**Principles of the network layer**

The transport layer works on an end-to-end service. Segments are sent from a sender to a receiver over multiple hops.

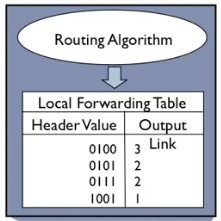
The network layer is a point-to-point service, that means it works on every connection on its path (points) in the network. Every device is using the network layer. Routers do not use the transport layer, but they do use the network layer. When something is sent over the network layer, the data is encapsulated and sent across every point in the network until it ends up at the receiver which decapsulates the data.

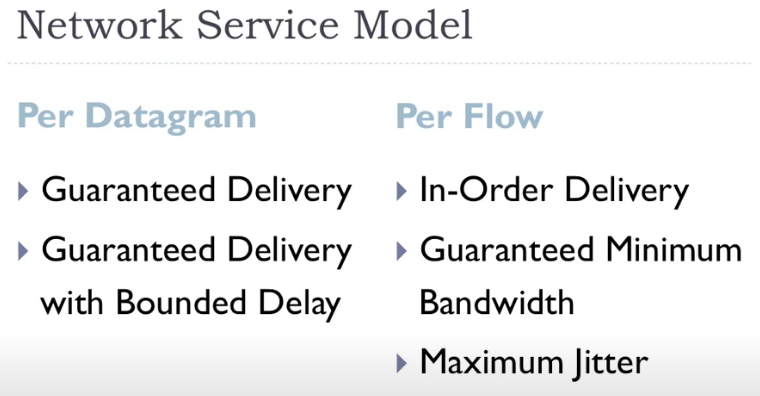
**Routing and forwarding**

The networking layer uses 2 key functions; Forwarding and routing.

Forwarding is local, which means that each router can perform. A router forwards data to another.

Routing is global, which consist of multiple routers forwarding data. Rounting is done by the routers in the network with a local forwarding table. This is created from a routing algorithm:

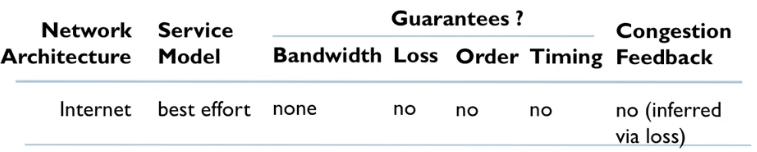




(services the network layer might provide)

(IP does not guarantee delivery or in-order delivery)

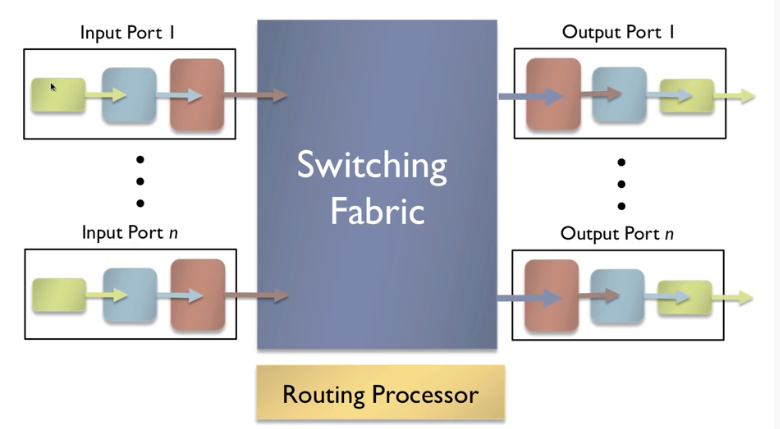
Jitter = the variation in interpacket spacing (packets ariving with different spacing)



**Router architecture overveiw**

The router is involved in 2 processes, routing and forwarding. Routing is a global proces that requires multiple forwards. Forwarding happens when a router forwards data from router to router (point to point).

The router has 4 main components. Input ports, switching fabric, output ports and a routing processor. An illustation of all components:



Input ports = might be multiple, the input port takes an input and puts it on the switching fabric.

Switching fabric = The switching fabric connects the input and output ports.

Routing processor = implements the local forwarding table.

Output ports = sends the datagrams out on the network

**Input port function**

The input port is responsible for unpacking bit-level reception from the physical layer and pass the data to the link layer (maybe Ethernet). The link-layer frame will then be encapsiolated with something like IP which can be passed to the switching fabric. The input port uses a buffer to store packets if packets flow faster than the input port can pass packets to the switching fabric. If the switching fabric is slower than the input port, it will slow some packets in the input port (head of line blocking).

**The switching fabric**

Switching fabrics might work in different ways:

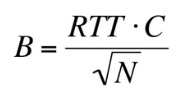
Switching Memory (copy it into output port memory) = old school, slow. Limited by the bandwidth of the network. Because it has to use two crossings per datagram.

Bus (transfer the packet from input to output) = Switching over a bus (limited by the bandwidth of the bus (how many bps can the bus transfer)). This is good for most access and enterprice routers.

Crossbar (unique connections between I/O ports) = complicated, costly and fast. This is not limited by bandwidth (some cisco routers implements crossbar switching with 60 gbps). The fabric has to be really fast to keep up with I/O port.

**Output port function**

The opposite of the input layer: Network 🡪 link 🡪 physical layer by encapsiolation. If the switching fabric is fater than the output port, it uses a buffer to store packets. Sometimes a scheduling policy (are some packets more important than others?) by a queue-structure. (if the buffer is full, the packet is lost). Buffer speed is calculated by:



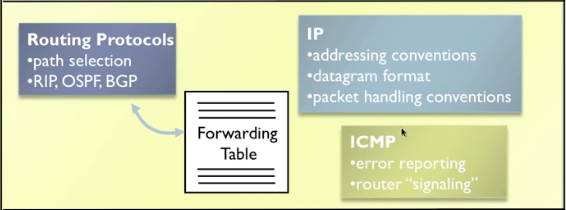
RTT = Round trip time.

C = capacity of the link.

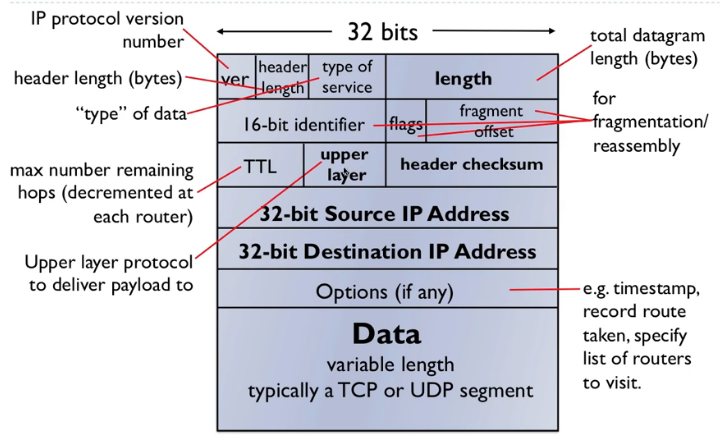
N = number of loads (sometimes not used).

**The internet protocol**

The IP protokol is a well known protol used for connection networks to other networks (interconnected networks... the internet). IP is used for addressing, datagram formats and handing packets with conventions. (with help of routing protokols and ICMP).



**Datagram format**



An IP datagram consist of 32 bits. The highlighted text in the illustration are the most important.

Length = length of the datagram (total datagram in bytes, including the payload).

16 bit identifier, flags and fragment offset = used for fragmentation of datagrams.

TTL = time to live (a number that decrements for each hop, if the number is 0 it is dropped).

Upper layer = The upper layer protocol that the payload is delivered to (UDP, TCP).

Header checksum = like the UDP checksum to check the header.

Source/dest. IP-address = 32 bit number specifies a host (ca. 4 billion different IP-addresses).

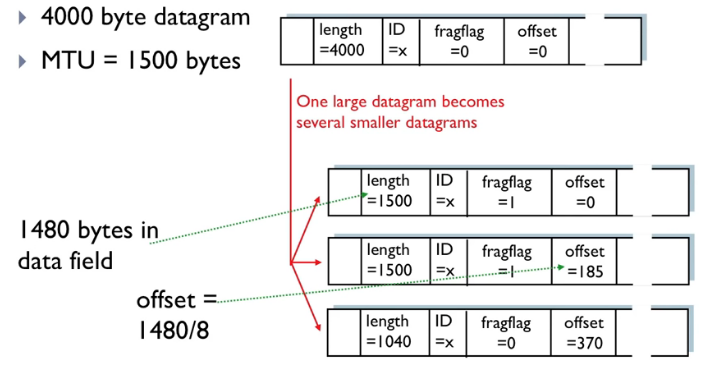
The IP overhead is atleast 20 bytes (UDP and TCP is also 20 bytes).

**IP fragmentation & Reassembly**

MTU (maximum transfer unit) = max size of link level frame (1500 bytes for ethernet, which means a packet cannot be more than 1500 bytes).

IP uses fragmentation if the packet is larger than the MTU. It will split the packet into the MTU (or less). Ip uses an algorithm to fragment and reassemble packets (reassembly happens at end-host).

Example:



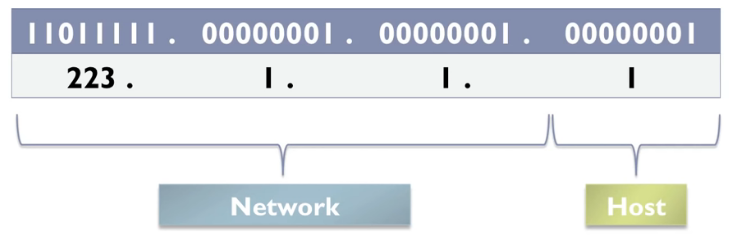
Fragflag marks the packet if the packet has been fragmented (true/false). The last fragflag is marked 0 to mark the last fragment.

Length is the size of the fragmentation. (20 bytes is reserved from the minimum ip packet, in case of the example where the data-field is 1480 bytes).

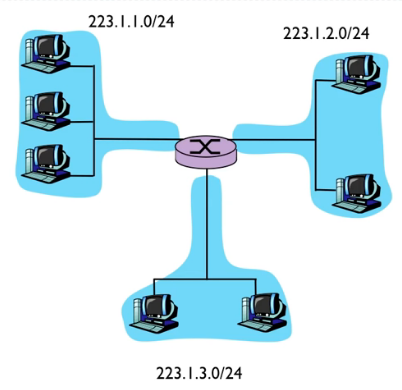
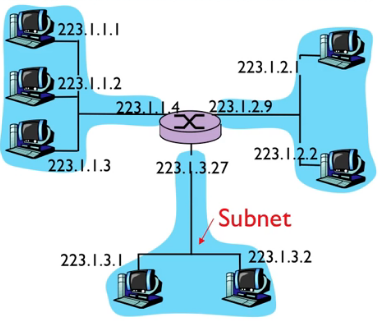
Offset = Marks what part of the packet the fragment belongs to (in 8 byte units, offset = 1 is 8 bytes).

**Ipv4 Addressing**

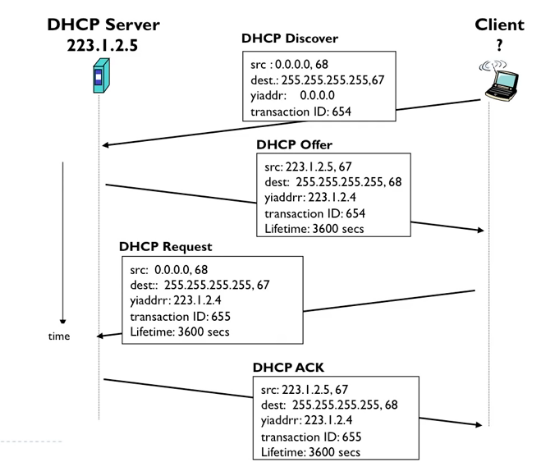
Ipv4 addresses is connected with interfaces in the router. This means the router has multiple IP’s: one for each port/interface (1 for wireless interface and wired interface in a laptop). This is NOT number a host. But an IP is used for numbering an interface. The IP is stored in binary which is not very useful for humans to remember. This is why a dotted decimal notation is used.



one number for each byte (8 bits in the 32-bit address). Some IP-addresses use subnet masks. It might be 223.1.1/24. This is for identifying a network and a host on that network. (some Ips are reserved like 1.1.1.1 for broadcasting which is not usable).



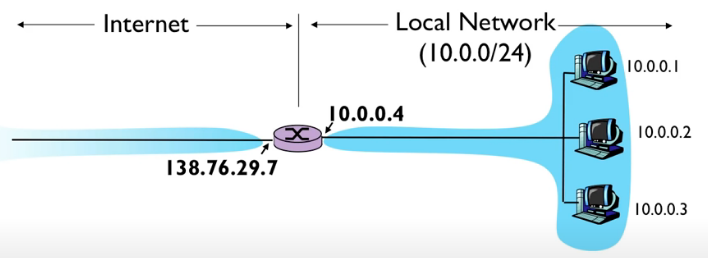
A host-computer can get the IP-address either manually or automatically. Usually automaticly. This is done with the Dynamic Host Configuration Protokol (DHCP).



Once a client makes the discover request, the DHCP server will respond with a src IP and ”yiaddrr” which is the IP address of the client joining a network. The client will make another request with the new yiaddrr which is payload of the DHCP packet. (the client will always send an empty src IP because it is unknown). Once DHCP server accepts the request, it will an ACK and the client will use Yiaddrr as its IP-address on the network. DHCP also asesses the default gateway, DNS server and subnet mask.

**NAT (Network Address Translation)**

NAT is used because the 4 billion available IP’s are already taken by the billion of devices today. This is done by using private addresses within a certain network:



The router will use the same IP-address for all data that leaves the local network. To distinguish the origin of outgoing data, ports are used. The requirement for the router is to create and remember mapping from each private IP to a public source port number. Ingoing data uses the same mapping with ports. This restricts the amount of IP’s and is good for cost-efficientcy, flexibility, convenience and security.

**Ipv6**