04. The Network Layer: Data Plane

4.1 Overview of Network Layer

The network is the layer responsible for sending the data to its target destination. It does this through two methods; forwarding and routing.

- Forwarding: upon receiving a packet, the router determines which link to forward it to
- Routing: the end-to-end process that determines all the nodes in the network a packet should be forwarded through

One should also be aware of the difference between software and hardware level routing.

The IP protocol is a best-effort protocol which means it has no guarantees for packets reaching the other end or order of packets.

4.2 What's Inside a Router?

A router formally consists of four parts; input ports, switching fabric, output ports, and a routing processor.

- Input ports: packets arrive and look into the lookup table to determine which output port to transfer to.
- Output ports: packets exit through the output ports
- Switching fabric: hardware which guides the packets from the input ports to the output ports
- Routing processor: typically hardware which decides what goes where, but could also be software. Hardware is a lot quicker here, and typically preferred.

Deciding which output to forward to can be decided in multiple ways, but is usually decided by the destination address in the IP datagram. This is usally destination based forwarding with longest prefix matching.

Switching in the fabric can be done in multiple ways including in-memory, bus, or crossbar.

- In-memory is simply copying the input packet into memory and delivering it to the right port. This is slow because only one read/write operation may happen in memory at any time, making it slow for multi-port routers.
- Bus: uses a bus system to transfer packets to all ports, but only the forwarding port keeps the packet. This causes a bus bottleneck because all packets travel through the bus.
- Crossbar, or interconnection network uses a set of buses between the inputs and outputs that may be opened or closed. This makes it possible for data to flow to multiple ports in parallell, but will still slow down if two packets from different input ports are going to the same output port.

When we talked about packet loss in the transport layer, we were talking about queue loss. Queue loss happens when there becomes a bottleneck in the switching fabric, and packets are lost because the queue becomes too big.

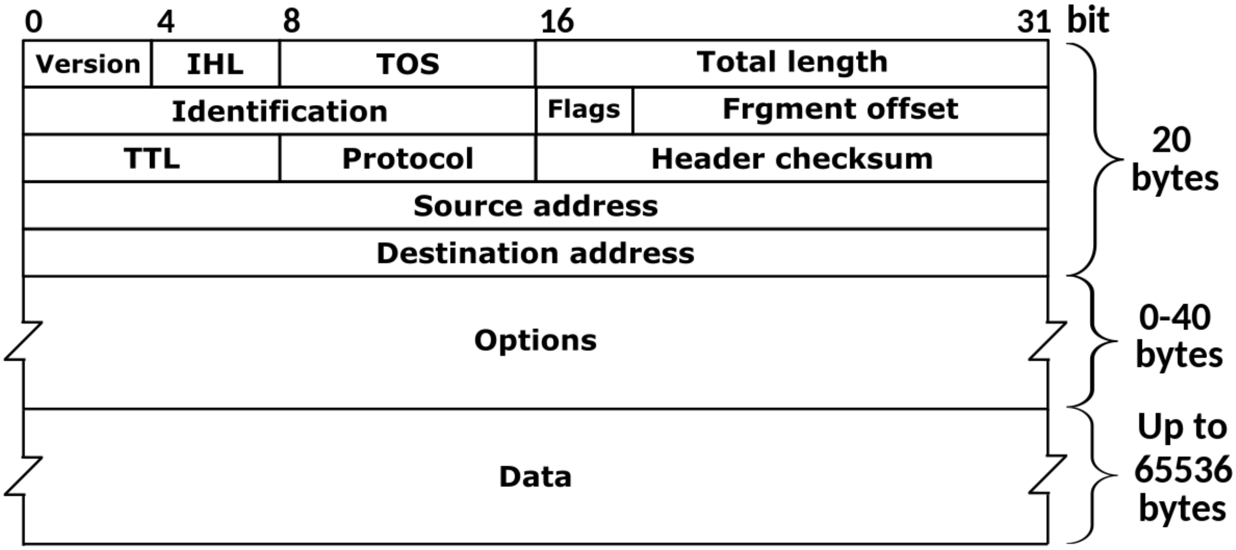
HOL-queuing is when two packets in the same input are going to different outputs, but the front packet’s output already has a queue. This means the second input packet is queued due to the first input packet.

Most routers can handle some queuing by keeping a packet buffer which acts as a queue. Anything outside of the buffer size will be lost. We can also prioritize different packets over other based on their type of class in the IP datagram header. The question is how to make this fair?

This is also where net-neutrality comes in, should one be able to pay for priority here? (No, ple

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

The IP protocol has two major versions in use; v4 and v6. IPv4 is old, and we are actively trying to move away from it. It has the following header datagram.



- Version: denotes which IP version is in use (because IPv6 headers are different).
- IHL: The length of the whole leader (because IPv4 headers are variable-length)
- TOS: Type of service that can allow above layers to distinguish diagrams (such as priority for realtime apps)
- Total Length: The length of the entire datagram
- Identification, flags, fragment offset: IP Fragmentation related - not in curriculumn
- TTL: Amount of router the diagram can pass through before having to be discarded. (Decreased at each router, 0 means router should discard).
- Protocol: Transport layer protocol number, see https://www.iana.org/assignments/protocol-numbers
- Header checksum: 1s-complement of every two bytes interpreted as an integer checksum
- Source address: IP address of origin
- Destination address: IP address of target destination
- Options: Possibly extend an IP diagram, rarely used
- Data: actual data in the diagram

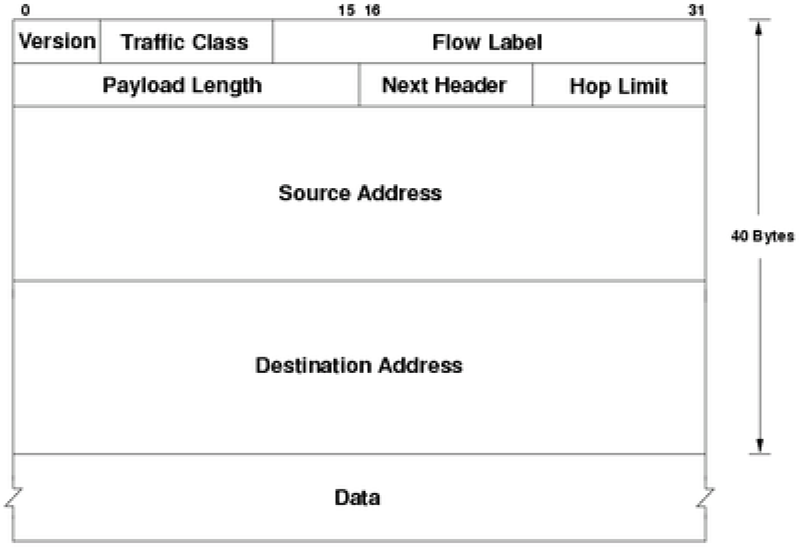
Addressing in IPv4 is a complex task. IP addresses are managed by ICANN (who also manage domain names and root DNS servers).

A subnet is a part of an IP address, and usually have reserved n bits of an address. The indicates 123.57.21.3/24 indicates that it's an address in the 123.57.21.x subnet. The number after the slash indicates how many bits from the left are part of the subnet. ISPs typically get a subnet dedicated to them, then they can split that among their customers.

On each network, you will typically get your address in the subnet automatically through the DHCP protocol. It's a protocol designed to hand out spaces in the subnet to devices. It's a server-client protocol. Your device contacts the DHCP server and receives an address. DHCP can use and store metrics about the device if you, for example, wish to give the same address to the same device.

On local networks you can set up a NAT on a subnet. This is a private network which means you can re-route addresses (that may be used outside the NAT) locally. This is one way to "get more" IPv4 addresses than there exist (because they can be hidden private behind a NAT).

Because we are running out of IPv4 addresses, is IPv6 the new version with a 128-bit address (as opposed to IPv4's 32 bits). Its header looks like this.



- Version: 6 in IPv6
- Traffic class: Same as type of service in IPv4
- Flow label: Ability to label packets that belong to the same flow that the sender wants to be treated differently. Example flows being realtime audio or video.
- Payload Length: Same as Total length in IPv4

- Next header: Same as Protocol in IPv4
- Hop limit: Same as TTL in IPv4
- Source address: Same, but 128 bits
- Destination address: Same, but 128 bits
- Data: Same as in IPv4

Transition from IPv4 is not easy. It’s not possible to replace all devices like some may think (there are billions of devices). Instead we can trick IPv4 hardware to deliver IPv6 by wrapping the entire IPv6 datagram in an IPv4 payload. (Set IANA number to 41 in IPv4 header).

4.4 Generalized Forwarding and SDN

To make it easier and more customizable to perform forwarding (as well as extend existing functionality), the OpenFlow protocol was developed. It’s a protocol that allows servers to tell network switches where to send packets based on a match-plus-action scheme.

There exists a table with matching patterns that will map to actions such as forwarding, dropping, rewriting, or duplicating packets. This makes it very easy to create load balancers on the network level or other software such as firewalls.

4.5 Middleboxes

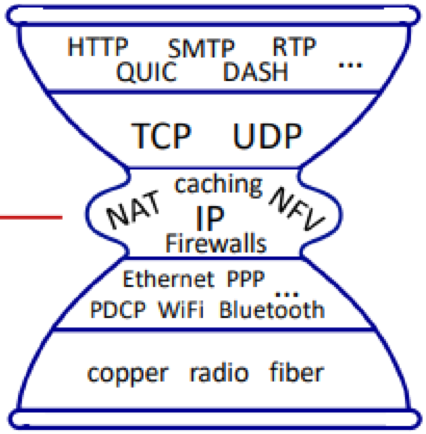
A middlebox is any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host.

They’re essentially services that allow us to modify the packets between end systems. Examples are NATs, firewalls, load balancers or caches.

The IP hourglass, at middle age

Internet's middle age "love handles"?

- middleboxes, operating inside the network


HTTP
SMTP
RTP
QUIC
DASH
...
TCP
UDP
caching
NAT
IP
NFV
Firewalls
Ethernet
PPP
...
PDCP
WiFi
Bluetooth
copper
radio
fiber