50005 - Networks and Communications - Lecture 6

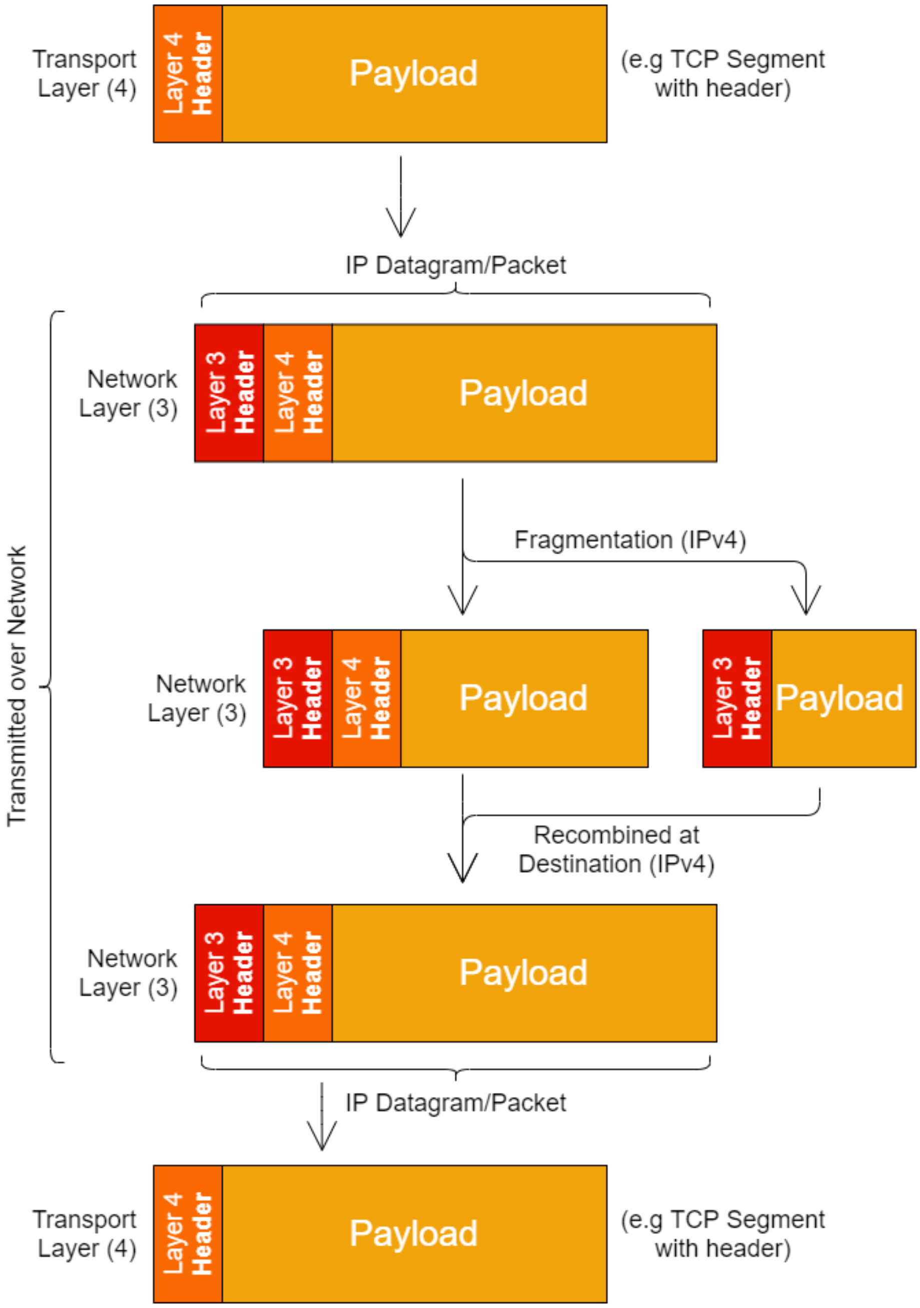
Oliver Killane

17/02/22

Network Layer

Lecture Recording

Lecture recording is available here

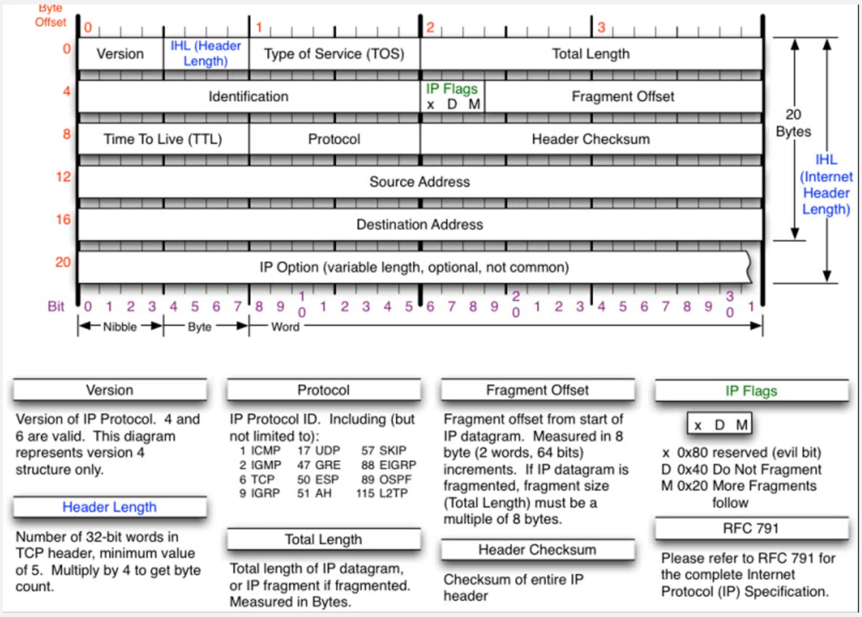


The network layer contains the **Internet Protocol** and is responsible for routing packets though the internet, and across networks with differing hardware, protocol stacks.

Definition: (IP) Internet Protocol

The main protocol use for this layer.

- Datagram Format
- Fragmentation (IPv4 only)
- IP addressing
- Packet handling



Version

Version of IP Protocol. 4 and 6 are valid. This diagram represents version 4 structure only.

Header Length

Number of 32-bit words in TCP header, minimum value of 5. Multiply by 4 to get byte count.

Protocol

IP Protocol ID. Including (but not limited to):

1 ICMP	17 UDP	57 SKIP
2 IGMP	47 GRE	88 EIGRP
6 TCP	50 ESP	89 OSPF
9 IGRP	51 AH	115 L2TP

Total Length

Total length of IP datagram, or IP fragment if fragmented. Measured in Bytes.

Fragment Offset

Fragment offset from start of IP datagram. Measured in 8 byte (2 words, 64 bits) increments. If IP datagram is fragmented, fragment size (Total Length) must be a multiple of 8 bytes.

Header Checksum

Checksum of entire IP header

IP Flags

x D M

x 0x80 reserved (evil bit)
D 0x40 Do Not Fragment
M 0x20 More Fragments follow

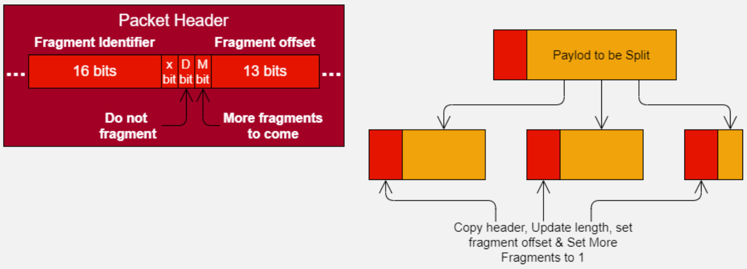
RFC 791

Please refer to RFC 791 for the complete Internet Protocol (IP) Specification.

Note that:

- Type of Service is now called **DiffServ**.
- Most IP Options are not used (security issues).

Definition: Fragmentation

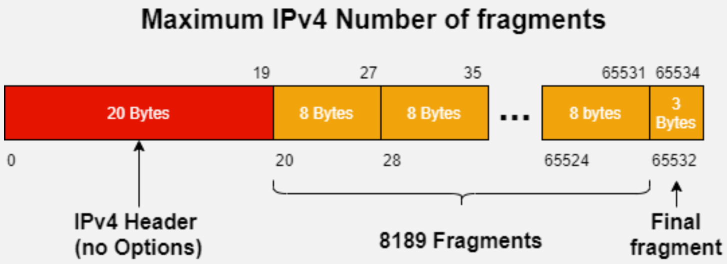


When the data sent to **IPv4** is larger than the **MTU** (maximum transmission unit) of the output link it is being forwarded through the **IP datagram** needs to be split (fragmented).

- Fragmentation at the start, or any intermediate routers.
- Only reassembled at the destination.
- Each fragment is identified by its 16-bit **fragment identifier**.
- Each fragment offset is the offset in units of 8-bytes (all fragments will be multiples of 8 bytes, plus a last byte).
- The **more fragments** bit (**M**) informs the receiver if it should expect more fragments to arrive, this is set when an intermediate router fragments a packet.

Maximum Fragments

It is not possible to fit the maximum number of fragments allowed by the 13-Bit fragment offset (8192) inside an **IP Datagram/Packet**.



- **Total Length** in the **IP Header** is 16-bits, hence maximum size is $2^{16} - 1 = 65535B$ (65536 sizes including 0).
- The minimum header (for IPv4) with no options is $20B$ long.
- Hence the maximum amount of data that can be stored as payload is $65515B$
- We can calculate the maximum number of $8B$ fragments as $\left\lfloor \frac{65515B}{8B} \right\rfloor = 8189$.

Hence it is only possible to have 8189 $8B$ fragments, with a single $3B$ fragment on the end.

Terminology

Networks Types

PAN	Personal Area Network	(e.g phone connected to PC, connected to bluetooth speakers)
LAN	Local Area Network	(Small network in a single geographical location, e.g home PC connected to home wireless network)
MAN	Metropolitan Area Network	(City wide network, e.g subway digital signalling network spanning large parts of the city.)
WAN	Wide Area Network	(Over a large geographical area, largest example is the **internet**.)

Devices

Definition: Repeaters/Hubs

In the **Physical Layer/Layer 1** and simply repeat wireless network traffic to boost signal. They do not processing of any kind, and just repeat any signal they intercept.

Definition: Switches/Bridges

In the **Data Link Layer/Layer 2** and make interconnections based on **MAC** addresses (which identify a given **NIC**).

Definition: Gateways/Multi Protocol Routers

In the **network Layer/Layer 3** to mak e decisions on forwarding packets (as well as splitting them - e.g fragmentation) based on **IP Addresses**.

To connect two **IP**-based networks together, a gateway (or router acting as one) is required between them.

Other Internet Protocols

Definition: The Internet

A collection of **Autonomous Systems** (separate networks, can run independently of each other) connected together by **backbones** (large long distance network infrastructure to link networks).

Designed in accordance with the principles of RFC 1958.

- Applications send data through a connection-oriented or connectionless transport layer protocol.
- Transport layer creates TCP Segments or UDP datagrams.
- Network layer converts TCP/UDP into **IP Datagrams**.
- Data Link Layer pass datagrams between routers, across networks.
- Physical layer (cables) transmits data.

Definition: (ICMP) Internet Control Message Protocol

Used for sending standardised control messages inside **IP Datagrams** (e.g ping is an ICMP Echo Request, Destination host unreachable).

- Each message has a type (e.g destination unreachable, time exceeded and more)
- Each message type also has a code (e.g destination unreachable (3), unsupported protocol (2))

For example **ping** sends an **ICMP** (type = 8, code = 0) which is responded to with an **ICMP** reply of (type = 0, code = 0).

Definition: Dynamic Routing Protocols

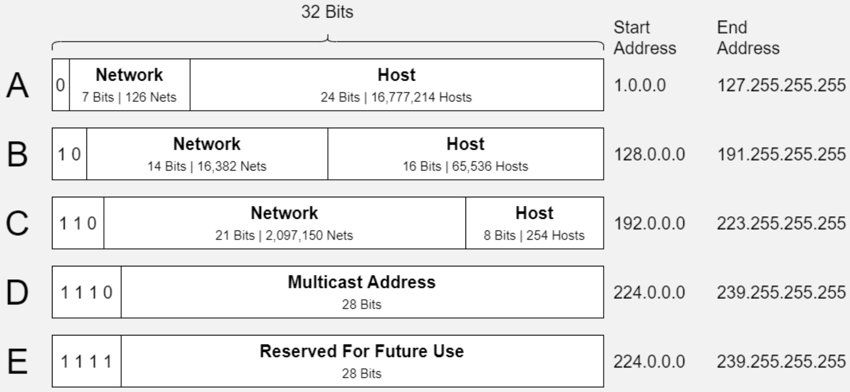
Include RIP (Routing Information Protocol), EIGRP (Enhanced Interior Gateway Routing Protocol), OSPF (Open Shortest Path First), BGP (Border Gateway Protocol). They determine how a packet is routed through a network, and create/manage routing and forwarding tables

IPv4 Addressing

- Addresses are contained in 32 bits.
- Displayed as $XXX.XXX.XXX.XXX$ where $XXX \in [0, 255]$.
- Each IP address is associated with an interface (not a host), so hosts may have more than one address.

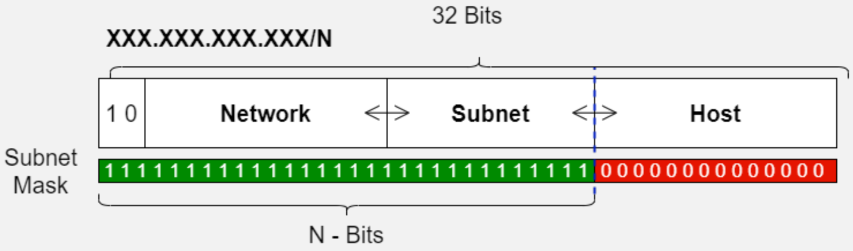
Definition: Classful Addressing

IP Addresses are split into several classes, each with a different length prefix to denote the organisation.



Critical Issue: All hosts on the network must share the network address section, so if an organisation hast hosts with several different IPs, it must publicly announce/claim multiple network identifiers.

Definition: Classless Addressing



A single network address is used for the entire organisation, internally addresses are divided into subnet addresses and host identifiers.

- External routers only consider the network address, and forward to a router of the associated organisation.
- Subnet routers apply the subnet maskand check if the IP is in their subnet, or if they need to forward to another subnet in the organisation.
- Once a host is found, routers know which interface to forward packets to.
- Network, Subnet and Host can have their sizes different for each network, according to the prefix length (N).
- The any-length prefix scheme is called **CIDR** (Classless Inter-Domain Routing).
- Routers attempt to match the longest prefix in order to select the correct network to pass a packet onto.

A simple python script for conversion is available with this lecture:

```python
# Super basic IP mask creation, string conversions and range.
from typing import Tuple

def ipv4_to_str(ip: int) -> str:
    assert(0 <= ip < 2**32)
    return ".".join([str((ip // (2**(8 * i))) % 256) for i in range(3,-1,-1)])

def get_mask(prefix_len: int) -> int:
    return (2**(prefix_len+1) - 1) * 2 **(32 - prefix_len)

def get_ipv4(ip: str) -> int:
    ip = ip.split(".")
    assert(len(ip) == 4)
    ip_num = 0
    for sub in map(int, ip):
        ip_num *= 256
        assert(0 <= sub < 256)
        ip_num += sub
    return ip_num

def apply_mask(ip: str, mask: int) -> str:
    return ipv4_to_str(get_ipv4(ip) & get_mask(mask))

def get_range(ip: str, mask: int) -> Tuple[int, int]:
    ip = get_ipv4(ip)
    subnet_mask = get_mask(mask)

    return (ip & subnet_mask | ((2**32 -1 ) & (~subnet_mask)), ip & subnet_mask)

# Example code
# print(apply_mask(input("Input IP:  "), int(input("Prefix:    "))))
# (bot, top) = get_range(input("Input IP:  "), int(input("Prefix:    ")))
# print(f"From {ipv4_to_str(top)} to {ipv4_to_str(bot)}")
```

It is also covered in the lecture below:

Lecture Recording

Lecture recording is available here

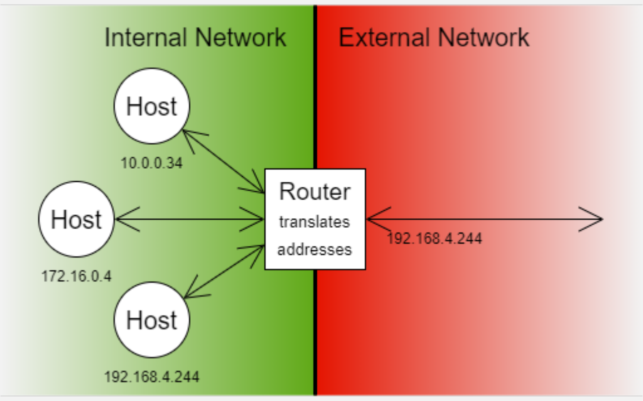
Definition: (DHCP) Dynamic Host Configuration Protocol

Allows hosts' interfaces to safely be assigned an **IP Address**.

- On boot, host broadcasts a **DHCP Discover** packet, a listening **DHCP** server will respond with an assigned **IP Address**.
- **DHCP** servers can maintain static mappings (hosts to addresses), and also assign different addresses each time a host connects.
- Hosts lease an **IP**, and must refresh periodically (to prevent hosts hogging an **IP**).

For example your router requests an **IP** from your **ISP**'s **DHCP** servers periodically.

Definition: (NAT) Network Address Translation



An attempt to solve the **IPv4** address shortage. Translates many private **IP**s into a single public **IP address**.

- Translation occurs when packets leave or enter the local network.
- On the local/internal network, every computer gets a unique **IP address**.

This is managed with a table of mappings between hosts & their processes (**Transport Layer**/**Layer 4** header contains this information) and ports on its own **IP**.

The following address ranges are also *private* and can only be used in local networks:

10.0.0.0	→	10.255.255.255/8	16, 777, 216 addresses
172.16.0.0	→	172.31.255.255/12	1, 048, 576 addresses
192.168.0.0	→	192.168.255.255/16	65, 536 addresses

There is much valid criticism of **NAT**:

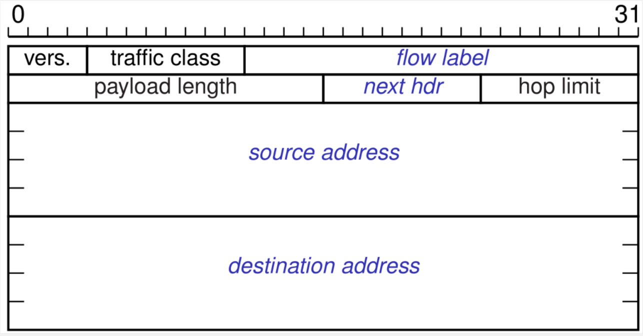
- It violates the **IP Model** (that each IP Address uniquely identifies a host).
- It changes the internet from connectionless to connection oriented (as router must keep track of connections, associate with a mapping for translation).
- It violates the fundamental rule of the protocol stack: That layers do not make assumptions about protocols above. As **NAT** uses **Transport Layer** information, if new transport protocols are used, **NAT** does not work.
- It cannot easily support new transport protocols (due to the previous port).
- Many Peer-to-Peer protocols require full connectivity between hosts which **NAT** cannot provide. Hence prot forwarding, TURN relays, NAT punching holes, 3rd party servers and other solutions are required.

Special IP Addresses

0.0.0.0/0	The **default route**, used when no other **IP address** matches.
0.0.0.0/8	This host on this interface. Must not be sent, only used to acquire an **IP Address**.
127.0.0.0/8	”Loopback” (reference to the host), that can be sent (127.0.0.1 is localhost).
169.254.0.0/16	”Link Local” (something went wrong which acquiring an **IP Address**).

Ipv6

Definition: IPv6



Intended to fix **IPv4**'s address shortage (**IPv6** has $\approx 3.8 \times 10^{38}$ addresses), while also:

- 128 bit addresses, displayed in hexadecimal (e.g 2001:630:12:600:1:2:0:10b)
- Reducing routing table size and simplifying the protocol for higher performance.
- Improving security.
- Better support for "type of service" (now DiffServ in **IPv4**).
- Support scopes with multicasting (sending a packet to many hosts in a certain scope, e.g network).
- Support roaming hosts without address changes (more on roaming scopes here)
- Better support coexistence of old and new protocols, while making it easier to develop new ones.

It has several key differences with **IPv4**:

- Fragmentation is done by end-systems. Hence packets do not have to deal with fragmentation.
- No header checksum, it is redundant as both the **Transport** and **Data Link** Layers have error detection features.
- Fixed length header is easier to process, **IPv4**'s options were almost always unused.
- Better modularity for extensions.

Extensions are done by placing an extending header after the **IPv6** header.

- Hop-by-hop options (provides information to routers, e.g quality of service)
- Routing (Provides a full or partial route to follow)
- Fragmentation (Information for end systems)
- Authentication (Sender identity verification)
- Encrypted payload (Info on the payload)
- Destination Options (extra information, e.g mobile **IP** (moving networks but maintaining the same IP address))

Routers

Lecture Recording

Lecture recording is available here

Routing Requirements

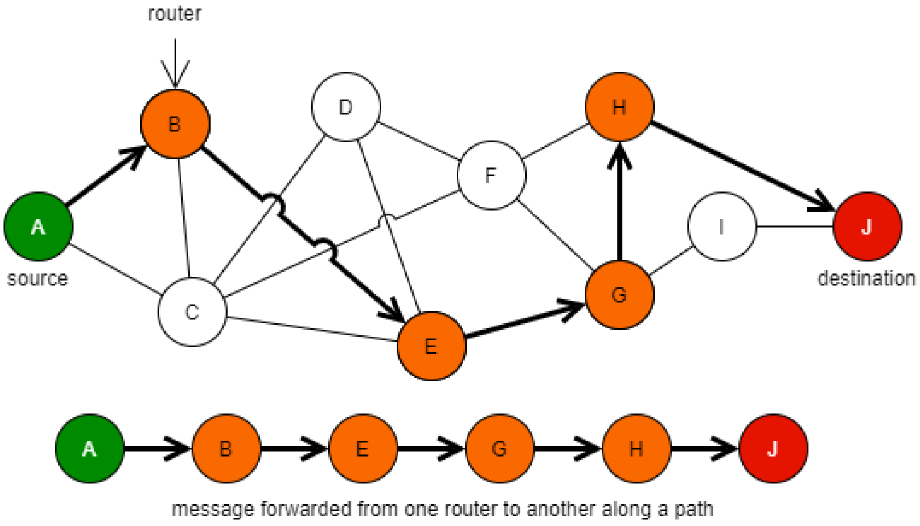
A routing system needs to provide facilities for moving data from source tro destination as well as:

- Allow for multiple hops on nodes in the network.
- Be able to consider the topology of the network to choose appropriate routes.
- Perform load balancing.
- Allow for/deal with network heterogeneity (different types of networks connected).

The internet is a **packet switched** network, providing a connection-less service (no setup/teardown phase, each message is independent and self contained with no steup/tear down phase).

The internet is also a best effort service, and does not provide guarentees of delivery, maximum latency, bandwidth, congestion indication or in-order delivery.

Datagram Networks



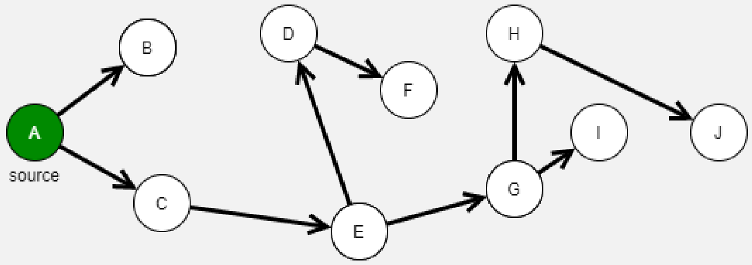
- There are potentially many paths for the same source to destination.
- Paths can be asymmetric (path from $A \to J$ is different from return path $J \to A$).

Each router uses a **forwarding table** to determine which router to forward packets to, based on their final destination.

Routing

Definition: Sink Tree

A tree from a source node, to every destination node, where each path in the tree is the optimal route/shortest path to the destination. As a tree, there are no cycles.



Definition: Djikstra's Algorithm

Each arc is labelled with a cost (e.g delay, hops, some function of parameters potentially including congestion).

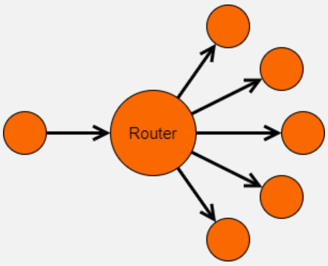
1. Visited = {}
2. Add the start node (at *distance* = 0)
3. Loop wihile there are unvisited nodes:
   (a) Label each fringe node (one arc from a visited node) with the minimum of (weight from visited node) + (weight of connecting arc).
   (b) Add the closest node (based on previous step)
4. The shortest paths are the distances each node is labelled with.

Routers cooperate to find the best routes between all pairs of nodes in the network.

Definition: Shortest Path Routing (SPR)

Use **Djikstra's Algorithm** to determine the shortest routes from each router, to every destination. The forward packets accordingly.

Definition: Flood Routing



Forward incoming packets to every outgoing link. Except for the link the packet was recieved on.

We can use several strategies to avoid drowning the network in packets:

- **Hop Counter** Disgard a packet after it reaches a maximum number of hops.
  Must decide on a correct number of hops to avoid drowining, but allow packets to reach their destination.
- **Forward Once** If recieving the same packet again, do not forward again.
  This solves the issue where packets are sent though cycles (e.g $A \rightarrow B \rightarrow C \rightarrow A$), however it requires storing sequence numbers per source address to identify packets. Furthermore must decide on how lonbg the sequence numbers are stored.
- **Selective Flooding** Flood only in selective directions.
  Rather than flood every outgoing link (except packet source), send to only some of the outgoing links, based on some heuristic (to decide which diurections make the most sense).

Flooding always chooses the shortest path (all paths explored in parallel), however it creates significant overhead (must send many packets at every router in every path).

Use case is when the packet must be recieved, but when the route to the destination is unknown.

Definition: Distance Vector Routing (DVR/Bellman-Ford)

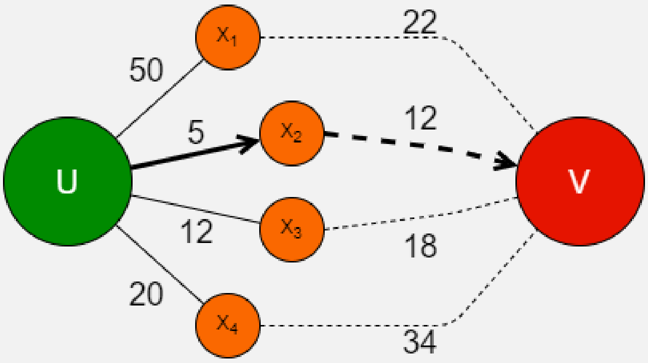
Both **Flood Routing** and **Shortest Path Routing** are static and do not take into account current network conditions (e.g network load). **DVR** is dynamic and does consider this.

- Every router advertises its costs to each destination.
- Router's use their cost to neighbours, and their neighbour's cost to determine how to route packets for the minimum cost, and to then update and advertise their cost.

This is expressed by the **Bellman-Foprd Equation** for the cost from node u to node v:

$$D'_u[v] = \min_{x \in neighbours(u)} (cost(u,x) + D_x[v])$$

(Cost from u to v is the minimum of the cost from u to a neighbour, plus the cost of that neightbour to v)



However there is a **count-to-infinity** problem. When a node goes down, and routers continually update their costs based off seachothers, resulting in the cost incrementing constantly.

$$A \rightarrow C = B \rightarrow C + 1$$
$$B \rightarrow C = A \rightarrow C + 1$$

We can resolve this by defining infinite cost as:

$$cost\ \infty = \text{longest acceptable path} + 1$$

Examples of distance vector algorithms include **RIP**.

Definition: Link State Routing

A replacement for **DVR**.

- Broadcasts alll information on network topology to all rputers.
- Each router can use this to calculate a **sink tree**.
- Identifies neighbours using a special ”hello” packet, to which neighbours respond with their network address.
- Link costs is determined using a special ”echo” packet, and measuring the **rount trip delay**.

The basic algorithm for each router is as follows:

1. Get direct neighbours & their network addresses (so they are uniquely identifiable on the network) (”hello”).
2. Calculate the cost of sending packet to each neighbour (”echo”).
3. Build a **Link State Advertisement/LSA** describing the router, its connections to its neighbours.
4. Send the **LSA** packet to every router on the network (**flooding**).
5. Recieve **LSA** packets from every other reouter on the network.
6. Now the router has the status of all links between all routers, it runs dijkstra’s algorithm locally, to create a **sink tree** for use in routing.

This algorithm allows better routes to be chosen using current network conditions.

However routers may redirect traffic towards the best routes so much, that these routes become overloaded, and are no longer the best routes.

An example of **Link State Routing** is **OSPF**.

We can compare **DVR** and **Link State Routing**:

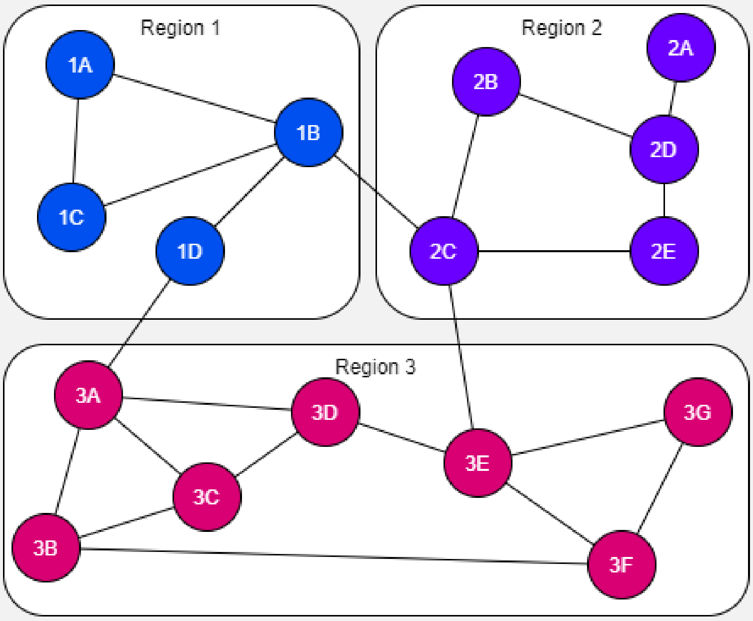
	Distance Vector Routing	**Link State Routing**
Network Info	Local	Global
Computation	Global	Local
Synchronisation	Gradual (routers update & advertise)	Instance (once the **SPR** computation is done)

Definition: Heriarchical Routing

All the previous routing methods are difficult to scale as each router needs to know about all other routers. On the scale of the internet, memory and processing power requirements would be too high.

To solve this the network is split into regions. With different algorithms used for **intra-region** (inside regions) and **inter-region** (between region) routing.

- Can scale the network massively.
- Suboptimal routes chosen between node in different regions, (we just know which region to go to)
- Can use other algorithms within the regions, each group can effectively been its own, autonomous network with its own design, structure, routing algorithms.
- 2/3 levels of regions are generally enough.



We can consider each region a different network, all connected together.

Definition: Broadcast Routing

Another way to solve scaling, send to every host on a network (only feasable for **LAN**s and small **WAN**s), even through we do not know the route to a destination, by sending the packet to every host, it is sure to be recieved.

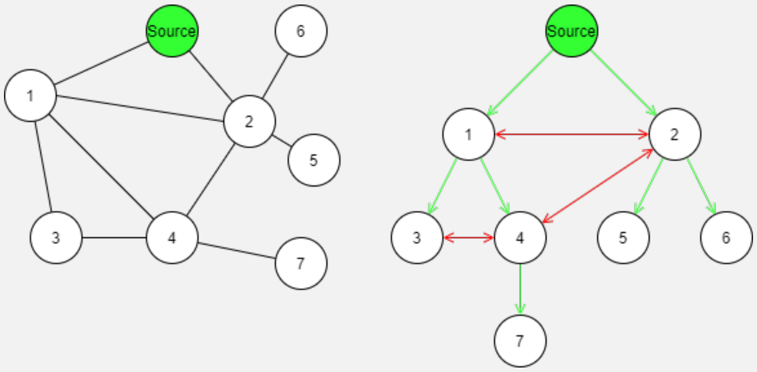
- **Send packets individually** Not efficient
- **Flood Routing** Acceptable if the flood can be limited
- **Multi Destination**
  A list of destinations is sent with the packet. Routers check this list, splitting the list and forwarding the packet to its neighbours.

  However the packet must contain all destinations (size limitations).
- **Multicast Routing** See definition.

Definition: Reverse-Path Forwarding (RPF)

Used to construct spanning trees from a router, for a low cost.

- Every router forwards/broadcasts a packet on every adjacent router, except the router the packet was recieved from (like **flooding**).
- Routers only accept packets if it is on a direct path from the source.
- Hence the paths of packets forwarded, and accepted represents a spanning tree from the source router.
- Note: This can also be used to detect and prevent IP spoofing (as packet will come from an odd path, given the spoofed IP)



Definition: Multicast Routing

Sending a message to a subset of the nodes (groups, each with a group id).

A first solution is to construct a spanning tree at each router, and prune all paths that do not contain members of the group we want to send to.

Alternatively we can use **Core based trees**.

- A single spanning tree per group, with a root (central to the group to reduce cost between it and members of the group).
- To send a multicast message to the group, just send it to the core, which will retransmit to all nodes in the group.
- Not optimal for all sources, however scalable and much lower overhead.

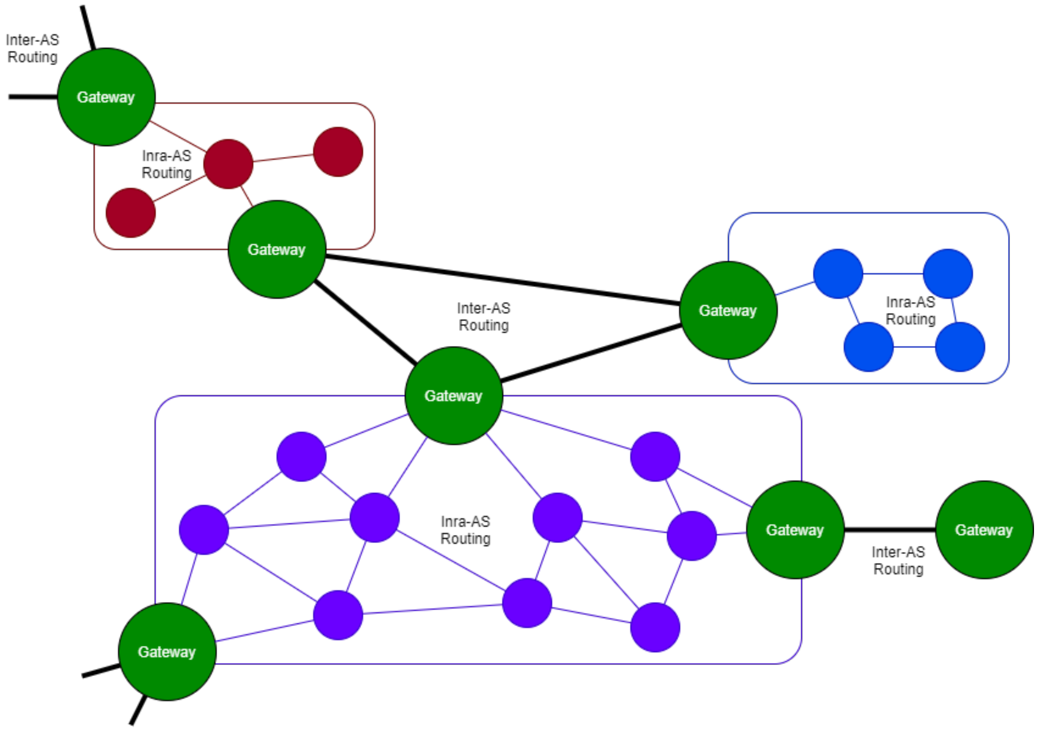
Note this is how it is done in the internet (multicast IP Address/Broadcast Address is effectively the core for an entire network).

Inter-AS Routing

Lecture Recording

Lecture recording is available here

Inter-AS Routing	**Intra-AS Routing**
Routing between autonomous systems (e.g between two different networks)	Routing within an autonomous system (e.g within a **LAN**).
Autonomous systems can be hetergeneous (different protocols, routing algorithms, topologies, hardware) so use **Gateways** to link between.	Within an autonomous system (depending on size) typically uses one design controlled by one organisation.
Cannot support optimal routes at scale, but makes best attempt practical.	Attempts to provide optimal routes on a smaller network.



External → *External* Gateway (Inter-as router) recieves packet, if it can forward to the next gateway it does, if another gateway in its AS can send it, the packet is routed by intra-as routers through the network to the gateway to send on.

External → *Internal* Gateway recieves the packet, then sends it on to intra-as routers to route to the destination.

Internal → *external* Intra-as router sends packet on to a gateway that advertises it can reach the destination, gateway then forwards to the relevant gateway, routing across networks.

Internal → *Internal* Intra-as routers route packets.

Definition: Open Shortest Path First (OSPF)

A **link state routing** algorithm to replace **RIP** (a distance vector routing algorithm).

- Algorithm is publically available for anyone to implement.
- Supports different distance metrics (hops, delays, etc).
- Can adapt dynamically to changing network topology (nodes added or removed).
- Supports routing based on **ToS** (Type of service).
- Supports load balancing (not overwhelming routers e.g by flooding)
- Offsers some scurity features (though some have been compromised).

It also supports hierarchical routing. It is possible to split an AS into several ”areas”, then each area has one or more ”area border routers” which are in a ”backbone area” (contains all border routers) to route traffic between the ”areas”.

Definition: Border Gateway Protocol (BGP)

The Inter-AS routing protocol used in the Internet.

- Adjacent routers maintain connections for reliability.
- Gateways transmit reachability information to routers inside an AS.
- Good routes determined based on reachability information and routing policies.
- Routers only check for & discover new paths if allowed.
- Uses a **path-vector** protocol (based on **DVR** but paths instead of distances are announced).

BGP advertises routes/paths to networks:

- Destinations are denoted using the address prefixes (see subnetting).
- ASes may not propagate an advertisement by a gateway, as doing so would imply the network is willing to carry traffic through the AS.
- Routers can aggregate prefixes (merge prefixes together)

Example: Aggregating Prefixes

We can merge several ips (with prefixes) into one, with a shorter subnet mask.

$$\left.\begin{matrix}127.134.126.0/24\\127.134.127.0/24\end{matrix}\right\} \to 127.134.126.0/23$$

Here the 24th bit is the only difference between the two, and hence we can reduce the subnet mask size.

This is also referred to as *supernetting*.

In **BGP** each AS has a unique identifier (**Autonomous System Number** (**ASN**))and several attributes:

- **AS-PATH** Sequence of AS identifiers through which the advertisement was sent
- **NEXT-HOP** Next IP address to forward packets towards advertised destination (resolves ambiguity when there are multiple AS reachable through multiple inferfaces)

The **BGP** import policy determines to accept or reject route advertisements.

Router preference is ranked according to:

- Policy used.
- Shortest **AS-PATH**
- closest **NEXT-HOP** router.

The *count-to-infinity* problem is sovled by *path exploration/hunting* (actiuvely seeks paths), furthermore routers can send withdrawl messages (e.g before taking a node down, can tell others to remove the path).

This allows routers to identify invalid paths, at the expense of some delays.