NAT in the router must:
        - Outgoing datagrams: Replace (source IP address, port number) of every outgoing datagram to (NAT IP address, new port number)
            - Remote clients/servers will respond using (NAT IP address, new port number) as destination address
        - Remember (in NAT translation table) every (source IP address, port number) to (NAT IP address, new port number) translation pair
        - Incoming datagrams: Replace (NAT IP address, new port number) in destination fields of every incoming datagram with corresponding (source IP address, port number) stored in NAT table
    - The NAT has some limitations
        - The new port number that is used to replace the original port number is limited
            - i.e. If you only have a single public IP address, and the total number of ports is 2^16, then the NAT can (only) support at most 2^16 concurrent connections
                - Note: 2^16 = 65536
- NAT: Network Address Translation (4)
    - i.e. Diagram of NAT In Action





1. Assume that an application on host '10.0.0.1', in the local area network (LAN) above, operating on port #3345 connects to an external HTTP web server with a destination IP address of 128.119.40.186 operating on port #80
    - Source IP is: 10.0.0.1
    - Source port is: 3345
    - Destination IP is: 128.119.40.186
    - Destination port number is: 80
2. When the datagram arrives to the NAT module on the gateway, it changes the source address from 10.0.0.1 to 138.76.29.7, and the port number from 3345 to 5001
    - Then, it stores the IP address and port number mappings in the NAT table
        - The private/source IP address is: 10.0.0.1
        - The port number on the application side is: 3345
        - The public IP address is: 138.76.29.7
            - Note: The public IP address is also referred to as WAN side address
                - WAN = Wide Area Network
        - The new (unique) port number that is assigned by the NAT is: 5001
    - Note: Only the source IP address and port number are changed. The destination IP address and port number are not changed, because these numbers correspond to the (HTTP) web server
3. Once the web server receives the datagram it responds to it in a normal manner; as if it is talking to a normal host
    - In the response message, it will utilize the same IP address and port number as the datagram it received from the host
        - Source IP address is: 128.119.40.186
        - Source port is: 80
        - Destination IP address is: 138.76.29.7
        - Destination port number is: 5001

        - From the web server's point-of-view, it has no knowledge that the application data is sent from a host inside a private network
            - To the web server, the connection originates from a normal, valid host
    4. When the datagram, sent from the web server, arrives at the gateway, the network address translation module inspects the packet's headers and compares it against the entries in its table
        - Upon finding a mapping between the WAN side address in the packet's headers, and a LAN side address in its table, it replaces the WAN side address with the LAN side address
            - It does this for the corresponding port number as well
        - The destination IP address is changed from '138.76.29.7' to '10.0.0.1'
        - The port number is changed from '5001' to '3345'
    5. The packet is delivered to the host on the LAN
- NAT: Network Address Translation (5)
    - To some extent, (the) NAT has addressed the IPv4 address exhaustion problem by allowing many-many hosts to utilize their own private IP address and share only a limited number of public IP addresses
    - NAT's main limitation is its 16-bit port number field, for every unique public IP address
        - The total number of concurrent connections is limited to 2^16, which is 65536
        - With a single LAN-side address, only `> 60,000` simultaneous connections are possible
    - NAT is controversial because it does not strictly follow the layered architecture of the Internet
        - NAT needs to utilize information in both the network layer and transport layer
            - Routers should only process up to layer 3
        - It violates the end-to-end argument because it sits in the middle, and performs translations for the end host
            - NAT possibility must be taken into account by app designers
                - i.e. NAT needs to be taken into account for P2P Applications
                - In P2P, since the network host exposes itself as a client and server, you need to find a way to let other people know what is the port number you are operating on - not the one on your localhost, but the one that is being mapped and stored by the NAT
        - The success of NAT has made adoption of IPv6 much harder
            - This is not a good side-effect, because NAT does not address the fundamental problem of IPv4 address

        exhaustion
          - NAT is a 'hack' that makes things work
      - IP address shortage should ideally be solved by IPv6
- Outline
  - Introduction
  - What's inside a router
  - IP: Internet Protocol
    - Datagram format
    - IPv4 addressing
    - IPv6
    - ICMP
      - This is the next topic
  - Routing Algorithms
- ICMP: Internet Control Message Protocol
  - Used by hosts & routers to communicate network-level information
    - There are 2 types of information that can utilize ICMP:
      1. Error reporting
         - Hosts can use error reporting to indicate whether a certain port is not available
         - The network can use error reporting to indicate information such as unreachable host, network, port, protocol, etc.
      2. Status check
         - ICMP can be used to convey information about the status
           - i.e. echo request/reply
             - Used by ping to check whether a host, or router, is online or not
  - ICMP has its limitations
    - i.e. Routers/hosts can be configured to not respond/send ICMP messages
      - The limitations depend on the implementation
  - Note: `traceroute` is implemented using ICMP
  - Even though ICMP is a network layer protocol, it operates on top of IP
    - This means that all ICMP msgs are carried in IP datagrams, in the protocol field
      - The protocol field corresponds with which of the upper layer services that utilize IP
      - Since ICMP operates on top of IP, in the protocol field, you can specify '1'
        - An IP datagram with a protocol ID of '1', then the router knows that the message needs to be delivered to the ICMP processing
  - The ICMP message stores information such as:
    - Type of errors, status information, etc.
      - It utilizes a combination of 'type' and 'code', and it includes the first 8 bytes of the IP datagram that caused the error

        - If there is an error that generates an ICMP message, the first 8 bytes will provide some information of what is going on
- i.e. Table of ICMP Message Types

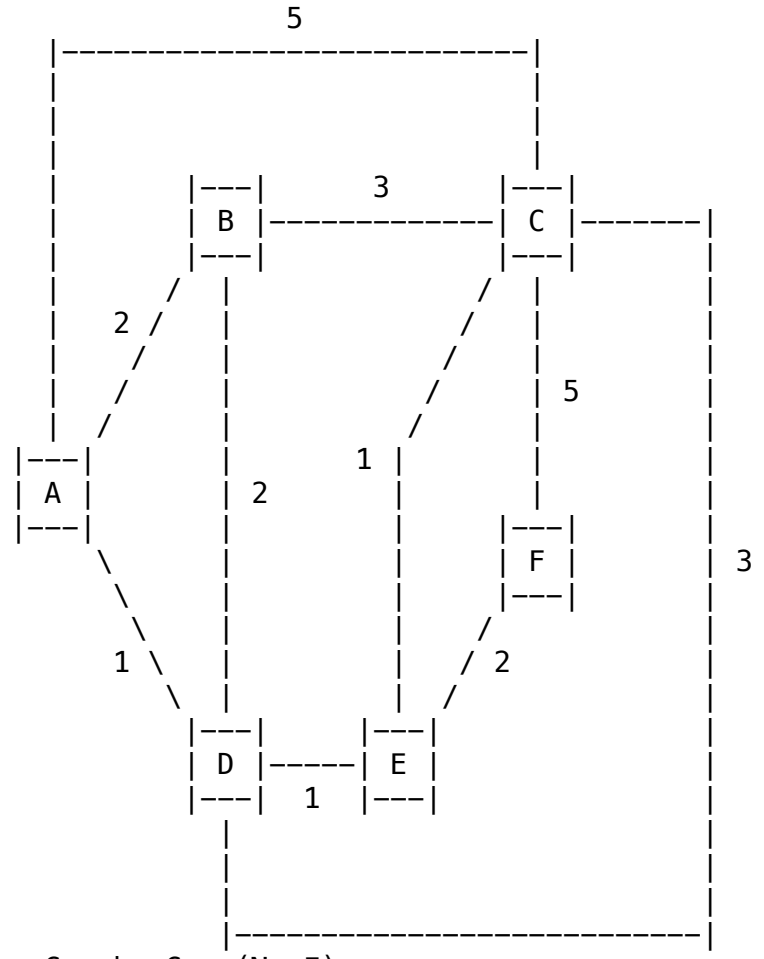
Type	Code	Description
0	0	echo reply (ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

    - This table summarizes the different combinations of type and code that corresponds to different types of reply or response messages, or different types of errors
    - Type 0 and Code 0 correspond to echo reply, and Type 8 and Code 0 correspond to echo request
        - If you ping a host, your machine will send it an ICMP message with Type 8 & Code 0, and the end-host will reply with Type 0 & Code 0, assuming that the end-host has enabled ICMP messages
- Question: What ICMP messages do you use for `traceroute`?
    - Answer: Type 11 & Code 0
        - This corresponds to 'TTL expired', and is generated by the host
    - In `traceroute`, UDP/TCP segments are sent to particular routers along the path to the destination host. We want to restrict how far the segment can traverse by

utilizing the 'TTL' field in the IP header. We also need to know if the segment has eventually reached the destination host by incrementing the 'TTL' field by 1 for each router/hop it visits. Eventually, we should be able to know if the segment has reached the destination host, so we can terminate the session.
        - Each router that the segment visits will decrement the 'TTL' value by 1. If the 'TTL' value reaches zero, then the ICMP message 'TTL expired' is generated and sent back to the source-host
- Traceroute & ICMP
    - `traceroute` works by sending a series of UDP segments to the destination IP address, on a port number that is unlikely to be used
        - By utilizing a port number that is unlikely to be used, and placing it in the UDP datagram, `traceroute` can take advantage of the ICMP message that corresponds to Type 3, Code 3
            - The Type 3, Code 3 protocol means that the destination port is unreachable
        - When the UDP datagrams reach the end-host, an ICMP error message will be generated because the port specified in the UDP datagram does not exist on the end-host
            - The ICMP message will be sent to the source-host that initiated the `traceroute`
    - There are 2 ICMP messages that are utilized in `traceroute`:
        1. Type 11, Code 0
            - Used for the purpose of determining if the router of certain hop counts has been reached
                - Combined with the 'TTL' field in the IP header
        2. Type 3, Code 3
            - By putting an unutilized port number in the UDP/TCP segment, the destination host will generate an ICMP error message, and send it to the source-host
                - The corresponding error message is: "destination port unreachable"
        - Using these 2 ICMP messages, you can create your own implementation of `traceroute`
    - To summarize:
        - Source sends series of UDP segments to dest.
            - First has 'TTL' = 1
            - Second has 'TTL' = 2, etc.
            - Unlikely port number
        - When the `n'th` datagram arrives to the `n'th` router:
            - Router discards datagram
            - And sends to source an ICMP message
                - Type 11, Code 0
            - Message includes name of router & IP address
        - When ICMP message arrives, source calculates 'RTT'
            - `traceroute` does this 3 times

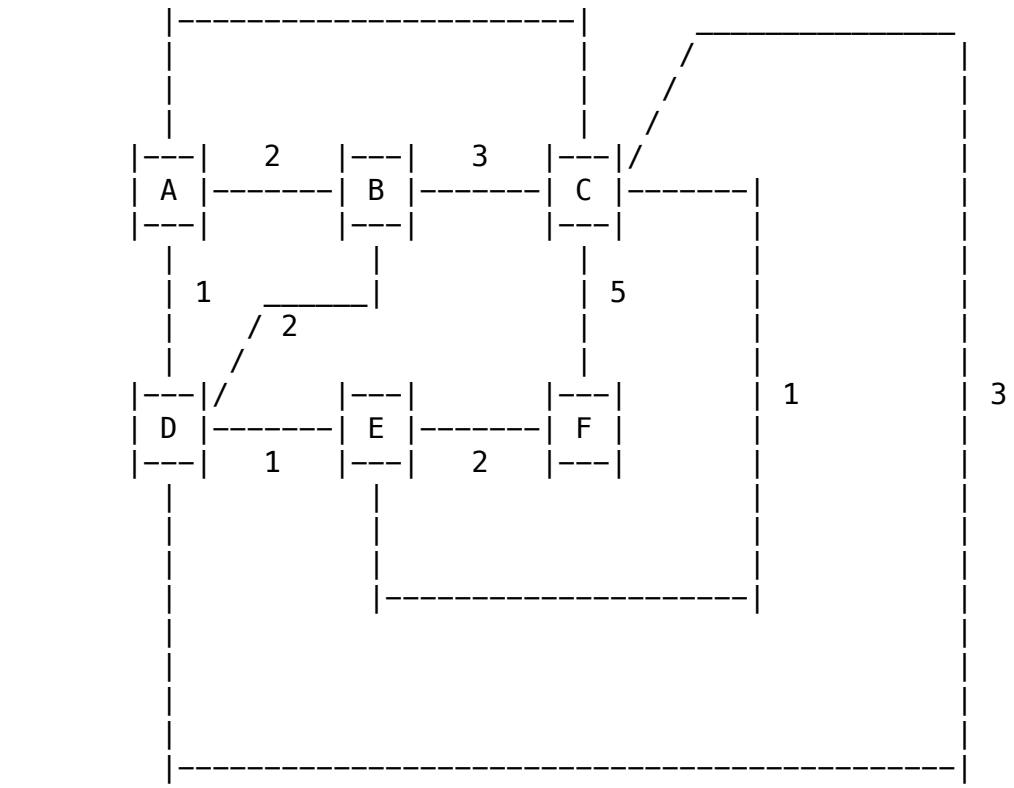
        - Stopping criterion
            - UDP segment eventually arrives at destination host
            - Destination returns ICMP "dest port unreachable" packet
                - Type 3, Code 3
        - When source gets this ICMP, it stops
- Routing Protocols
    - Answers the question: "How exactly do routers figure out the path from an arbitrary source to an arbitrary destination in the internet?"
        - Routing protocols are the core of the network layer
    - There is no single entity in the Internet that calculates routes and disseminates it to different routers
    - Routes are calculated by distributed protocols that are implemented by every single router
        - These routing protocols exchange information that will help each other to be able to figure out some kind of efficient route
    - When discussing routing protocols, the network is abstracted as a graph
        - In a graph, we typically have:
            - Vertices
            - Edges
                - In a network graph, each edge has a number to indicate the cost
                    - Typically, the bigger the number, the higher the cost
    - The point of network routing protocols is to figure out a good path, from source to destination
        - "A good path" implies a path with a lower cost
            - The least cost path is the shortest path
            - Less hop count passes is desirable
            - If the edges have different (positive) weights associated to them, then we are looking for the least cost path from source to destination
    - Questions to consider:
        - How to figure out the least cost path on a graph?
        - How to implement algorithms utilizing routing protocols?
            - How can they exchange information that's necessary to calculate least cost paths?
        - How is information maintained in a distributed network?
        - How to deal with changes inside the network?
            - i.e. Router failure, the cost associated with an edge may change, etc.
    - To summarize:
        - Routing protocols are implemented on the routers of a network
            - Establish paths between nodes
        - Network modeled as a graph
            - Routers are graph vertices

        - Links are edges
        - Edges have an associated "cost"
            - i.e. Distance, loss, latency, etc.
    - Goal: Compute a "good" path from source to destination
        - "good" usually means the shortest, least cost, path
        - Question: How to compute a "good" path?
            - Answer: Centralized VS. Distributed algorithms
- Graph Abstraction
    - i.e. Graphical Representation of a Network



    - Graph: G = (N, E)
        - N = Set of routers = {A, B, C, D, E, F}
        - E = Set of links = {(A, B), (A, D), (B, D), (B, C),
                              (D, C), (D, E), (C, E), (C, F),
                              (E, F)}
        - Cost of path(x1, x2, x3, ..., x_p)
            = c(x1, x2) + c(x2, x3) + .. + c(x_p-1, x_p)
                - i.e. Cost of path (B,A,D) = c(B,A) + c(A,D)
                                            = 2 + 1
                                            = 3
    - Each vertex corresponds to a router
    - The address corresponds to links at connecting

neighbouring routers
  - The cost of a path is the sum of the cost of the edges in the path
    - i.e. Cost of path (D,E,F) = c(D,E) + C(E,F)
                                = 1 + 2
                                = 3
  - In the example above, there are:
    - 6 routers, in total
    - 10 edges, that connect the routers
      - Each edge has its own cost
        - i.e. The cost of (C,E) is: 1
- i.e. Alternate Graphical Representation of a Network



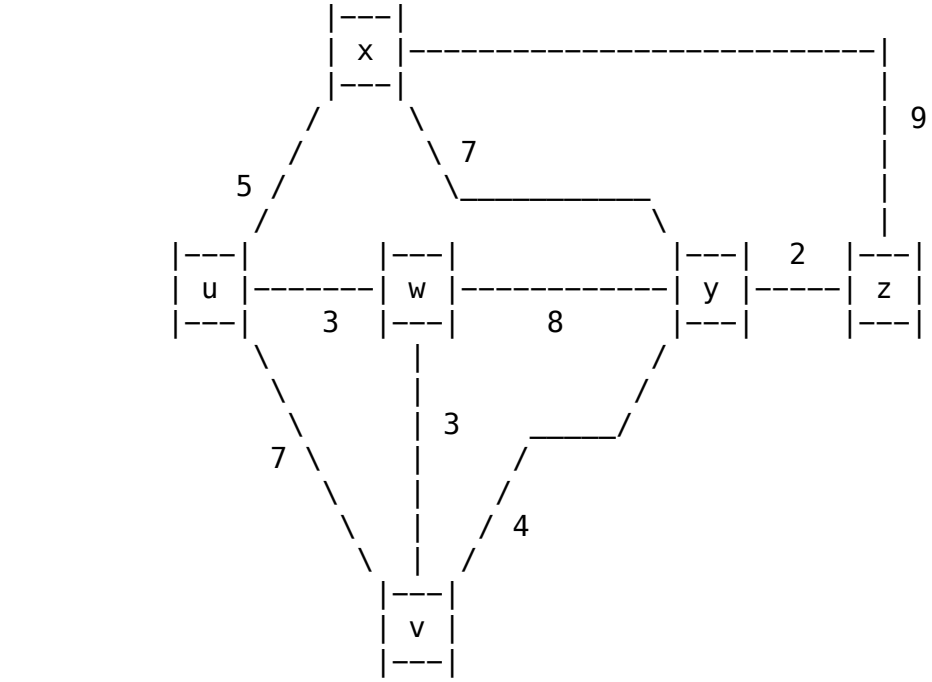
- A Link State Routing Algorithm
  - Dijkstra's algorithm falls into the category of link state algorithm
    - In order to execute Dijkstra's algorithm, every single router needs to have the knowledge of the complete topology
      - i.e. How many other routers are present, and what are the link costs connecting those routers?
    - Based on topology information, every router will be able to calculate the least cost path from a source to all other nodes
      - In Dijkstra's algorithm, this is done in an iterative manner
        - After `n` iterations, you will be able to know

                the least cost path to a destination that is `n` hops away
    - To implement Dijkstra's algorithm:
        - Net topology, link costs known to all nodes
            - Accomplished via "link state broadcast"
            - All nodes have same info
        - Compute least cost paths from one node ('source') to all other nodes
            - Gives forwarding table for that node
        - Iterative: After `k` iterations, know least cost path to `k` destinations
    - The following notation is used in Dijkstra's algorithm:
        - c(x,y): Link cost from node `x` to `y`
            - If `x` and `y` are direct neighbours, then the link cost is a finite number, equal to the cost of the associated edge
            - If `x` and `y` are not direct neighbours, then the link cost is set to 'infinity'
        - D(v): Current value of cost of path from source to destination, `v`
            - It is assumed that the source is fixed
            - The least cost path from the source to all other nodes in the network
            - By the end of the procedure we should have a tree with the source as the root node
        - p(v): Predecessor node along path from source to `v`
        - N': Set of nodes whose least cost path definitively known
            - In other words, these nodes have already been computed
- Dijkstra's Algorithm
    - i.e. Pseudocode of Dijkstra's Algorithm

```
Initializaion:
    N' = {u}
    for all nodes v:
        if v adjacent to u:
            then D(v) = c(u,v)
        else: D(v) = 'infinity'
Loop
    find w not in N' such that D(w) is a minimum
    add w to N'
    update D(v) for all v adjacent to w and not in N':
        D(v) = min(D(v), (D(w) + c(w,v)))
    /* new cost to v is either old cost to v or known
       shortest path cost to w plus cost from w to v */
Until all nodes in N'
```

    - Note: (N') = 'N prime'
    - On every iteration, the loop adds one node into the (N') set
        - The (N') set contains nodes where the shortest path from the source node to that node is known

        - The total number of times that the loop needs to run is at most `N`, which corresponds to the total number of vertices in the graph
    - Within each iteration, `D(v)` needs to be updated for all vertices adjacent to the newly added vertex
        - This is done using the following equation: D(v) = min[D(v), (D(w) + c(w,v))]
        - Only nodes that are currently not in `N'` are considered
    - The key idea is that:
        - For the nodes in the set (N'), their shortest path from the source is already determined, and for other nodes that are not in the set, their shortest path will come from routing through the nodes in (N'), because their shortest path has already been computed
            - Therefore, you want to test whether the current distance from the source to a node, `v`, is bigger or less than if you route through a node in (N'), whose shortest path has already been computed
        - At the end of the iteration, the vertex outside (N') that has the least cost is selected and included in (N')
            - By doing this, the number of undecided nodes is decremented by one, and the number of decided nodes is incremented by one
                - Thus, after at most `N` iterations, all the nodes will have their shortest path from the source node determined, and Dijkstra's algorithm will terminate
- Dijkstra's Algorithm: Example
    - i.e. Graph of a Network

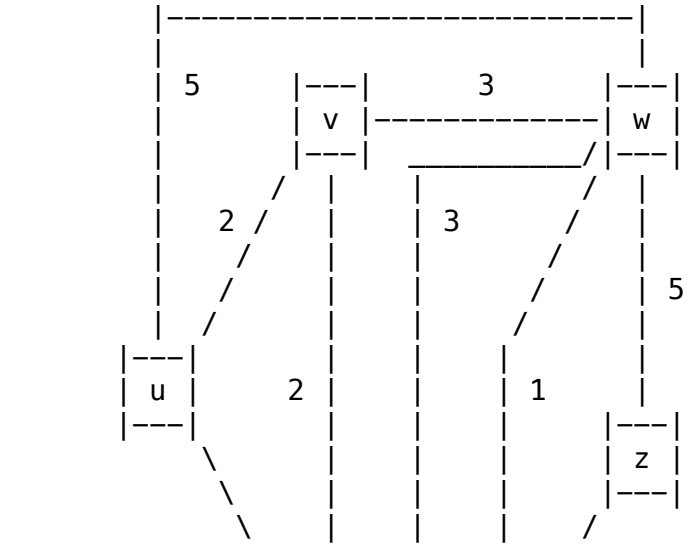


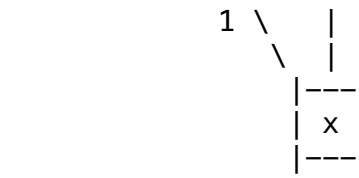
    - i.e. Table of Shortest Path

		D(v)	D(w)	D(x)	D(y)	D(z)
Step	(N')	p(v)	p(w)	p(x)	p(y)	p(z)
0	u	7,u	3,u	5,u	**	**
1	u,w	6,w		5,u	11,w	**
2	u,w,x	6,w			11,w	14,x
3	u,w,x,v				10,v	14,x
4	u,w,x,v,y					12,y
5	u,w,x,v,y,z					

    - Note: '**' represents infinity
- First off, (N') is initialized to the source node, `u`
- The goal is to figure out the least cost path from `u` to all the other nodes in the network
- The shortest path from `u` to `u` is 0, because it is both the source and destination
- Before the Dijkstra's algorithm is executed, you need to populate the distance from `u` to other nodes based on whether or not they are direct neighbours of `u`
  - i.e. In the graph above, `x`, `w`, and `v` are direct neighbours of `u`
    - The other nodes are not populated because it is unknown if there is a path to them from `u`
      - Thus, the distance from `u` to these nodes is infinity
- At each the beginning of each iteration of Dijkstra's algorithm, the node/vertex with the least cost is selected
  - The selected node is placed into (N')
  - In the example above, node `w` is selected for the FIRST iteration of the algorithm
    - Also, the shortest path from `u` to `w` is through their direct link
  - Node 'x' is selected for the SECOND iteration of Dijkstra's algorithm
- In order the update the path cost, the algorithm takes the minimum of the current (path cost) value, and the cost of the path that it would get if it routed through the newly added vertex
  - i.e. The node `v` has a direct link coming from `u` with a path cost of 7. There is also an additional node in (N'), `w`, that has its least cost path figured out. So, from the point-of-view of `v`, there are two alternatives:

            1. The path taken can be directly from `u` to `v`
               OR
            2. The path taken can be from `u` to `w`, and then
               `w` to `v`
        - The path from `u` to `v` is 7, and the path from `u`
          to `v` via ``w` is 6. Thus, it is lower, and
          shortest path table is updated with this new value
    - This procedure is completed for other vertices, except
      for vertices that are not direct neighbours of the newly
      added vertex; because it is unknown if a path exists
        - Based on this, the table is updated accordingly
- After the iteration completes, if you trace back the
  predecessors, then you can figure out what is the least cost
  path
    - i.e. In order to determine the least cost path from `u`
      to `z`, you look at the predecessor, last value,
      under the `D(z)` column. The predecessor is `y`.
      Then, you lookup the predecessor for `y`, which is
      `v`, and the predecessor of `v` is `w`. You repeat
      the process of looking up the predecessor until you
      reach the source node, `u`
    - By tracing out the predecessor nodes, this process can
      be used to determine the shortest path tree
        - Also called "minimum spanning tree (MST)"
        - The source node is the root node, and other nodes
          are leaf/intermediate nodes
        - i.e. For the graph above, the shortest path tree is:
            `x` -> `u` -> `w` -> `v` -> `y` -> `z`
- Notes:
    - Construct shortest path tree by tracing predecessor
      nodes
    - Ties can exist (can be broken arbitrarily)
- Dijkstra's Algorithm: Another Example
    - i.e. Graph of a Network





- i.e. Table of Shortest Path

		D(v)	D(w)	D(x)	D(y)	D(z)
Step	(N')	p(v)	p(w)	p(x)	p(y)	p(z)
0	u	2,u	5,u	1,u	**	**
1	u,x	2,u	4,u		2,x	**
2	u,x,y	2,u	3,y			4,y
3	u,x,y,v		3,y			4,y
4	u,x,y,v,w					4,y
5	u,x,y,v,w,z					

  - Note: '**' represents infinity
- Initially:
  - (N') = `u`
    - `u` is the source node, and Dijkstra's algorithm will calculate the least cost path from `u` to other nodes
  - `D(v), p(v)` = 2
    - `v` is a direct neighbour to `u`, and the link cost is 2
    - The predecessor is `u`
  - `D(w), p(w)` = 5
    - 'w' is a direct neighbour to `u`, and the link cost is 5
    - The predecessor is `u`
  - `D(x), p(x)` = 1
    - `x` is a direct neighbour to `u`, and the link cost is 1
    - The predecessor is `u`
  - `D(y)` and `D(z)` cannot (currently) be calculated, because they are not direct neighbours to `u`
    - Their value will start off as 'infinity'
- After the initial step, Dijkstra's algorithm iterates over the elements in (N'), and adds the least cost node to (N')
  - For the first iteration, since the only element in (N') is `u`, the algorithm adds `x` to (N') because the least cost link attached to `u` is `x`
    - Now, (N') = [`u`, `x`]

- Then, the least cost paths for the newly added node, to its direct neighbour nodes, are computed
  - i.e. D(w) = min(D(w), (D(x) + c(x,w)))
    = min(5, (1 + 3))
    = min(5, 4)
    = 4
    - The path from `u` to `x` to `w` is shorter than the path from `u` to `w`
      - The new least cost path value for `D(w)` is 4, instead of 5
      - The predecessor is updated to `x` from `u`
- This process is computed for all nodes in (N'), and the least cost link attached to the node is added to (N')
  - Note: We do not determine the order of (N'). After we update the distance value of all vertices corresponding to a node, we select the direct-neighbour node with the lowest link. If there are two (or more) least cost links, then the selection is arbitrarily made