RCOMP - Redes de Computadores (Computer Networks)

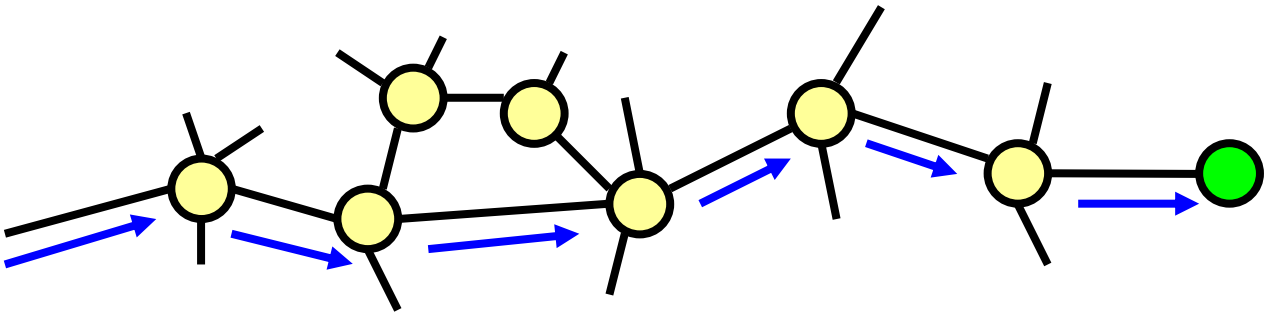
2023/2024

Lecture 06

- IPv4 routing.
- Static routing and dynamic routing.
- Routing protocols: RIP, RIPv2, EIGRP and OSPF.
- Autonomous systems and route redistribution.

Intermediate nodes – switches and routers

Intermediate nodes have a key role in any kind of packet-switched network. In such networks, data packets are retransmitted between a sequence of intermediate nodes so that, ultimately, they reach the required destination node.



Intermediate nodes receive packets intended to other nodes, they forward those packets to other intermediate nodes and ultimately to the destination node. Layer two intermediate nodes are usually called switches, and they forward layer two frames.

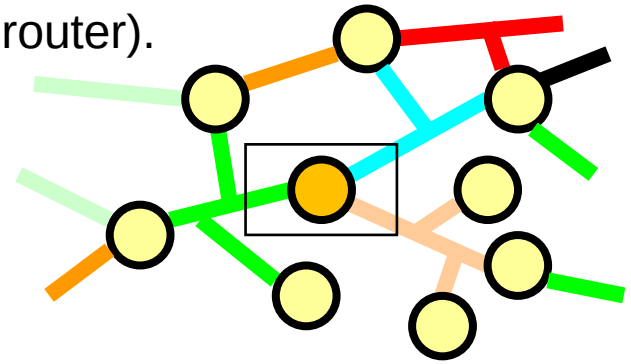
IP packets (often called IP datagrams) also reach their destination due to the effort of intermediate nodes, of course, operating at layer three. These layer three intermediate nodes are named **routers** or **gateways**.

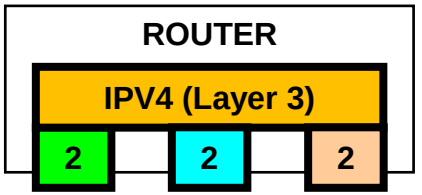
Routers or gateways

Routers forward (retransmit) layer three packets, they operate by using a layer three protocol, nowadays, mostly IPv4 and IPv6.

Usually routers have several network interfaces (layer two connections), and they may even be of different types, e.g. Ethernet, ATM, DSL, 3G. These interfaces are used to receive and then retransmit layer three packets.

When a router receives a packet, the destination address in the header is analysed. Based on the destination address the router must then decide to where it should be sent (some neighbour router).

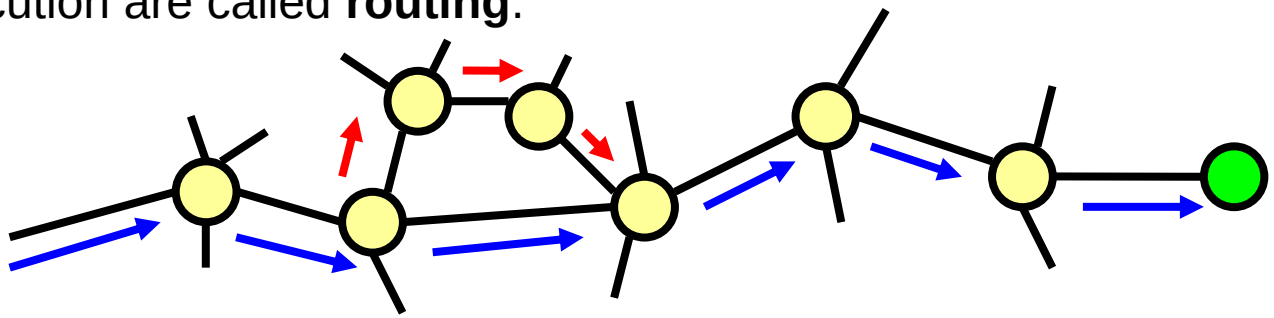




The images above represent one central router (inside a rectangle). It has three network interfaces and a total of eight neighbour routers (directly connected routers). For each packet this router receives, it will have to decide to which of the eight neighbours the packet should be forwarded. Neighbour routers are also known as **next-hops** as they represent a hop in the packet path from its origin to its destination.

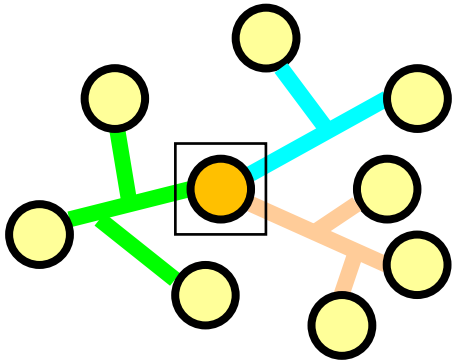
Routing

The essential decision a router must make for each packet it receives is **where should it be sent to,** in other words, **what is the correct next-hop**. This decision taking and the its execution are called **routing**.



Each router makes its own local decision by picking the appropriate next-hop. For a packet to reach its destination, all routers in the path must make the correct choices. Nevertheless, alternative paths may exist, and less optimal or wrong decisions may be corrected by other routers.

One thing to bear in mind is that routers interact only with other neighbour routers, those are the only possible next-hops. A remote router can never be a next-hop. On the left image, the central router has eight neighbour routers, so there are only eight different possible next-hops.



Routing tables

IP packets emitted somewhere on the internet must be routed to the desired destination node address. This will only happen if all routers along the path take correct decisions.

Routing decisions (picking the appropriate next-hop) are taken by using information present in the **routing table**.

Routing Table	
DESTINATION	NEXT-HOP

Each router has its own routing table, they are made of a sequence of lines. Routing tables lines have two fundamental elements:

DESTINATION – identifies a destination by specifying an **IP address and a network mask or prefix-length**. Usually this represents a network, but it can also be a single node (32-bits prefix-length) or a set of aggregated networks.

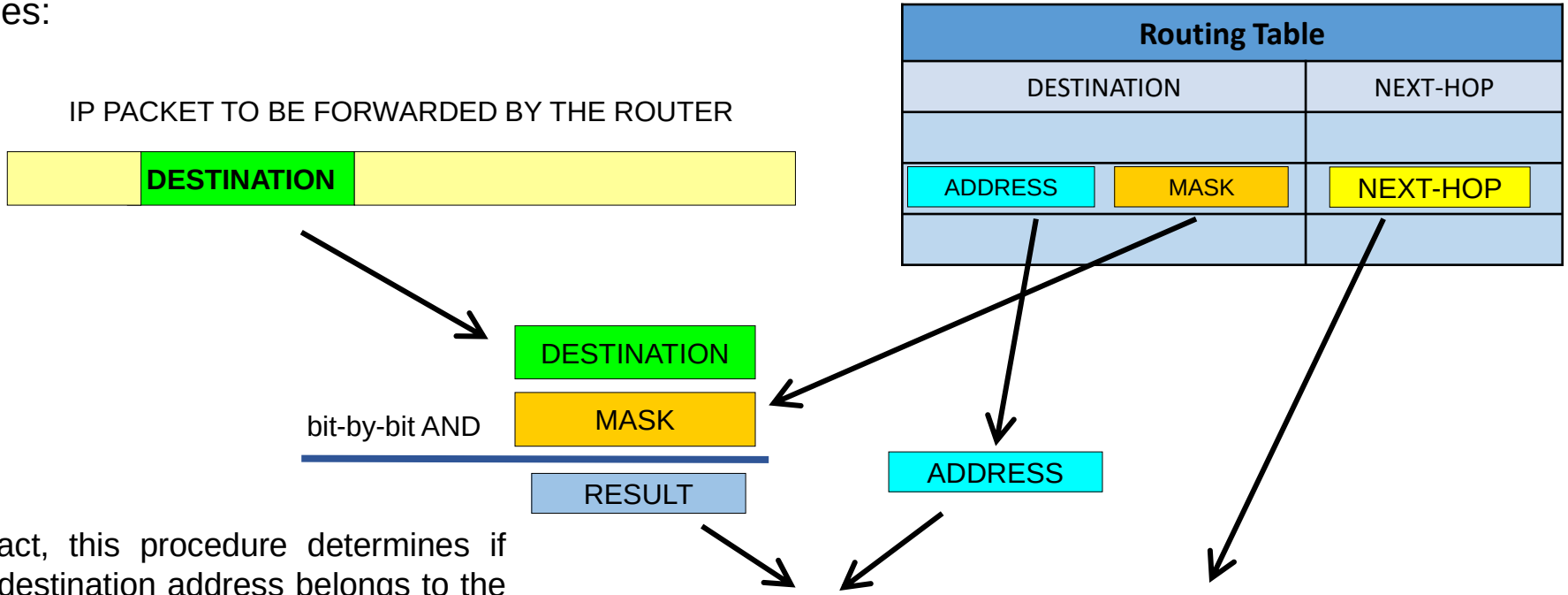
NEXT-HOP – to where packets should be forwarded to ultimately reach such DESTINATION. It's the IP address of a neighbour router, that router is the next intermediate node on the path to reach the intended DESTINATION.

Routing and routing tables

When a router (and in fact any node) wants to send an IP packet, it will search the **datagram's destination node address** in the **destination column of the routing table**.

If found, the datagram is **sent to the corresponding next-hop** and the router mission is finished. If, after sequentially examining all the routing table lines, no match is found, **the datagram will be discarded**.

The following schema explains how routers match destination addresses against routing table lines:



In fact, this procedure determines if the destination address belongs to the network/prefix at the routing table line. Routers can perform these operations very fast.

Local networks and the default route

A node can have several **network interfaces**, each connected to a different IP network. Under the node's point of view, these are **local networks**, others are **remote networks**. A node can send to local networks all by itself, to send to remote networks at least one router must be used.
Local networks are also part of the routing table, yet for those lines, the next-hop value is the network interface name instead of an IP address.

Routing tables can't hold all possible destinations available on the internet. This issue is solved by adding one last line called **default-route**. The default-route is 0.0.0.0/0, therefore, it matches any destination address.
This means if some destination address doesn't match any previous line, it will ultimately always match the default-route at the last line. The default route´s next-hop is usually known as **default-gateway** or default-router.

On the right, there's an example of a routing table. Just by looking to it, we can tell the node is connected to two local networks. We can also see that the default gateway is 192.168.10.200.
Mind that all next-hops must belong to a local network, otherwise, something is wrong.

DESTINATION	NEXT-HOP
192.168.10.0/24	Ethernet interface 1
172.14.0.0/16	Ethernet interface 2
194.121.12.0/24	172.14.5.100
0.0.0.0/0	192.168.10.200

The minimal routing configuration

Every IP node has a routing table, not just routers. The humblest imaginable routing configuration is for a node connected to a single isolated local network; this will result in a single line routing table.

DESTINATION	NEXT-HOP
Local Network	Network Interface

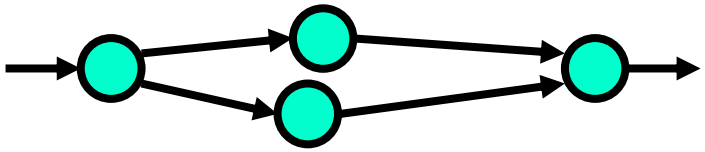
Most end nodes, like clients and servers, have only one network interface, they are, therefore, usually connected to a single local network. But unlike the above scenario, the local network is most likely not isolated, there's a router providing access to other networks and ultimately to the internet. This router's address on the local network is the default-gateway.

If a node doesn't know the default-gateway or other local router's address, it won't be able to send packets to remote networks. All communications would then be restricted to local networks.

DESTINATION	NEXT-HOP
Local Network	Network Interface
0.0.0.0/0	**Default-gateway**

Truly, the routing configuration required for a simple end-node is just the default-gateway. This results from local networks being already known from the moment the IP addresses and network masks are assigned to each network interface.

Alternative paths



When there's only one possible path along the network, two columns routing tables are suitable. Yet, this is not the case if there're several alternative paths.

Having alternative paths has obvious advantages, networks may become fault tolerant, and in addition, those paths can be used for traffic load balancing.

If alternative paths exist, this is what will happen on some routers' routing tables: **same destination on multiple lines**.

There must be a criterion to decide the best alternative, not just the first match.

DESTINATION	NEXT-HOP
192.168.10.0/24	ETHERNET 1
172.14.0.0/16	ETHERNET 2
194.121.12.0/24	172.14.5.100
194.121.12.0/24	192.168.10.2
0.0.0.0/0	192.168.10.200

To sustain a decision for the best path, the routing table needs one additional column called COST or METRIC. It's a numerical value expressing how adverse to performance will be using that line. When there are several alternatives to reach the same destination, the router uses the one with the **lower cost**.

The exact formulas used to calculate the metric/cost are very diverse, they can take into account values like transmission rates along the path, delays, MTU and routers' traffic load.

Dynamic Routing concepts

Routing tables can be manually created (**static routing**), but we can also make routers talk to each other to automatically build routing tables (**dynamic routing**).

Application protocols used by routers to enforce dynamic routing are called **routing protocols**, of course, this will only work if all routers talk the same language (same routing protocol).

Even though that's a pleasant facet, the big motivation for dynamic routing is not avoiding the burden of manually creating routing tables.
Dynamic routing not only ensures the initial building of tables but also **keeps them updated**. This is fundamental, it means when there's a change in the infrastructure, that change is automatically **reflected in the routing tables**. This can't be achieved through static routing.

The process of reflecting in the routing tables a change in the infrastructure is called **convergence**. The time it takes, the **convergence time**.

Taking advantage of alternative paths (fault tolerance and load balancing) requires the use of dynamic routing. This comes obvious, as, with static routing, packets will always follow the same exact path.

Routing protocols

Routing protocols are used by routers to inform neighbour routers about links availability and performance. **The idea is: if every router transmits to neighbours what it knows, ultimately, every router will know everything about the entire infrastructure.** Routers make the layer three infrastructure fabric, because they interconnect networks, they know all networks.

Operational details vary depending on the routing protocol:

- Information may be sent to broadcast/multicast addresses, or, once neighbour routers are detected, it may be sent in unicast to the neighbour routers.
- Once detected, neighbour routers may be monitored (link-state).
- Information may be sent periodically or may be sent only when there's a status change.

There are two major classes of routing protocols

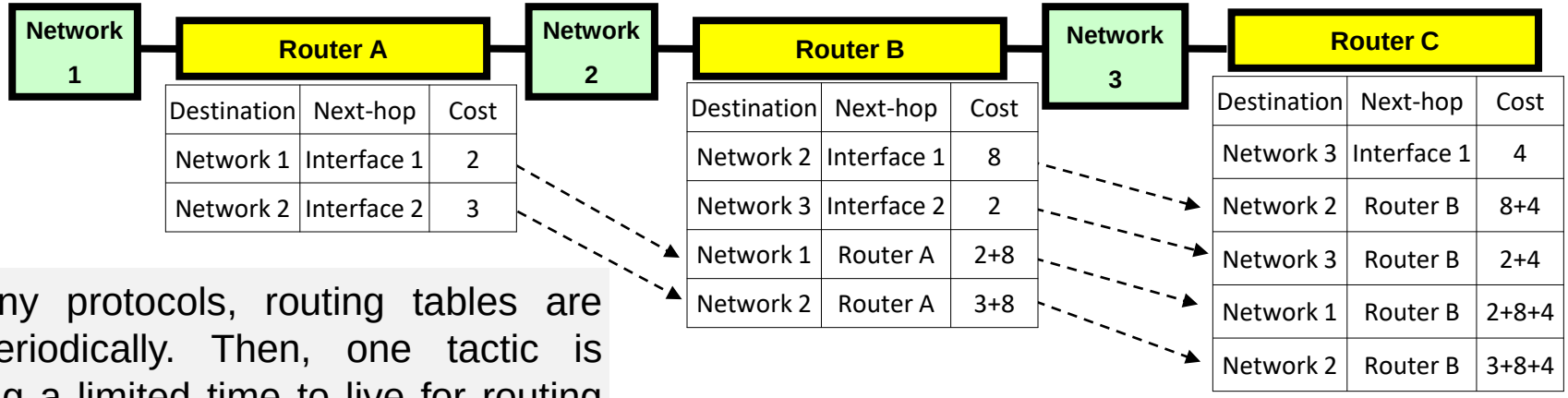
Distance-vector – Each router sends its own routing table to neighbour routers. Each router receives routing tables from neighbours and merges them with its own.

Link-state – Each router detects and monitors the neighbour routers, then sends to neighbours the list of routers it knows. Routers receive from neighbour routers, lists of known routers and connected networks, then merges each with its own. Using this list of routers and networks, each router builds on its own the routing table.

DISTANCE-VECTOR algorithms

The idea behind these algorithms is rather simple, as we get further apart from a destination the greater should be the cost at the routing table line for that destination. To achieve this, **each time a routing table line is received from a neighbour router an additional value is added to the cost**. This additional value can be one (then the metric becomes the number of hops) or a variable value representing the performance and dependent on the network interface.

One other thing routers do when they receive routing tables from neighbour routers is **setting the next-hop** in received lines **to the source address** from where the information came. The following diagram tries to illustrate how it works:



On many protocols, routing tables are sent periodically. Then, one tactic is enforcing a limited time to live for routing table lines incoming from neighbour routers. If a line isn't refreshed, it ends up removed.

Thanks to the routing protocol, Router C is aware of Network 1 and Network 2, and on how to reach them.

LINK-STATE concepts

Unlike with distance-vector, in link-state algorithms routing tables are not progressively constructed when the information travels along the paths. Now the goal is providing to every router full information about the infrastructure layout.

To achieve this, each router detects and monitors (link-state) neighbours routers (next-hops). The link-state list of next-hops is then transmitted to all neighbours routers. Each router, merges all received lists with the local list before retransmitting to neighbours.

Once a router acquires the full list, it will use a search algorithm (usually tree based) to find the shortest path (lowest cost) to reach each network, and thus, builds the routing table, all by itself.

Each router must actively detect and monitor neighbours routers. There is no need for cyclic announcements, whenever a change is detected (new neighbour or neighbour becoming unavailable), and only then, a link-state list is sent again.

Notice that, one router sending a changed list will have a chain effect throughout all routers. During a short period, networks will be flooded with link-state lists updates.

Autonomous systems (AS)

For the sake of efficiency and security, a routing protocol can't be implemented in an unlimited scale, it must be restrained to a zone of the infrastructure.

Let's imagine that was not so, and every router on the internet was using a routing protocol to communicate with all other routers and establish routing tables:

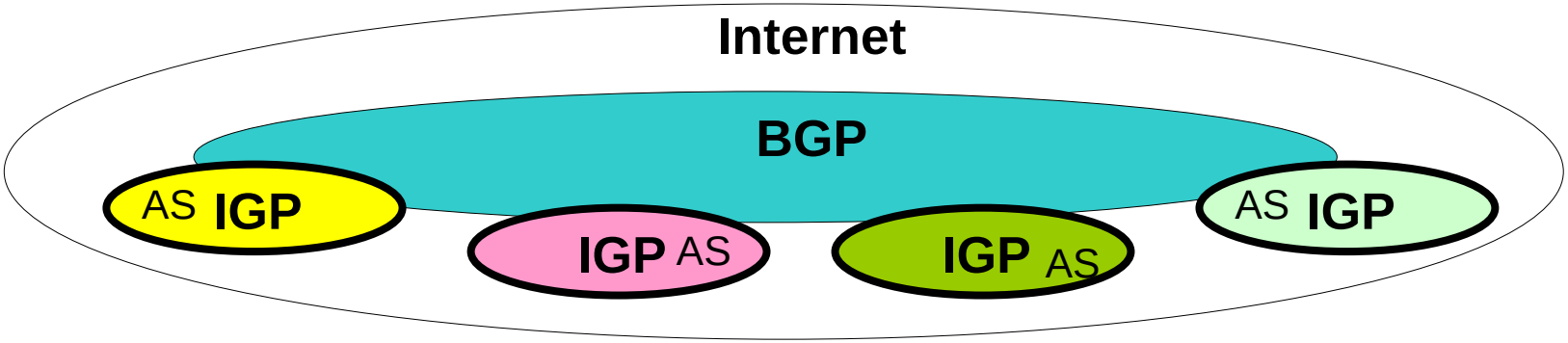
- Routing tables would be massive, with millions of lines and the amount of information to be managed would be huge.
- The traffic, due to the routing protocol itself, would be intolerable.
- The time it would take, for a change to be propagated everywhere, would be appalling.
- One routing error, introduced anywhere, would disturb the entire internet.

This issues are solved by, segmenting the infrastructure into routing independent zones called **autonomous systems**. Within each autonomous system, a routing protocol is used to build routing tables. Autonomous system limits are settled by **boundary routers**, these routers are configured for not forwarding routing protocols information.

Individual networks inside an autonomous system become an internal issue. Under the external point of view the entire autonomous system should be **addressable as a single address block**.

Interior Gateway Protocols (IGPs) and the Border Gateway Protocol (BGP)

Internet routing is managed by BGP, this protocol is used to settle routing between locally managed autonomous systems. To be able to interact with BGP, each autonomous system needs a **unique** Autonomous System Number (ASN), assigned by IANA (Internet Assigned Numbers Authority) or a delegate authority.



Each autonomous system has a unique ASN and some settled addresses blocks. Within the autonomous system, routing within the assigned address blocks is administratively autonomous and IGPs can be used for that purpose. Actually, BGP itself can also be used internally in a local AS but with private ASNs and not connected to the exterior BGP.

Furthermore, internally a local AS can be split into several autonomous systems, nevertheless that's irrelevant for the exterior BGP.

Interior Gateway Protocols (IGPs)

IGPs are routing protocols used within autonomous systems connected to the exterior BGP internet infrastructure. As mentioned before, BGP can also be used as an IGP, in that case a private ASN, ranging from 64512 to 65534, should be used.

Some significant IGP protocols

Distance-vector: RIP (Routing Information Protocol) and IGRP (Interior Gateway routing Protocol).

Link-state: OSPF (Open Shortest Path First) and IS-IS (Intermediate System to Intermediate System) .

Our attention will be focused on RIP and OSPF. We will also be paying some attention to the Cisco proprietary Enhanced Interior Gateway Routing Protocol (**EIGRP**), it's usually classified as hybrid because has features of both classes.

RIPv1 (Routing Information Protocol version one)

- RIP path metric is the number of routers to cross to reach the destination (hops), this is because the cost assigned to every interface is usually 1. The maximum path metric value is 15 hops, above this value the destination is unreachable.
- Each router broadcasts over UDP its routing table in all connected networks. The transmission is made approximately every 30 seconds.
- When a line in the routing table is not refreshed for 180 seconds it will be marked as unreachable by setting the metric to 16 hops.
- In RIP version one no network masks are transmitted, thus classful networks are assumed. RIPv1 can't be used with classless addressing (CIDR).
- When a router receives a line, it increments the metric (adds the interface's cost) and set the next-hop to the incoming source address. There is no AS concept nor AS numbers, thus, a router cannot be connected to two different RIP autonomous systems. RIP autonomous systems can exist, but there must be a non-RIP autonomous system between them.
- The protocol is unsafe, all information is accepted from neighbours without authentication.

RIPv2 (Routing Information Protocol version two)

Planned to be partially compatible with version one, yet, overcoming some notorious limitations of RIPv1.

- The path metric works the same as for version one, and it stays limited to a maximum of 15 hops.
- Each router sends its routing table by using multicast to address 224.0.0.9 instead of broadcast.
- The sent information, has now network masks, thus, supporting CIDR (Classless Inter-Domain Routing). Unlike RIPv1, RIPv2 can be used with classless addressing.
- Authentication is supported by using the MD5 algorithm together with a pre-shared secret key (PSK).

Although RIP is not link-state, in addition to the periodic sending, some implementations can force the immediate sending of updated tables once changes are detected. This enhances the convergence time for some cases. Convergence time is notably long in RIP, up to 180 seconds for each hop.

OSPF (Open Shortest Path First)

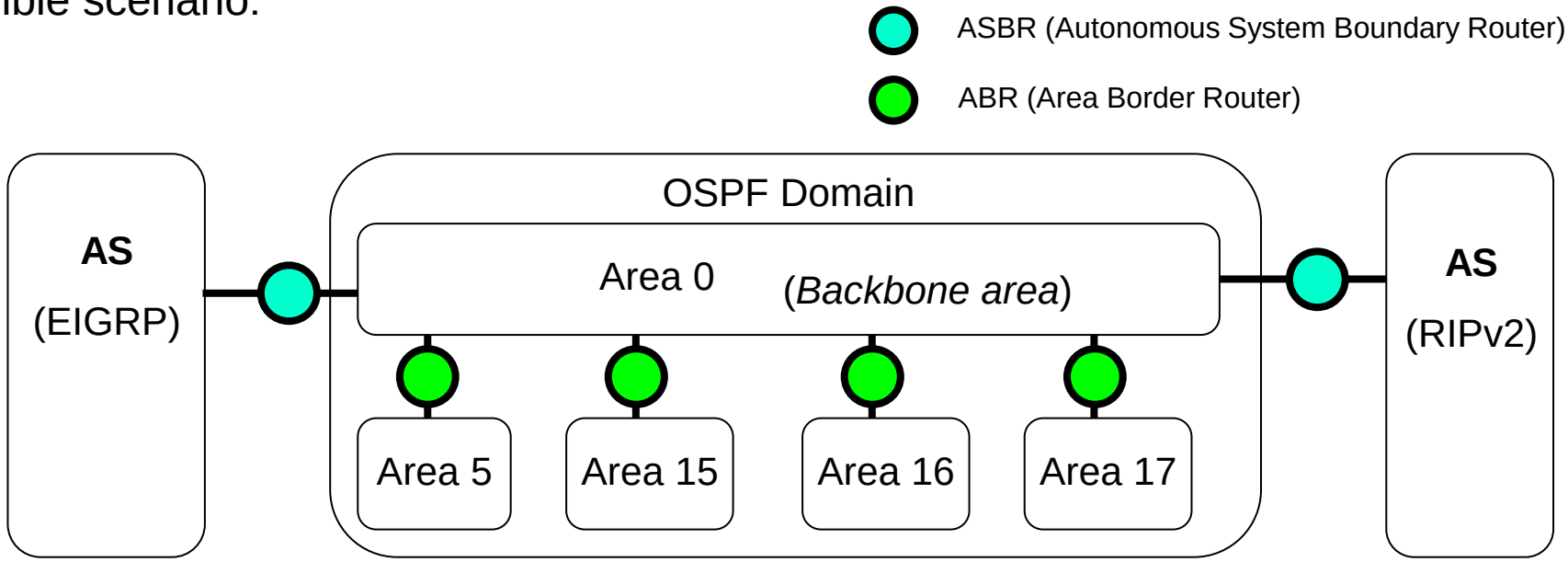
This is a link-state protocol, each router identifies neighbour routers by using broadcast and multicast to address 224.0.0.5. Later the neighbour's list (LSA - Link-State Advertisement) is propagated by sending it to the multicast address 224.0.0.6.

- Unlike RIP (and EIGRP we will see ahead), OSPF information is directly placed into IP packets, neither UDP or TCP are used.
- Each router constantly checks the availability of neighbours, if a change is detected a new LSA is sent, this will also trigger LSA sending by all other routers (LSA flood).
- The LSA information includes network masks, CIDR is supported by OSPF.
- The LSA also includes a link metric, usually the link transmission rate.
- With LSA information received, each router build the network layout tree, and then finds the best path (lowest path metric) to reach every network, this results in its own made routing table.
- OSPF path metric is assessed through received link metric values.
- MD5 and HMAC-SHA authentication is supported.

OSPF areas

OSPF defines a single autonomous system called OSPF domain. Though, the OSPF domain can be split into areas. OSPF areas are themselves fairly equivalent to autonomous systems, however, routing information is forwarded between them.

OSPF areas are identified by 32-bits numbers, **area zero** must be created first and is called the **backbone area**. Additional areas must be adjacent to area zero.
Routers interconnecting areas are called Area Border Routers (ABR) and routers interconnecting the OSPF domain to other autonomous systems are called Autonomous System Boundary Routers (ASBR). The diagram below presents one possible scenario:



EIGRP (Enhanced Interior Gateway Routing Protocol)

EIGRP is an improvement to IGRP by Cisco to overcome several issues. In essence it's a distance-vector protocol, but it also includes some link-state features.

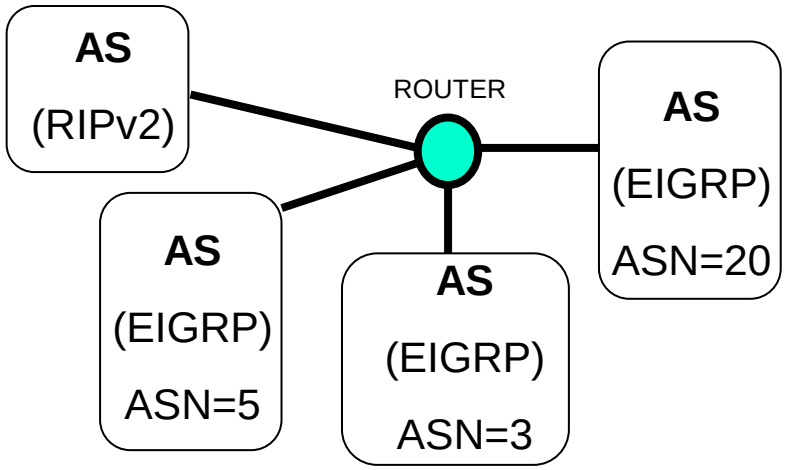
- Each router detects and checks the availability of neighbour routers by sending unicast/broadcast/multicast hello messages.
- Routing tables are propagated to neighbours by sending them to the multicast address 224.0.0.10, but this only happens when there's a change. Unlike with RIP, there's no systematic periodic sending of routing tables.
- Because neighbour routers are closely monitored, whenever there's a status change it is immediately reflected into routing tables and sent to neighbours. This ensures a short convergence time.
- EIGRP supports CIDR, even though, the default configuration on Cisco routers assumes classful addressing (auto-summary).
- The path metric is assessed by taking in account all used links status, each including the transmission rate, MTU value, network delay, reliability and load.
- The number of hops is not directly included in the path metric, however, by default, destinations taking more than 100 hops are accounted as unreachable, this limit may be adjusted by configuration to up to 224 hops.

EIGRP autonomous systems

EIGRP **always requires** the identification of the autonomous system by a number (ASN) from 1 to 65535. All information sent by EIGRP has an associated ASN and is ignored by EIGRP routers using a different ASN.

The use of autonomous system numbers by EIGRP, makes it possible for an autonomous system boundary router (ASBR) to be connected to several different and independent other EIGRP autonomous systems.

The diagram, on the right, illustrates such a scenario.



Notice, however, that EIGRP autonomous system numbers are not related to BGP autonomous system numbers.

Routing between autonomous systems

Autonomous systems are created to isolate network infrastructure zones under the point of view of routing tables management. The main advantages are:

- An easier administration (fewer networks).
- Administrative independence (routing changes are internal to the AS).
- Smaller (more efficient) routing tables.
- Less network traffic due to routing protocols (confined to the AS).

Nevertheless, data packets must be routed between networks whether networks are in the same autonomous system or not.

Because that's the way they are supposed to operate, autonomous systems boundary routers don´t transfer routing information about networks belonging to one autonomous system to other autonomous systems.

If no further routing configuration was provided to the autonomous system, communications would only be possible within the same autonomous system.

To enable routing between autonomous systems, **additional routing information must be inserted** into autonomous systems routing protocols.

Route redistribution

Routing tables, produced by any routing protocol within an autonomous system, only have data about networks inside the autonomous system. Hence, information about outside networks must be inserted, this is called **route redistribution**.

Static routes can be defined and then redistributed into an autonomous system. This is usually required for the default-route, and it can also be done for address blocks assigned to other autonomous systems. Any router in the autonomous system can be used for this purpose, once inserted it's propagated within the AS .

In boundary routers, something else can be done. Because they are connected to more than one autonomous system, it's possible to automate the redistribution of routes received from one autonomous system into another autonomous system. Also, while coping routing information between autonomous systems, several conditions and mangling may be enforced.

Either being static routes or received from another autonomous system, when they are redistributed into a protocol, an appropriate metric value must be settled.
If routing protocols are the same, the original metric value may be kept, otherwise, a new metric value must be established to meet the destination routing protocol metric format.