Justin Ciocoi

Oct. 10, 2023

CSCI 379 Textbook Notes

Computer Networking

4.3: The Internet Protocol (IP): IPv4, Addressing, IPv6, and More

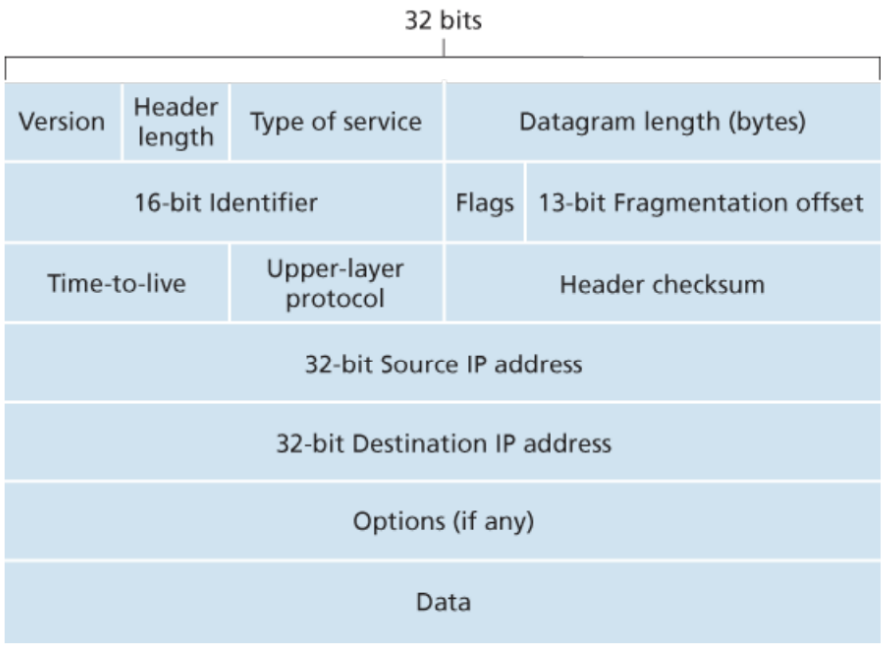
- **4.3.1: IPv4 Datagram Format**
  - 

Figure 4.16 IPv4 datagram format

  - As shown in the above, the key fields in an IPv4 datagram are
    - **Version number** (4-bit) which specifies the IP protocol version and tells the router how to interpret the remainder of the datagram
    - **Header Length** (4-bit) tells the router how long the header is and therefore where the payload actually begins
    - **Type of Service (TOS)** allows different types of IP datagrams to be distinguished from each other
    - **Datagram Length** (16-bit)

        - **Identifier, flags, fragmentation offset**
            - These only exist in IPv4 and not in IPv6
        - **Time-to-live** which is a field used to ensure that datagrams do not continue to circulate forever
        - **Protocol** specifies the transport-layer protocol to which the data from the datagram should be passed
        - **Header checksum** which helps in detecting bit errors during transmission
        - **Source and Destination IP Address** which specify the source and destination IPs of the datagram
        - **Options** which allow the IP header o be extended and used rarely to save the overhead used in the average IP datagram
        - **Payload** which is the actual data contained within the datagram
    - Some protocols can carry big datagrams whereas others only smaller ones
    - The maximum amount of data that a link-layer frame can carry is called the *maximum transmission unit (MTU)*
- **4.3.2: IPv4 Datagram Fragmentation**
    - If a router receives a datagram that has a size larger than its corresponding output link's MTU, it must fragment the payload into two or more smaller datagrams
        - Each smaller datagram is referred to as a *fragment*
    - When fragmented datagrams arrive, they must be reassembled before they reach the transport layer at the destination
    - This is where the identifier, flags, and fragmentation fields of the IPv4 datagram are used
    -

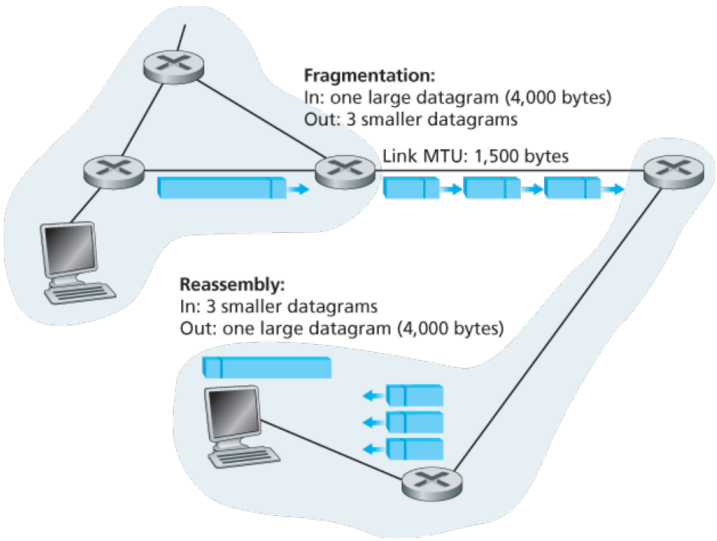


Figure 4.17 IP fragmentation and reassembly

- **4.3.3: IPv4 Addressing**
  - The boundary between a host and its physical link to the network is called an *interface*
  - IP addresses, rather than correlating to hosts directly, correlate more directly to the interfaces within those hosts
  - In IPv4, IP addresses are *32 bits, or 4 bytes*, long
  - A network which connects host interfaces to a router interfaces is known as a *subnet (also called IP network, or just network)*
  - A *subnet mask* is a prefix of bits that is the same across all devices connected to the same subnet
    - The length of a subnet mask. can vary
  - *To determine the subnets, detach each interface from its host or router, creating islands of isolated networks. Each network here is a* ***subnet***
  - The Internet's address assignment strategy is known as *classless inter-domain routing (CIDR)*, pronounced "cider"
  - CIDR generalizes 32 bit IP addresses in the form $a.b.c.d/x$, where x indicates the number of bits in the first part of the address
  - x is generally referred t as the *network prefix*
  -

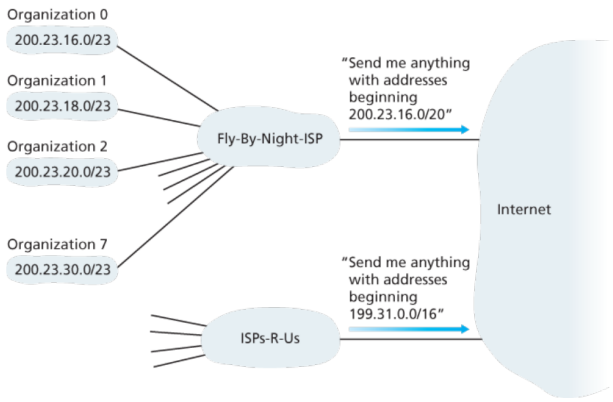


Figure 4.21 Hierarchical addressing and route aggregation

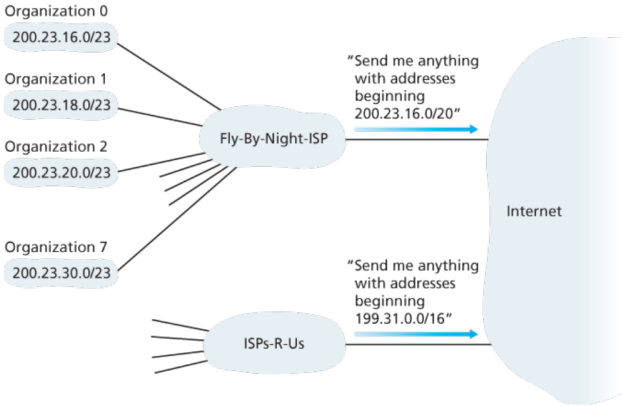


Figure 4.21 Hierarchical addressing and route aggregation

  - The above illustrates what would happen why a hierarchical structure for IPv4 addressing as it allows more dynamic reallocation of IP subnets

- **Dynamic Host Configuration Protocol**
  - After a block of IP addresses is obtained by an organization from an ISP, the host addresses must also be configured
  - While this *can* be done manually, it is most often done through the use of *DHCP*
  - DHCP can be configured to give a temporary IP address each time a network connection is initiated or to give the same IP address to each device
  - DHCP automated much of the process of connecting a host system to a network, and is thus often referred to as a *plug-and-play* or *zero-conf (zero configuration)* protocol
  - **DHCP Server Discovery**
    - When a new device connects to the network, it must first find a DHCP server using a *DHCP discover message*, which a client sends within a UDP packet to port 67
    - The message is sent with source IP 0.0.0.0 and destination IP 255.255.255.255 and is broadcast to all nodes on the subnet

- **DHCP Server Offers**
  - A DHCP server will respond to the client with a *DHCP offer message* that is also broadcast to all nodes on the subnet
  - Since their may be multiple DHCP servers on the network, a client may be able to choose between multiple offers
  - The DHCP server will assign the IP address, the network mask, and an *IP address lease time*, the amount of time for which an IP address will remain valid
- **DHCP Request**
  - After choosing an offer, the client will send a *DHCP request message*
- **DHCP ACK**
  - After receiving a DHCP request message, the DHP server will respond with a *DHCP ACK message* which confirms the parameters requested by the client

- **4.3.4: Network Address Translation**
  - Given everything above, we know that every IP-capable device needs an IP address
  - However let's consider some issues with the systems described earlier in this section
    - Given an already allocated block of addresses, what happens when an organization requires more addresses than are allocated within the block
      - This is especially concerning as more and more internet connected devices are in use in almost every workplace
    - Especially if two contiguous address blocks are already assigned, then an organization might be forced to have non-sequential blocks of IP addresses - something that would complicate network management
  - The solution to this problem is *network address translation (NAT)*
  -

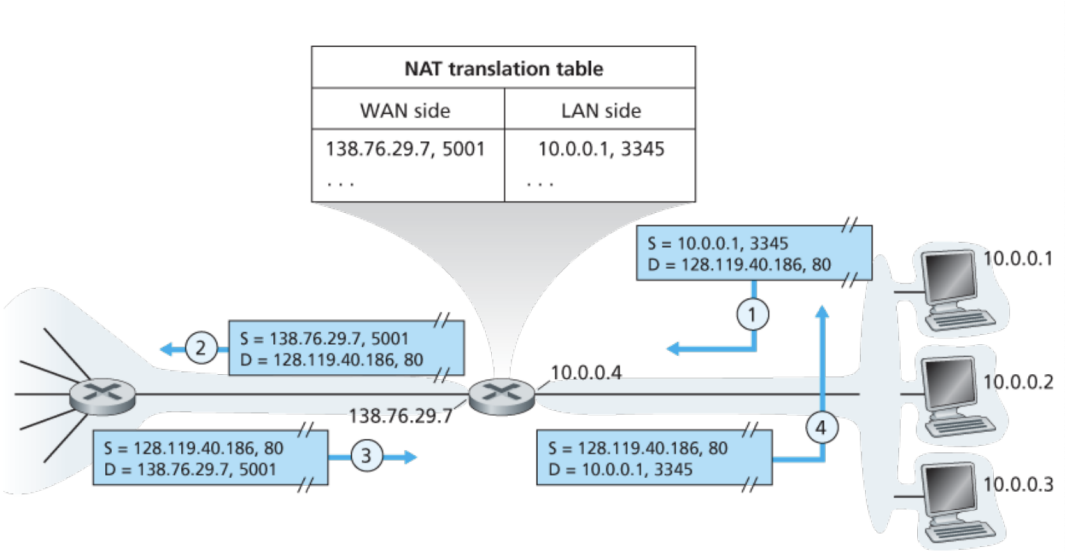


Figure 4.25 Network address translation

- The NAT-enabled router clearly functions as a router, but to the outside world, it *looks* like it is a *single* device with a *single* IP address
- In the above figure, all traffic leaving the home router has a *source address* of 138.76.29.7 and all traffic entering the home router has a *destination address* 138.76.29.7
- If all arriving traffic has an identical destination IP address, how does the router know which internal host should be receiving the data in question
    - A *NAT translation table* will be used at the NAT router and port numbers will be included alongside IPs in this table
- A NAT translation table indexes source ports assigned to outgoing traffic to the port numbers used on the WAN, and the table is then used to translate the public IP and port to host IP
- *Focus on Security*
    - Knowing a range of IP addresses belonging to an organization could enable an attacker to conduct various types of malicious packet attacks
    - Two popular defense mechanisms to these types of attacks are
        - *Firewalls*
        - *Intrusion Detection Systems (IDSs)*
    - For example, a firewall could be in place to prevent all ICMP echo request packets from coming through, thereby preventing port scanning or network mapping

attacks

    - An IDS uses an already known list of packet signatures that are part of attacks, and if a match between this database and entering packets is found, an alert is created
    - An *Intrusion Prevention System (IPS)* is similar but actually blocks the traffic rather than just creating an alert

- **4.3.5: IPv6**
    - In the early 1990s, researchers started to realize that the 32-bit IPv4 address space was beginning to run out of available addresses with new nodes and subnets being added at an exponentially growing rate
    - To respond to this, a new IP protocol, IPv6, was created
    - In addition to increasing the size of the address space, designers of IPv6 took the chance to tweak and augment various other aspects of the IPv4 protocol based on the operational experience they now had with it
    - In February 2011, IANA allocated out the last available IPv4 addresses within their pool
- **IPv6 Datagram Format**
    -

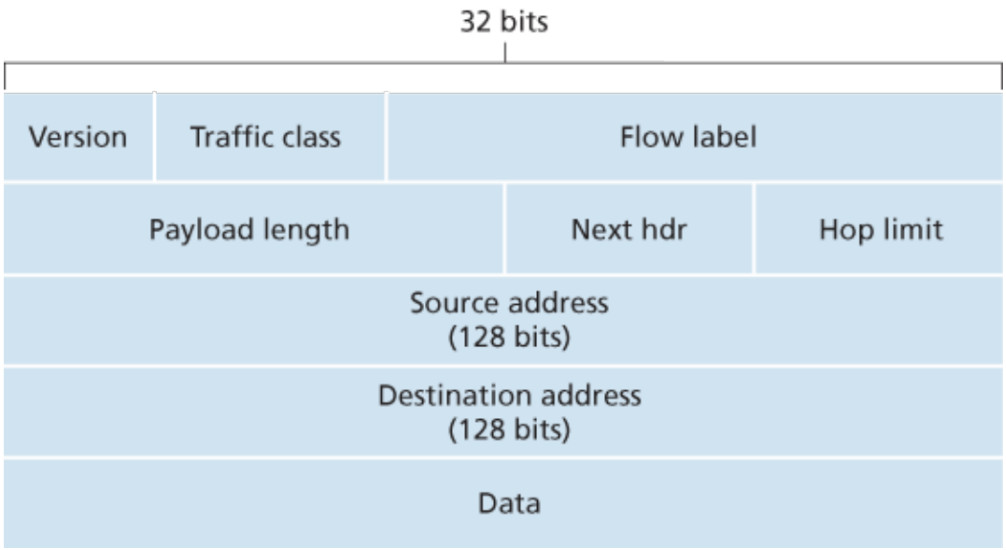


Figure 4.26 IPv6 datagram format

    - The key differences between IPv4 and IPv6 are:
        - **Expanded Addressing Capabilities**

        - IPv6 increases the size of the IP address from 32 bits to 128 bits (with IPv6 every grain of sand on earth could be assigned an IP address)
        - IPv6 introduces *any-cast addresses* which allow a datagram to be delivered to any one of a group of specified hosts
            - This could be useful, for example, for downloading a document from the closest, and therefore fastest, server hosting that document
    - **Streamlined 40-byte Header**
        - A number of IPv4 fields have been dropped or made optional, which results in faster processing of IP datagrams by a router
    - **Flow Labeling**
        - IPv6 can differentiate between "types" of *flow*
        - For example, a real time service would be treated as a *flow* since a continuous transfer of information would bee necessary to maintain the experience
- As shown in the above diagram, the key fields of the IPv6 datagram are
    - **Version**, which is a 4-bit field identifying the IP version number
    - **Traffic Class**
        - Can be used to prioritize certain datagrams *within* a flow
        - Can be used to prioritize datagrams from certain applications over other applications
    - **Flow Label**, a 20-bit field used to identify a flow of datagrams
    - **Payload Length**, which is a 16-bit value representing the length of the IPv6 datagram not including the 40-byte header
    - **Next Header**, which identifies the protocol to which the data will be delivered (i.e. UDP or TCP)
    - **Hop Limit**, The upper limit on the number of times a datagram can be transmitted by a router; a datagram is discarded when the limit is surpassed
    - **Source and Destination IP Addresses**

        - **Data**
    - As we can see here, there are several fields present in the IPv4 datagram that are missing here
        - **Fragmentation/Reassembly**
            - Under IPv6, fragmentation and reassembly can be done *only* at the source or the destination, and not by intermediate routers
            - If a datagram is too large, a router will simply drop it and send back a "Packet Too Big" ICMP error message to the sender, after which the sender should respond using a smaller datagram size
            - By removing fragmentation and reassembly functionality from the routers, considerably speeds up IP forwarding within the network
        - **Header Checksum**
            - Since protocols like TCP, UDP, and Ethernet already perform checksumming, the designers of IPv6 probably felt it was largely redundant to include it again in the IP protocol
        - **Options**
            - While not present in the header, the *options* field still can be one of the possible headers pointed to from within the original IPv6 header
            - Since there is no options field, we can have a fixed length IPv6 header
- **Transitioning from IPv4 to IPv6**
    - How will the widespread public internet, which is based on IPv4, be transitioned to IPv6?
    - One approach would be to declare a flag day - a given time and date where all internet machines would be turned off and upgraded from IPv4 to IPv6
        - Given the size of the modern internet, such an approach is unthinkable
    - The approach that has been most widely adopted in practice is one that involves *tunneling*

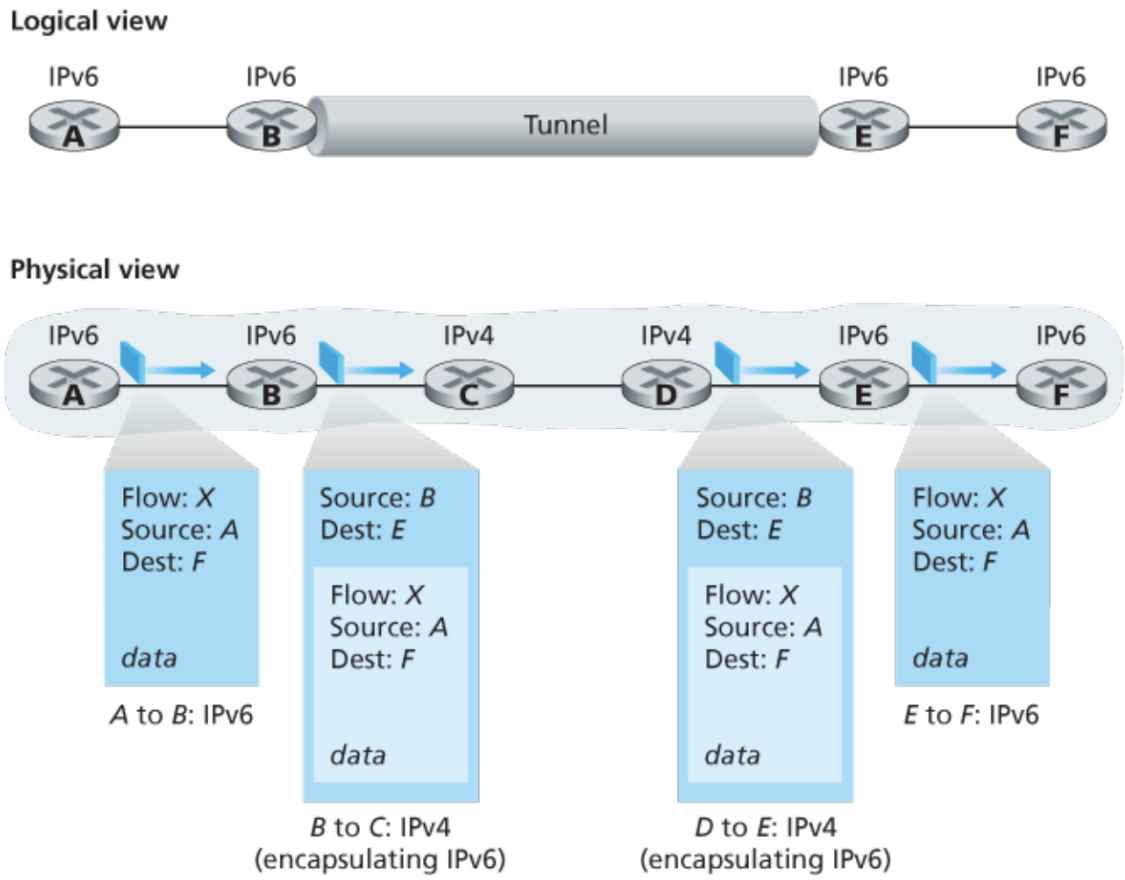


Figure 4.27 Tunneling

- Assume for example two IPv6 hosts are connected by a series of IPv4 routers
- If host A wants to send an IPv6 datagram to host B, it can encapsulate the *entirety* of the IPv6 datagram, including header, in an IPv4 datagram and send it that way
- This way, the IPv4 routers can act as a *tunnel* for IPv6 data while not understanding it, as only the receiving host must be able to decipher the data contained inside

- One important lesson to take from this is the difficulty that comes with changing protocols at any level, and the desire to design and create protocols that are more scalable and adaptable to deal with the future