

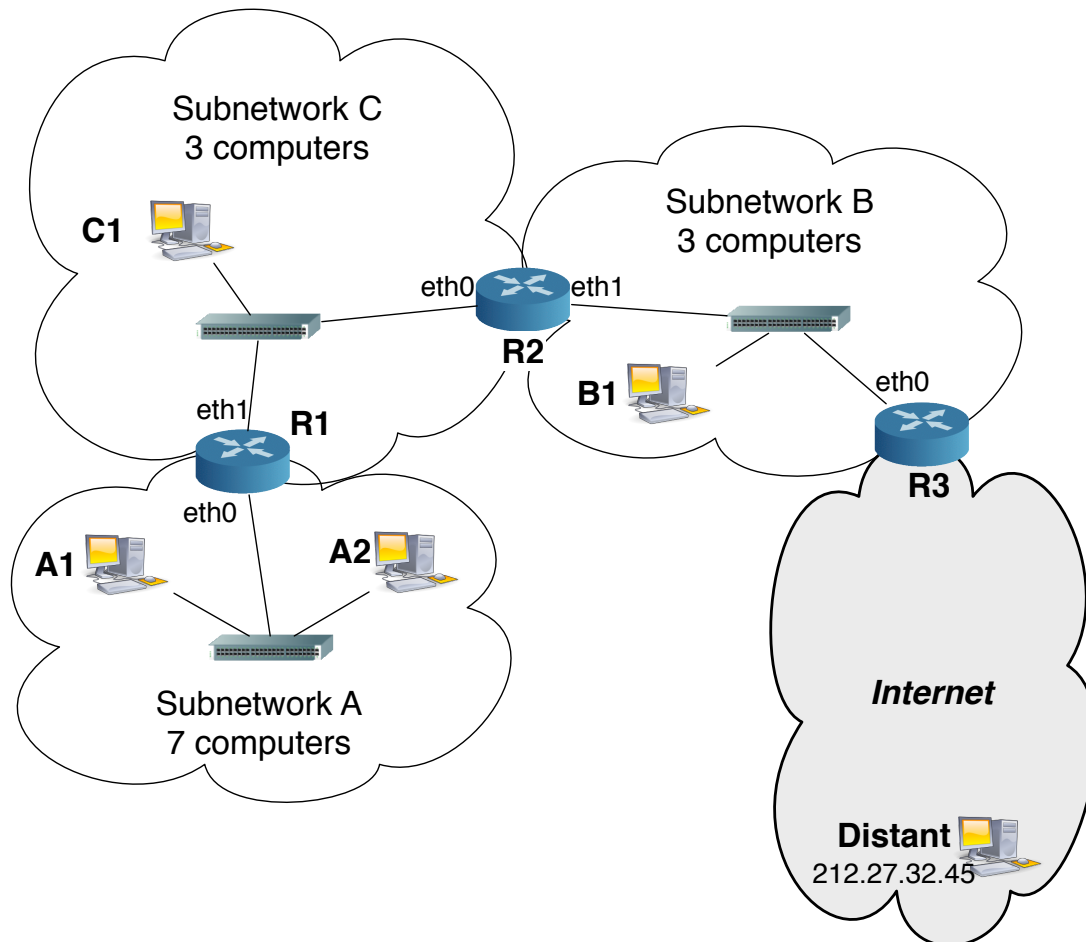
Lab 5 – Routing Protocols

RES 841

October 2013

1 Addressing plan - definition and implementation

You will start to launch the virtualized environment using VMWare player. Refer to the first lab for instructions and **change directory to TP_3a** before typing `lstart`. This will create the network represented on the figure below:



1. Your first duty is to build an addressing map, i.e. to separate the addressing space that will be allocated to you in non-overlapping sub-spaces. The network we are willing to configure is composed of three sub-networks, *A*, *B* and *C*, represented on the figure. You do not have to worry about the "Internet" part, or about the *Distant* computer.

To separate an addressing space into sub-spaces, you need to evaluate the number of machines that you can address in the global IP range and to split this range in parts. Each of these parts should be large enough for your sub-network and there should be no intersection between the different sub-spaces. You can only create spaces which have a size equal to 2^k machines (2, 4, 8, 16, 32, etc.). Therefore, you will probably need to create slices whose size is larger than the true number of machines that you want to address. You may also consider that:

- One of the addresses is reserved for the *network* address and cannot be allocated to any machine. It is the address whose interface identifier is composed only of 0s. (e.g. for 137.194.0.0 / 16, the network address is 137.194.0.0)
 - One of the addresses is reserved for the *broadcast* address and cannot be allocated to any machine. It is the address whose interface identifier is composed only of 1s. (e.g. for 137.194.0.0 / 16, the broadcast address is 137.194.255.255)
 - the network layer interconnection equipments (routers) need at least one IP address per active network interface. This address has to be part of the network to which the interface is connected (which means that a router is a machine that is part of several networks)
 - the link layer interconnection equipments (switches, bridges) do not need IP addresses. Some of them have one IP address, which is usually used for configuration.
2. From the previous information, what are the *minimal* sizes of the IP addresses spaces for all the sub-networks mentioned above? Translate these sizes in network masks and express the prefix length (i.e. prepare the CIDR notation).
 3. Consequently, what is the minimal size of the global IP addressing space?
 4. You were allocated the global IP addressing space 137.194.23.32/27 for your network. Establish the addressing space by dividing this space in sub-spaces for sub-networks A, B and C. Indicate, for each of these networks, its address, its network mask, the minimal and maximum address that can be allocated to a computer and the broadcast address. How many distinct configurations are possible?
 5. Once your addressing plan is defined, you will implement it on the lab machines, *including routers*. To configure an address and a sub-network on a machine, you will use the command `ifconfig ethX A.B.C.D netmask E.F.G.H`, where `ethX` is the interface identifier that you are configuring, `A.B.C.D` is the interface IP address and `E.F.G.H` is the subnet mask represented in decimal notation (e.g for a /25 network, the mask is 255.255.255.128).

Good practice consist in giving to the routers the "highest" addresses of a subnet and the lowest ones to the machines. However, nothing is mandatory.

6. From machine *A1*, try to emit a `ping` request towards *A2* (using its IP address, there is no DNS server in the virtual network). Try now to reach machine *B1*. What happens?
7. Your network is not yet functional, as routers need to be configured. Indeed, in this configuration, the routers in this lab do not run any routing tables update protocol and cannot self-configure. You need to configure the routers manually, entering all the entries in the routing tables.

To display a machine routing table, type `route -n`. This command will display an array with several columns, among which:

- *Destination* gives the address of the network that this entry describes.
- *Genmask* indicates the mask of the network that this entry describes.
- *Gateway* gives the address of the next router to reach the network described by the two previous entries.

From this information, explain why *A1* could not join *B1* in the previous question.

8. By default, the routers are configured well enough to allow a few local communications. Terminals can reach machines that belong to their own sub-network, but do not know how to get out of this sub-network. They need a *default* route that indicates, when no other rule applies, to which gateway a packet should be transmitted. To configure the default route, you need to type in a machine's terminal `route add default gw A.B.C.D`, where `A.B.C.D` is the IP address of the default gateway. Configure the different terminals gateways (*R1* for *C*, *R2* for *A*, *R3* for *B*).
9. Invoke `tcpdump -e -t` on *C1*. Try to emit a `ping` request from *A1* to *C1*. What can you notice? How do you interpret this result?
10. Configuring default routes is not enough. In this case, every packet that is destined to a sub-network that does not correspond to the emitter's one will be directed to the default gateway. Therefore, the answer to the `ping` request will be sent to *R2* that will not know where to redirect it. You need to configure the different routers to allow communication between sub-networks.
 To add such an entry in a router's table, you will need to use the following command on the router: `route add -net A.B.C.D netmask E.F.G.H gw I.J.K.L dev ethX`, where `A.B.C.D` is the targetted network address, `E.F.G.H` its mask, `I.J.K.L` the address of the next router on the path, and `ethX` the interface on which `I.J.K.L` can be reached.
 Configure all the routes that allow to reach *C* in routers *R2* and *R3*, and the route to reach *A* in router *R3*. The other routes are already configured automatically with the interface.
11. Display *R2* routing table. Distinguish the entries that correspond to each of the networks to which this router is directly connected, the entries that designate distant networks, and the default route. Note that a `0.0.0.0` often designates the default route. Explain why this `0.0.0.0` address is compatible with the longest prefix match paradigm.

2 Topology Discovery

You will now manipulate a network that corresponds to the Telecom ParisTech network (AS 1712), in which 9 routers are deployed and two terminals (ENST and INFRES) are connected. This autonomous system is connected to two other ASes, Cogent and Renater. The OSPF routing protocol is deployed within this AS and BGP is active on border routers (Avoranfix and Goudurix).

Start by launching the virtualized environment, change directory to `TP_3b` and launch the virtual network. You will get several windows, on the 9 school routers, the 3 external routers and the two stations.

1. When the network start, no routing protocol is launched. Therefore no routes are available. Using `ifconfig` and `ping`, draw the network topology, identify sub-networks and their masks.
2. If you try to send a `ping` from Avoranfix to Lug, what happens? Examine the routes of all routers and explain.

You can now stop the virtual machines and clean temporary files (`lclean`).

3 Dynamic Routing within an AS: OSPF

1. Change directory to `TP_3c` and launch the new virtual network. The network is the same except that OSPF is now active. Some terminal windows will not open to reduce clutter, but all machines are present on the network.
2. Try again the previous `ping` request. What happens? Look at the various routing tables and explain OSPF role.
3. Send a `traceroute` request from Avoranfix to Lug. Identify the router(s) that are crossed on this path. Draw this route on your topology.

4. Capture the traffic on any router with `tcpdump`. Open the capture file with `wireshark` and examine the OSPF *Hello* messages. What is the destination address of these messages? Does it correspond to a known router?
 It is indeed a particular address for one-to-many communication address: a *multicast* address. These addresses allow to reach a set of nodes (we talk about a multicast group). Here, the multicast group is called *all OSPF routers* and is equal to 224.0.0.5. Routers that receive a message towards this address will make sure to reach all the relevant destination nodes but will try to minimize the number of duplications of this message.
5. Examine now the payload of the *OSPF Hello* packets. You will notice that there is a certain number of fields that are irrelevant to you at this stage. Look closely at the `Hello_interval` and `Dead_Interval` fields. What are, in your opinion, these two values? What can you conclude about the convergence time, about the reactivity of the protocol, etc. ?
6. Try to find a message emitted by the *Avoranfix* router. Look in the *Hello* part the various neighbors that are declared. Compare with the topology that you determined in the first exercise. What can you deduce?
7. Capture traffic on *Avoranfix* using `tcpdump` in background (`&` suffix after the command). Then, send a `ping` from *Avoranfix* to *Lug*. On *Avoranfix*, again, type the command `mtr A.B.C.D`, where *A.B.C.D* is *Lug*'s IP address. The tool (`mtr`) displays the same information as `traceroute` but constantly refreshes the successive results. Now switch down the router by where the `ping` traffic passes by typing in the host window `!halt nomRouteur` where `nomRouteur` is the name of the router to switch off (case sensitive, be careful) . Look at the `mtr` display ; stop the `tcpdump` capture (`fg` then `Control-C`).
8. Look at the capture trace with `wireshark`, look at ICMP packets that are not `echo` and note their date. Examine then the OSPF traffic close to these packets. Without getting too much into details, what can you deduce on OSPF behavior? What is the order of magnitude of the convergence time of the protocol? Is there a relationship with the `hello_interval` and `dead_interval`? Explain.
9. Re-launch `tcpdump` then `mtr` before restarting the previously swithced down router (`!start routerName`). Analyze what happens in this scenario, similarly to previous questions. Why, in your opinion, does the route change while a working route exists?
10. Imagine how the scenario could have been handled in the case of a static routing scheme? What is the interest of a routing protocol?

4 Dynamic Routing outside an AS with BGP

1. Look at the *Toutatis* routing table. The school disposes of the address space `137.194.0.0 / 16`. You will notice that there are other prefixes in the routing table. How is *Toutatis* aware of these prefixes, if OSPF only works inside an AS?
2. Compare now *Lug* and *Cogent* routing tables. Look at the number of entries that correspond to the school network. Explain the variance in details level.
3. The BGP protocol is used between AS. On the machine called `transit`, you can display BPP tables with the `vttysh` command. This command will open a new command line, looking like the one of a true router, with a limited set of commands. You can find the available commands list on the `vttysh` man page, or on the Web.
 The `show ip bgp` command gives you a view of a router BGP table (AS-level routing table). Examine the table on `transit` and find the path that leads to the school network. What do you notice? Classify the AS 60000 (is it a *stub* AS, a *multihomed* AS or a *transit* AS)?