Redes de Computadores (RCOMP)

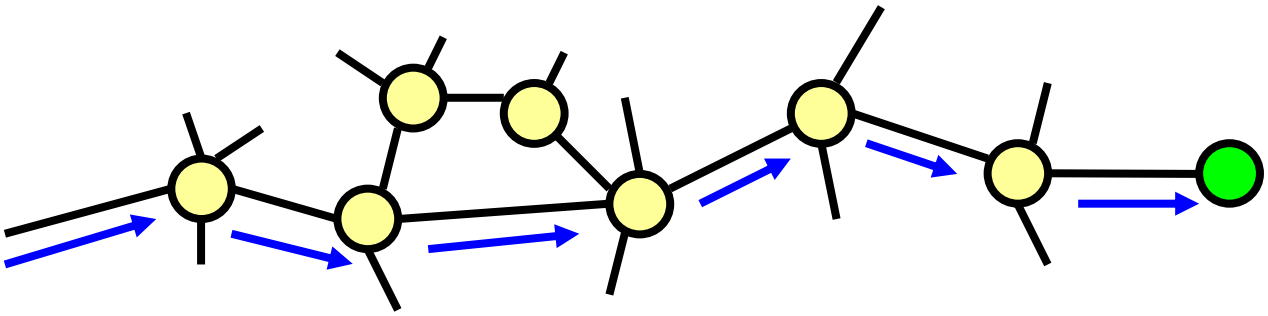
Lecture 06

2017/2018

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

Intermediate nodes

Intermediate nodes have a key role in any kind of switched network. In a switched network, 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, they forward layer two frames.

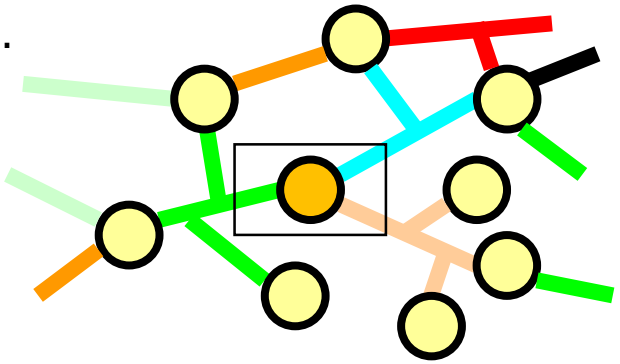
IPv4 packets (usually called datagrams) also reach their destination due to the effort of intermediate nodes, of course, operating at layer three. These layer three intermediate nodes are called routers or gateways.

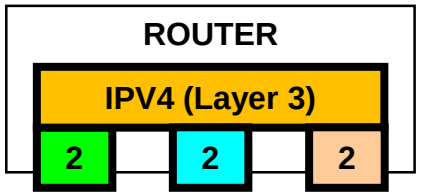
Routers (aka gateways)

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 implementations), possibly of different types. 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 (neighbour router).

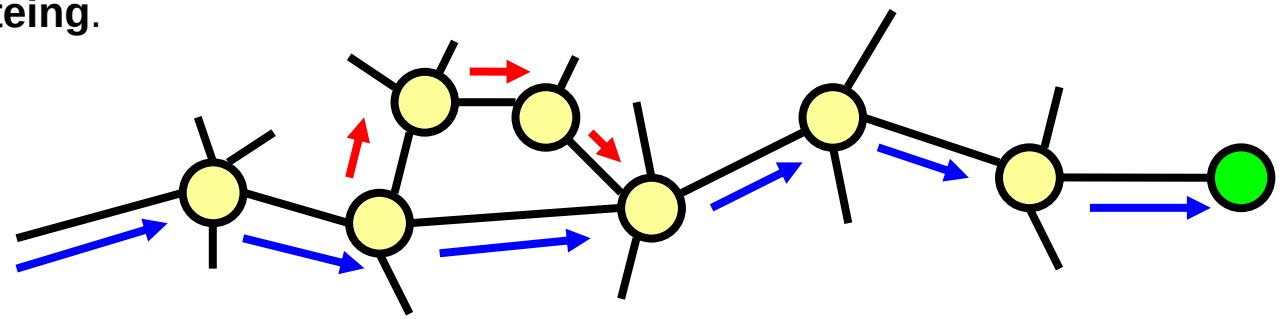




The above images represent one central router (inside a rectangle). It has three network interfaces and a total o eight neighbour routers (directly connected routers). For each packet this router receives, it will have to decide to which of the eight neighbours the packet should be forwarded. Neighbour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

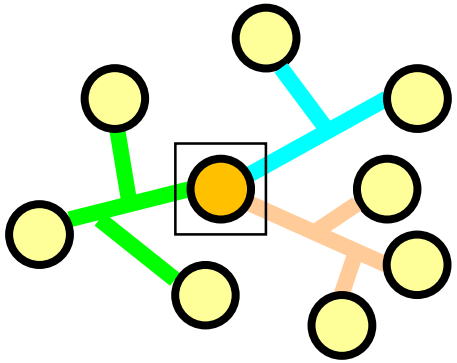
Routeing (or routing)

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



Each router makes its own local decision by picking the appropriate next-hop. For a datagram 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.

Routers interact only with neighbour routers, they are the only potential next-hops. On the left image, the central router has eight neighbour routers, so there are only eight different possible next-hops.



Routeing tables

IP packets sent anywhere 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.

Routeing decisions (picking the appropriate next-hop) are taken by using information present at the **routeing table**.

Routeing Table	
DESTINATION	NEXT-HOP

Each router has its own routeing table, they are made of a sequence of lines. Routeing 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.

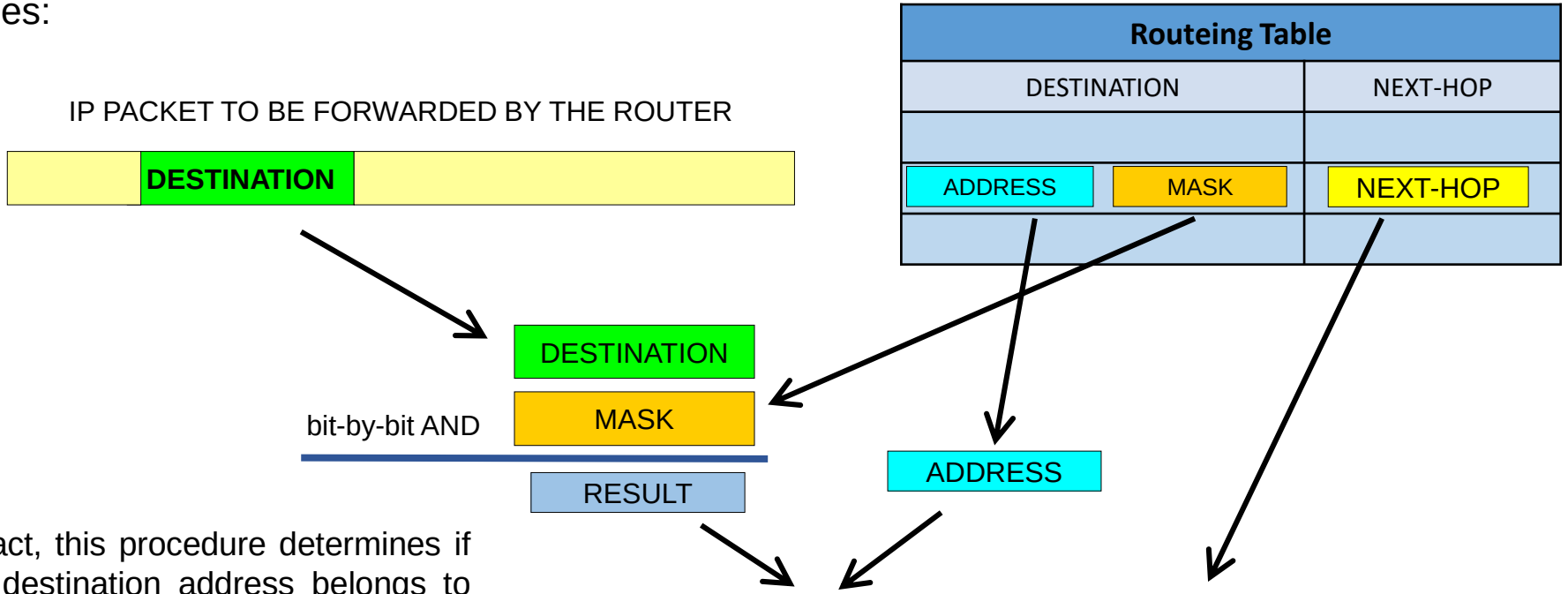
NEXT-HOP – to where packets should be forwarded to ultimately reach this DESTINATION. It's the IP address of a neighbour router, that router is the next intermediate node on the path to reach the intended DESTINATION.

Routeing and routeing tables

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

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

The following schema explains how routers match destination addresses with routeing table lines:



In fact, this procedure determines if the destination address belongs to the network/prefix at the routeing 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 routeing table, yet for those lines, the next-hop value it's a node´s network interface name instead of an IP address.

Routeing 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 does not 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**.

On the right, there's an example of a routeing 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.
Take note that any next-hop must always belong to a local network.

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 routeing configuration

Every IP node has a routeing table, the humblest configuration we can imagine is a node connected to a single isolated local network, this will result in a single line routeing table.

DESTINATION	NEXT-HOP
Local Network	Network Interface

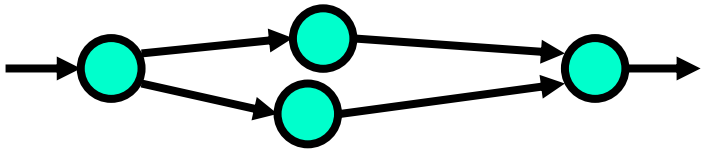
Most end nodes, like clients and servers, have only one network interface, they are, therefore, usually connected to only one local network. But unlike the above scenario, the local network is not isolated, there is a router on it providing access to other networks and ultimately to the internet. This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

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

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

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

Alternative paths



When there's only one possible path along the network, a two columns routeing tables is enough. This is not the case if there are 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' routeing 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 routeing table needs one additional column called COST or METRIC. It's a numerical value expressing how adverse to performance will be using that line. Of course, if there are several alternatives to reach the same destination, the router embraces to the one with lower cost.

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

Dynamic Routeing concepts

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

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

Even though that's a pleasant facet, the big motivation for dynamic routeing is not avoiding the burden of manual routeing tables creation.
Dynamic routeing 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 **reflected in the routeing tables**. This can't be achieved through static routeing.

The process of reflecting in routeing 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 routeing. This comes obvious, as, with static routeing, packets will always follow the same exact path.

Routeing Protocols

Routeing protocols are used by routers to inform neighbour routers about links availability and performance. **If every router transmits to neighbours what it knows, ultimately, every router will know everything about the whole infrastructure.** Don't forget routers make the infrastructure fabric because they interconnect networks.

Some details about routeing protocols operation can have some impact:

- Information may be always sent in broadcast/multicast or, once neighbour routers are detected, may be sent in unicast.
- Neighbour routers may be monitored (link-state).
- Information may be sent periodically or may be sent only when there is a status change.

There are two major classes of routeing protocols

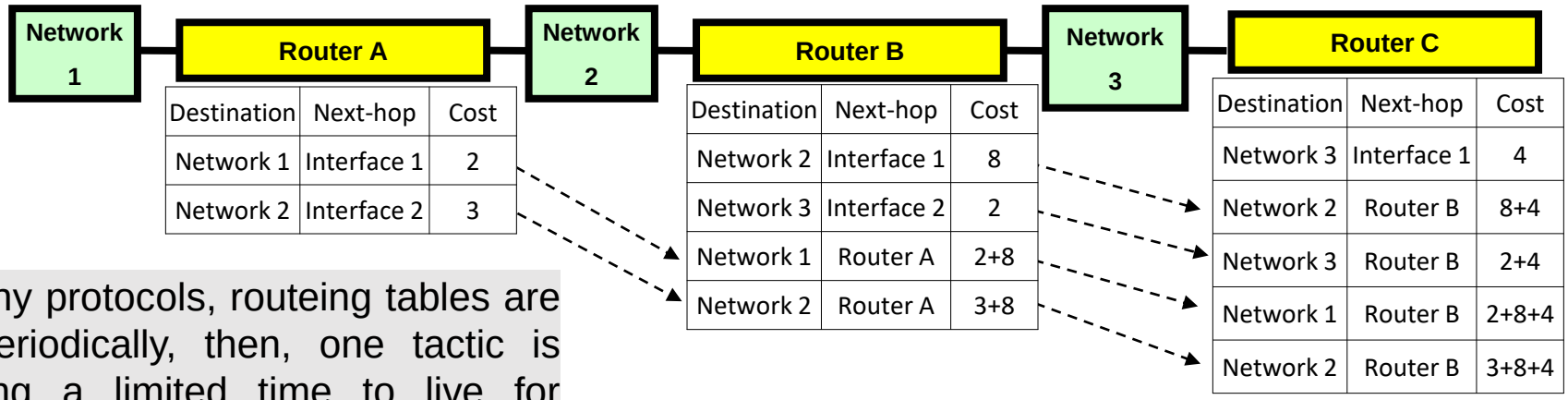
DISTANCE-VECTOR – Each router sends its own current routeing table to neighbour routers. Each router receives neighbours routeing tables and merges each with its own.

LINK-STATE – Each router detects and monitors neighbour routers. Sends to neighbours its list of known routers. Receives from neighbour routers, lists of known routers and merges each with its own. Using the information on this list, each router builds by itself a routeing 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 routeing table line for that destination. To achieve this, **each time a routeing 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 dependent on the network interface.

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



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

Thanks to the routeing protocol, Router C now knows about Network 1 and Network 2, and how to reach them.

LINK-STATE concepts

Unlike with distance-vector, in link-state algorithms, routeing tables are not progressively constructed as 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 each network, and thus, builds the routeing 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), 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 routeing protocol can't be implemented in a limitless scale, it must be restrained to an infrastructure zone.

Let's imagine if that was no so, and every router on the internet was using a routeing protocol to communicate with all other routers and establish routeing tables:

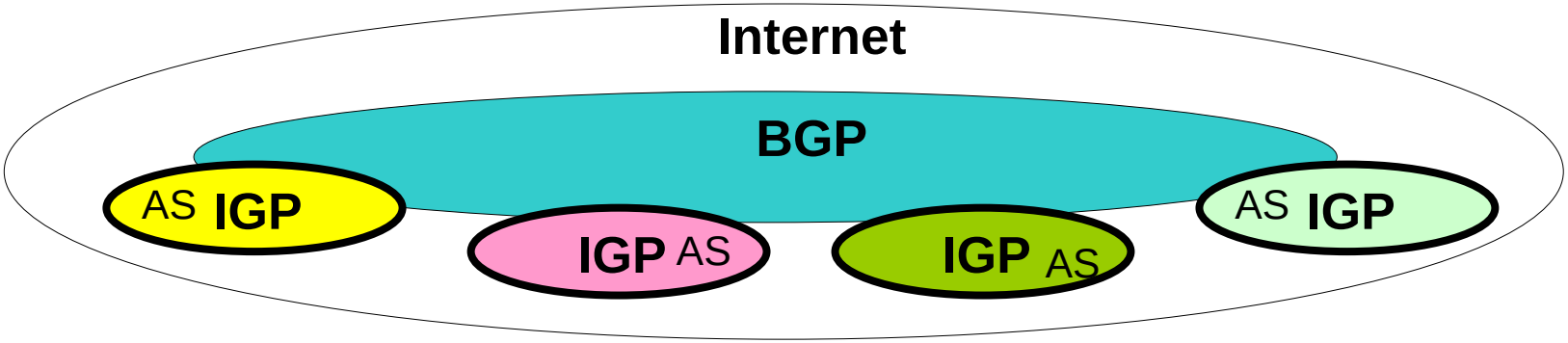
- Routeing tables would be massive, with millions of lines and the amount of information to be managed would be huge.
- The traffic, due to the routeing protocol itself, would be intolerable.
- The time it would take, for a change to be propagated everywhere, would be appalling.
- One routeing error,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

This issues are solved by, segmenting the infrastructure into routeing independent zones called **autonomous systems**. Within each autonomous system, a routeing protocol is used to build routeing tables. Autonomous system limits are settled by **boundary routers**, these routers are configured for not forwarding routeing protocol 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 routeing is managed by BGP, this protocol is used to settle routeing between locally managed autonomous systems. To be able to interact with BGP, each autonomous system need 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 address blocks. Within the autonomous system, routeing 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, however that is irrelevant to the exterior BGP.

Interior Gateway Protocols (IGPs)

IGPs are routeing 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, must be used.

Some significant IGP protocols

DISTANCE-VECTOR: **RIP** (Routeing Information Protocol) and IGRP (Interior Gateway routeing Protocol).

LINK-STATE: **OSPF** (Open Shortest Path First) and IS-IS (Intermediate System to Intermediate System) .

Our attention will be focused over RIP and OSPF. We will be also be paying some attention to the Cisco proprietary protocol Enhanced Interior Gateway routeing Protocol (EIGRP), it is usually classified as hybrid because has characteristics of both categories.

RIPv1 (Routing Information Protocol version 1)

- 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 metric value is 15 hops.
- Each router broadcasts over UDP its routeing table in all connected networks. The transmission is made approximately every 30 seconds.
- When a line in the routeing 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 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 another autonomous system between them using another routeing protocol.
- The protocol is unsafe, all information is accepted from neighbours without authentication.

RIPv2 (Routing Information Protocol version 2)

Planned to be partially compatible with version 1, yet, overcoming some notorious limitations of RIPv1.

- The metric works the same as for version one and it stays limited to a maximum of 15 hops.
- Each router sends its routeing table by using multicast to address 224.0.0.9 instead of broadcast.
- Sent information, now has network masks, thus, supporting CIDR (Classless Inter-Domain routeing). 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 addiction to the periodic sending, some implementations can force the immediate sending of updated tables once changes are detected. This enhances the convergence time in some cases. Convergence time is notably long in RIP, up to 180 seconds for each hop.

OSPF (Open Shortest Path First)

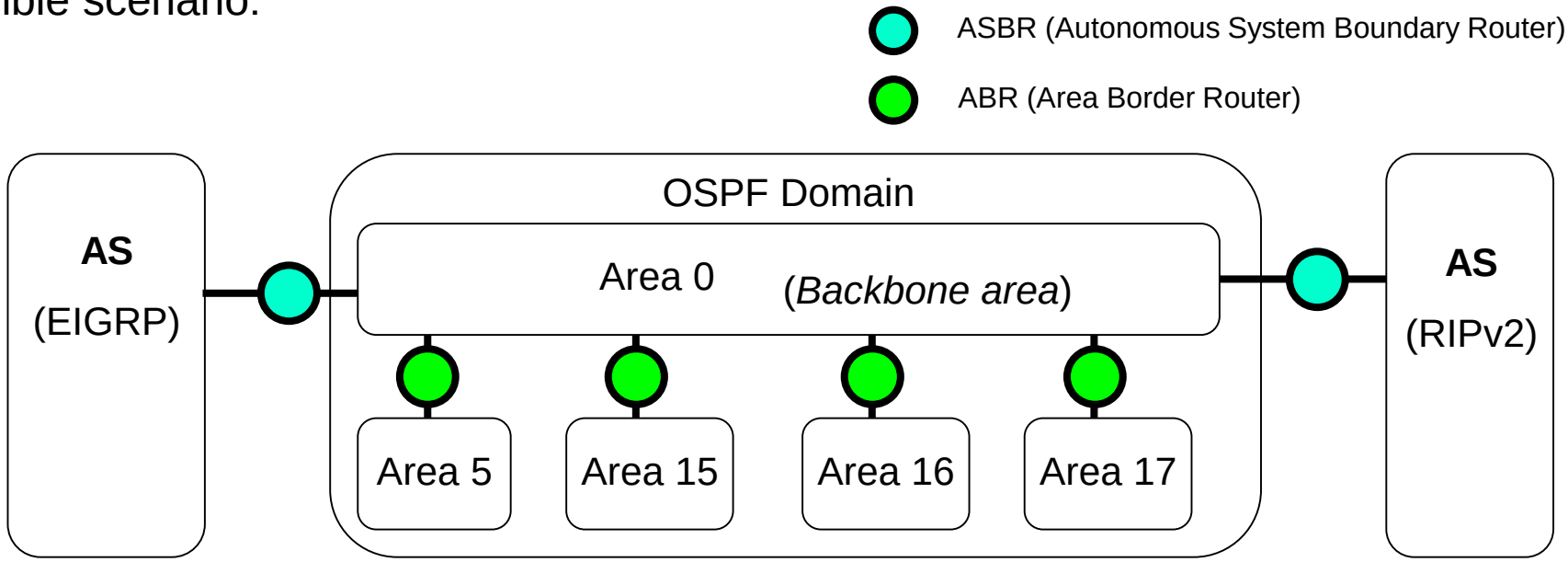
This is a link-state protocol, each router identifies neighbour routers by using broadcast and multicast to address 224.0.0.5. Later the neighbour list (LSA - Link-State Advertisement) is propagated by sending it to the multicast address 224.0.0.6.

- 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).
- LSA information includes network masks, CIDR is supported by OSPF.
- LSA information 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 metric) to reach every network, this results in its own made routeing 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, routeing information is forwarded between them.

OSPF areas are identified by 32-bits numbers, **area zer**o 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 a possible scenario:



EIGRP (Enhanced Interior Gateway Routeing 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 features from link-state.

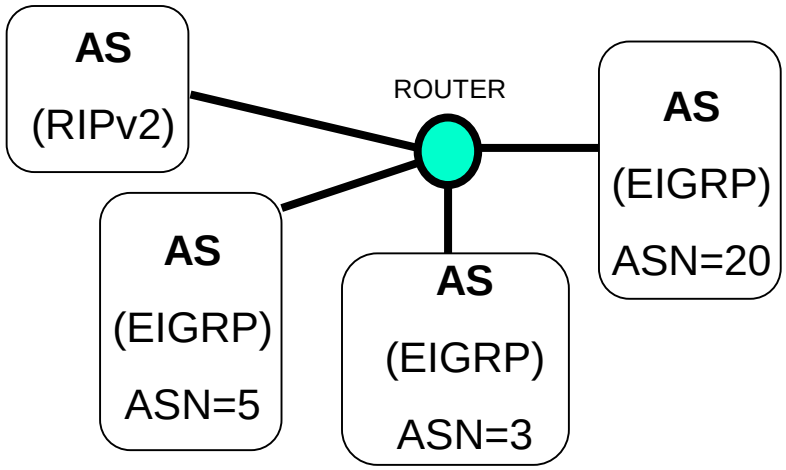
- Each router detects and checks the availability of neighbour routers by sending unicast/broadcast/multicast hello messages.
- routeing tables are propagated to neighbours by sending them to the multicast address 224.0.0.10, but this only happens when there's a change. Unlike with RIP, there's no systematic periodic sending.
- Because neighbour routers are closely monitored, whenever there is a status change it's immediately reflected into routeing tables and sent to neighbours. This ensures a short convergence time.
- EIGRP supports CIDR, even though, by default, it assumes classful addressing (auto-summary).
- The path metric is assessed by taking in account all used links status, 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 value may be adjusted up to 224 hops.

EIGRP autonomous systems

EIGRP **always requires** the identification of the autonomous system by a number (ASN) from one 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, EIGRP autonomous systems.

The sample diagram, on the right, illustrates that scenario.



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

Routeing between autonomous systems

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

- An easier administration (fewer networks).
- Administrative independence (routeing changes are internal to the AS).
- Smaller (more efficient) routeing tables.
- Less network traffic due to routeing 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 routeing information about networks belonging to one autonomous system to other autonomous systems.

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

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

Route redistribution

Routeing tables, produced by any routeing 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 can also be done for address blocks assigned to other autonomous systems. Any router in the autonomous system can be used for this purpose.

In boundary routers, something else can be done. Because they are connected to more than one autonomous system, its possible to automate the redistribution of routes received from one autonomous system into another autonomous system. Also, while coping routeing 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 routeing protocols are the same, the original metric value may be kept. Otherwise, a new metric value must be settled to meet the destination routeing protocol metric format.