Politecnico di Torino

Exercises on Static IP Routing and Route Aggregation

Fulvio Risso

February 23, 2018

License

Acknowledgments

The author would like to thank all the persons that contributed to those exercises. Particularly, special thanks go to Flavio Marinone, Guido Marchetto, Santo Vario, Dario Orfeo and Jon Brenas.

Contents

I. Methodology **5**

1. Static routing and routes aggregation **6**

1.1. Main concepts . . . 6
1.1.1. Routing table . . . 6
1.1.2. Directly connected IP networks . . . 7
1.1.3. Remote IP networks . . . 8
1.1.4. Costs . . . 8
1.1.5. Aggregated routes . . . 10
1.1.6. Most specific routes . . . 11
1.1.7. Default route . . . 13
1.1.8. Different costs routes . . . 14
1.2. Deriving the routing table: methodology . . . 15
1.2.1. List and type of the IP networks . . . 16
1.2.2. Derive the spanning tree . . . 16
1.2.3. Determine the routing table . . . 17
1.2.4. Identify possible route aggregations . . . 18

II. Exercises **20**

2. Exercises **21**

2.1. Exercise 1 . . . 21
2.2. Exercise 2 . . . 21
2.3. Exercise 3 . . . 22
2.4. Exercise 4 . . . 22
2.5. Exercise 5 . . . 23
2.6. Exercise 6 . . . 24
2.7. Exercise 7 . . . 25
2.8. Exercise 8 . . . 26
2.9. Exercise 9 . . . 27
2.10. Exercise 10 . . . 28
2.11. Exercise 11 . . . 29
2.12. Exercise 12 . . . 30
2.13. Exercise 13 . . . 31
2.14. Exercise 14 . . . 32

III. Solutions 33

3. Solutions 34
3.1. Exercise 1 . . . 34
3.2. Exercise 2 . . . 35
3.3. Exercise 3 . . . 36
3.3.1. Routes with equivalent addressing space . . . 36
3.3.2. Routes with maximal aggregation . . . 37
3.4. Exercise 4 . . . 38
3.4.1. Routes with equivalent addressing space . . . 38
3.4.2. Routes with maximal aggregation . . . 38
3.5. Exercise 5 . . . 39
3.5.1. Routes with equivalent addressing space . . . 39
3.5.2. Routes with maximal aggregation . . . 39
3.6. Exercise 6 . . . 40
3.7. Exercise 7 . . . 41
3.7.1. Addressing done trying to maximize routes aggregation on R1 . . . 41
3.7.2. Addressing done trying to minimize the number of allocated addresses . . . 42
3.8. Exercise 8 . . . 43
3.9. Exercise 9 . . . 45
3.10. Exercise 10 . . . 47
3.11. Exercise 11 . . . 50
3.12. Exercise 12 . . . 55
3.13. Exercise 13 . . . 58
3.14. Exercise 14 . . . 60
3.14.1. Case 1 . . . 60
3.14.2. Case 2 . . . 60
3.14.3. Case 3 . . . 60
3.14.4. LAN 1 implemented as *switched LAN* . . . 61

Part I.

Methodology

1. Static routing and routes aggregation

The overall objective of this set of exercises consists in the capability to define and to manage the routing table of an IP network. For being able to manage the IP routing table, we suppose that the network has already been configured with the proper IP addressing plan and therefore that the various entities present in the network (hosts, routers) have been correctly configured with the proper IP addresses and netmask.

The ideas that are behind those exercises do not depend on the technology used to calculate the routing (i.e., static vs. dynamic) because, given a network topology and the costs associated with the network links, all algorithms converge to the same result. The difference between static and dynamic algorithms reside in the fact that the static routing has to be configured completely by hand in all the routers of the network and is not able to adapt dynamically to the variations (in terms of topology and costs) of the network. As a consequence, the concepts presented here are perfectly general and independent from the way the routing is computed.

To help the students to solve the exercises present in this set, we present first the main concepts that are behind the definition of the routing tables and route summarization, followed by a possible methodology that can be used to solve those exercises.

1.1. Main concepts

1.1.1. Routing table

The routing table is a table locally associated to a router whose objective is to keep the list of the *destinations* present in the given topology and the next hop that has to be used when forwarding a packet to that destination. In the IP world, the destinations are *all* the existing *IP networks* in a given topology[1]. For instance, in the topology of the figure below, five IP networks are present, each one corresponding to a single line in the routing table.

We can note that if the number of lines in the routing table depends on the number of IP networks present in a given topology, the size of that table will be the same for all the routers belonging to that topology. Obviously, since each destination corresponds to a single IP network, all the routers will have the same list of reachable destination, even though the *paths* toward them could be different.

For each destination in the routing table, the following information is usually present:

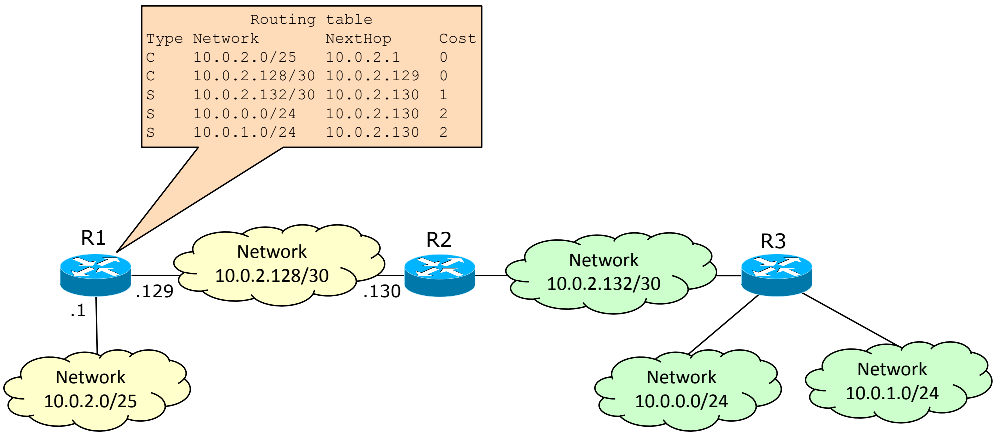
- **Route type**: shows how the router "learned" this network. Adopting in this set of exercises the convention used by Cisco routers, the networks directly connected are marked with the letter 'C', the routes known through the static routing are marked with the letter 'S', and more[2].

[1] This definition will be in fact modified later, as it is clear that such a table would become infeasible in the current Internet where we have millions of different destinations. However, for the time being we ask the student to stay with this definition, which associates each destination with a row in the routing table

[2] Other letters are used for dynamic protocols, e.g., 'R' identifies routes learned through RIP, and more. However, for the purpose of those exercises, 'C' and 'S' are enough.

- **Network/Netmask**: shows the network address (i.e., the *destination* of this route) and the associated netmask. Please note that the couple network address/netmask is, in fact, an address range.
- **Next Hop**: IP address of the interface of the *next hop* router that will be used to forward packets to the destination. The meaning of this field is different when the IP network is directly connected, and it will be detailed more in the following, when the difference between connected routes and the other will be presented (Sections 1.1.2 and 1.1.3).
- **Cost**: shows the distance from the current router and the destination network and it consists in a numeric value. For instance, a network reachable with cost 2 is much closer to the current router than a network reachable at cost 4. More details about costs will be given in the Section 1.1.4.

A possible example of a routing table is shown in the figure below.



While apparently directly connected routes (i.e., ‘C’ routes) and the others (e.g., ‘S’ routes) are equivalent in the routing table (with just the above letter that differentiates the two types), in practice connected routes are very different in the way they are *obtained* compared to all the other types of routes. Other differences are related also to the meaning of some *fields* in the routing table, which are slightly different in connected routes compared to all the other types.

1.1.2. Directly connected IP networks

Directly connected IP networks are the ones reachable with *direct IP routing*. For example, networks 10.0.2.0/25 and 10.0.2.128/30 for the router R1 in the previous topology are directly connected and are shown in yellow in the previous figure.

It is worthy noting that the directly connected networks are not the ones that are *physically* connected to the current router (e.g., all the IP networks that reside on a LAN connected to the router), but the *IP networks* reachable with direct routing. In fact, we can have an IP network configured on a LAN directly attached to the router, but that LIN may not be directly reachable from the router. In fact, if the router does not have any IP address belonging to that address space configured on its LAN interface, that network appears to be remote for the router. However, we should also mention that in the vast majority of cases the IP networks that are on *physically* connected LANs appears to be directly connected routes.

In a router, the knowledge of the directly connected networks is automatic and is determined by the fact that the router has an interface belonging to that IP network. For instance, the router R1 depicted in the previous network will automatically insert the networks 10.0.2.0/25 and 10.0.2.128/30 in its routing table as *connected* entries, without any intervention of the administrator of the router and even without dynamic routing.

In case of directly connected IP networks, the value of *next hop* field in the routing table identifies the address of the interface of the *current* router that will be used to reach this destination. For example, in the network of the figure the router R1 will reach reach all destinations 10.0.2.0/25 through its interface with address 10.0.2.1/25; hence the value of the field *next hop* will be `10.0.2.1`.

1.1.3. Remote IP networks

Remote IP networks are the ones that are not directly connected to the current router, hence are reachable through indirect forwarding. In case of indirect forwarding, packets directed to those destinations must be sent to a router that will forward them to this destination, either through direct forwarding or by sending them to another intermediate router. For example, networks 10.0.0.0/24, 10.0.1.0/24 and 10.0.2.132/30 (shown in green in the previous figure) represent remote networks for the router R1. A packet directed to host 10.0.1.1 will be handled with indirect forwarding by router R1 and forwarded to R2. R2 will use again indiret forwarding and send the packet to R3. Finally, R3 will use direct forwarding to send the data to the final host.

Differently from directly connected routes, remote IP networks are not automatically known by the routers; either some dynamic routing protocols or static routing has to be used to let the router know those networks. The former method is able to discover the LIN present in the topology and configure the proper routes automatically, while static routing has to be configured by hand by the network administrator. Once one of those two methods have been chosen (and properly configured on all routers), remote destinations will appear in the routing table. For instance, the previous figure shows remote networks with the letter 'S' in the routing table because they are defined using static routing.

For these networks, the value of *next hop* in the routing table identifies the address of the interface of the *next* router that will be used to reach this destination. For example, in the network of the figure the router R1 will reach all the destinations 10.0.0.0/24 through the left interface of the router R2, that has address 10.0.2.130/30: the value of the *next hop* field of this route will then be `10.0.2.130`.

The reason why the *next hop* is the interface of the next router and not the exiting interface of the current router is due to the fact that the two values would be equivalent only in the case of point-to-point links. For instance, in the figure it is obvious that a packet exiting from the interface `.129` of R1 can only be received by the interface `.130` of R2. However, this is not true in case of broadcast networks (e.g., an Ethernet LAN): for instance, a packet exiting from an Ethernet interface of a router can reach all the routers that are attached to the same LAN and whose IP address belong to the same IP network. It should be evident that the exiting interface is not always enough to determine the *next hop* for that destination, while the IP address of the next hop router along the path does not present ambiguities.

Finally notice that a router must reach its next hop **always** through direct forwarding, i.e. the next hop address must belong to one of the IP networks of the current router. If the address of the next hop results in a different IP network than the current router, that address results unreachable.

1.1.4. Costs

The cost of a route is required to choose a path that has a better (minor) cost instead of another path that appears less convenient (larger cost). For instance, the cost of a path (i.e., the cost for reaching a given IP network) is the sum of the costs associated to all the links of that path.

The cost is present in the routing table mainly for debugging, because it is not used by the router during the forwarding process. For instance, a router can be aware that multiple paths (i.e., multiple routes) with different costs exist toward a given destination, but the routing table shows only the path (i.e., the route) with the best cost. Therefore each route in the routing table represents the *best path* toward the destination, while any alternative route (with higher costs) may be kept in memory but are not shown in the routing table.

The cost of a route depends on many parameters, including the operating system of the device itself. For example Microsoft Windows assigns costs > 0 to both connected and static routes; others, as Cisco IOS, assume that both static and connected routes have cost 0. In addition, some of them (e.g. Windows) allow a single number as a cost, while others (e.g. Cisco IOS) define the cost as a couple *administrative distance/metric*. In this case the first number gives the *goodness* of the protocol that learned that route (for instance, a static route is usually considered better that a dynamic one) and the second one is the actual cost, knowing that the value of the administrative distance has the precedence over value of the metric, which means, for example, that a route with cost 110/1 will always be worse than another route with cost 1/12, while a route with cost 1/10 is better than a route with cost 1/12.

In current network devices, the cost for connected and static routes is fixed and well-known, although some minor differences can be found in different router manufacturers. In general, the cost for connected routes cannot be changed even by the network administrator; instead, the cost of static routes can be changed manually by setting the cost each route to the desired value[3]. The reason for this conventional cost for all the static routes is due to the impossibility, for the router, to know its actual distance from the destination network, as the router does not have any dynamic routing protocol on board and hence it has no idea about the topology of the network.

In this set of exercise, we will abstract from any particular router and the following convention will be adopted:

- the cost is a single number (not a couple *administrative distance/cost*)
- when not specified otherwise, connected routes have cost equal to 0 (that cannot be modified) while static routes have a cost that depends on the actual distance between the current router and the destination IP network, obtained summing the costs of traversing each link (and set conventionally to 1) till the destination[4]. For instance, the previous picture shows some static routes at cost 1 and others of cost 2. Anyway, the cost may be changed if the network administrator has some specific needs.

When moving to the real world, the student is required to check the convention used on real devices with respect to the cost of the static/connected routes and proceed to the needed adaptations from the theory presented here.

[3] In fact, Cisco IOS allows to change only the administrative distance but not the actual metric of static routes. In this case, it is possible to define a static route with a cost such as it is chosen only when the dynamic routing protocol cannot find any route to destination. This rather particular configuration is called *floating static route*, and leaves the precedence to the dynamic protocol (over static routes) when possible.

[4] The reason why we choose to assign different costs to static routes instead of configuring them all with the same (conventional) cost is because this should help the student to undestand routing. For instance, routing would be more intuitive if the cost is a function of the path.

1.1.5. Aggregated routes

The IP routing model foresees that two or more routes can be replaced by an aggregated equivalent route. The basic idea is that if a destination D_1 is reached through a given next hop NH_1 and a destination D_2 is reached through the same next hop NH_1, the destinations D_1 and D_2 can be "merged" together into an equivalent destination $D_{1,2}$.

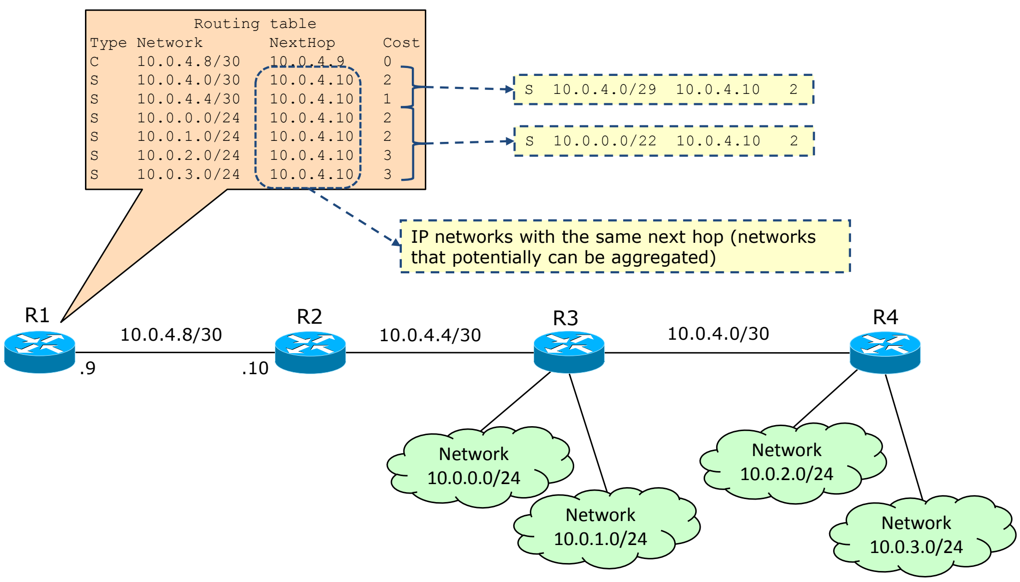
The advantage of aggregation is that the number of routes in the routing table decreases, hence the task of forwarding IP packets to the destination will become easier for the router.

There are two conditions for having the possibility to melt together two (or more) routes and replace them with an equivalent aggregated route:

- (*Mandatory*) the considered routes must share the same *next hop*;
- (*partially mandatory*) the considered routes must be aggregatable, which means that it must exist an address range that includes *exactly* the network addresses of the original routes[5].

The next figure contains an example of aggregation: the six green networks share the same next hop for the router R1 and thus may be potentially aggregated. Nevertheless it is important that the second condition also is respected: the new addressing space must be equivalent to the original ones. The second condition results in the impossibility to aggregate all the six networks with the same address range; the best we can do is to aggregate the two point-to-point networks with an equivalent route to the network 10.0.4.0/29, while the four /24 networks may be aggregated together in the address range 10.0.0.0/22.

Notice that the aggregation of the network 10.0.4.0/29 has been made possible by the way IP networks have been assigned in our topology: if the networks had been assigned in increasing order from left to right (10.0.4.0/30 between R1 and R2, 10.0.4.4/30 between R2 and R3, 10.0.4.8/30 between R3 and R4), it would not have been possible to aggregate them. Actually, remote networks 4.4/30 and 4.8/30 cannot be aggregated is a single addressing /29 range, as the valid ranges are 4.0/29 and 4.8/29, which do not include respectively the network 4.8/30 and 4.4/30. As a consequence, the way IP networks are assigned is the main key to allow (or prevent) the aggregation.



[5]It may be possible also to aggregate routes when the resulting address range is larger than the original address ranges, i.e., it does not include *exactly* the original routes. However, for the time being, let us consider only exact aggregations.

It is worthy noting that the value of the cost is not important when aggregating routes, hence we can aggregate routes with different costs. In fact, the router does not use the cost when it forwards packets. The router used the cost in the preceding routing phase, which consist in the selection of the best path among the many routes that exist toward the to the given destination. This is the reason why it is possible to aggregate routes with different costs; in this case the aggregated route will be configured with a "conventional" cost chosen by the operator.

The fact that the concept of aggregation changes the semantic of the information included in each route is important. While the original definition of the routes (*a router needs a specific route for each destination present in the topology*) assumes a that each route corresponds to an IP network, now *each route is associated to an address range.* In other words, the couple *network/netmask* that defines a route does no longer identify an IP network but it may identify a set of aggregated IP networks, i.e., an *address range.* As a consequence, an address that in the original network was not available for an host (for instance because it represents the network or the broadcast address), it appears as a normal host address when we consider the aggregated address ranges. This can be seen for example with the address 10.0.1.255 that is a broadcast address in the original network of the figure but that looks like an host address in the aggregated route 10.0.0.0/22.

Nevertheless, this does not represent a problem. In fact, although the router with aggregated routes will forward a packet directed to one of those addresses toward the destination, sooner or later that packet will be delivered to a router that does not longer have aggregated routes (e.g., while R1 has the aggregated route 10.0.0.0/24, router R3 will have two different routes for 10.0.0.0/24 and 10.0.1.0/24). At that point, that router will be able to understand that the packet is directed to an invalid address and therefore it will discard the packet. For instance, this remind us that in general the aggregation becomes more effective as we move far away from the networks that we want to aggregate. In fact, the closer to the edge of the network we are, the more effective should be the aggregation of the destinations that can be found on the distant Intenet, but the edge network may be hard to aggregate. Viceversa, in the Internet backbone we may have an easier task to aggregate our edge networks, but we may loose some aggregations related to other distant networks.

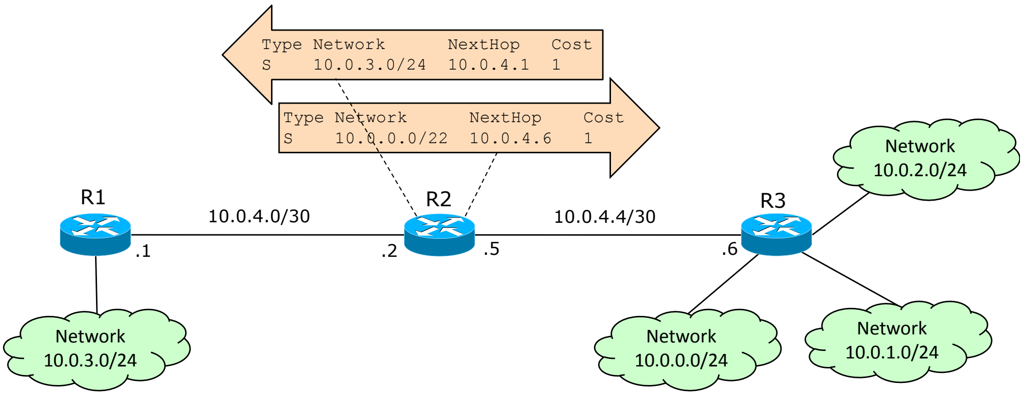
Finally, we should remember that directly connected networks cannot be aggregated. Although, in principle, directly connected routes are just routes and hence they may be aggregated, in practice those routes are automatically configured by the router because they originate from directly connected IP networks and they can not be cancelled in the routing table[6].

1.1.6. Most specific routes

The IP routing also supports the concept of *most specific route.* In practice, we are allowed to define two routes in which the address space assigned to the former is a subset of the address space assigned to the latter (e.g., 10.0.3.0/24 and 10.0.0.0/22). In case a packet is directed to a destination that matches both routes, we select the most specific one. This is equivalent to say that the route with the longest prefix length matches (a route to an address range /24 is more specific than a route to an address range /22).

The following figure may help to clarify this concept.

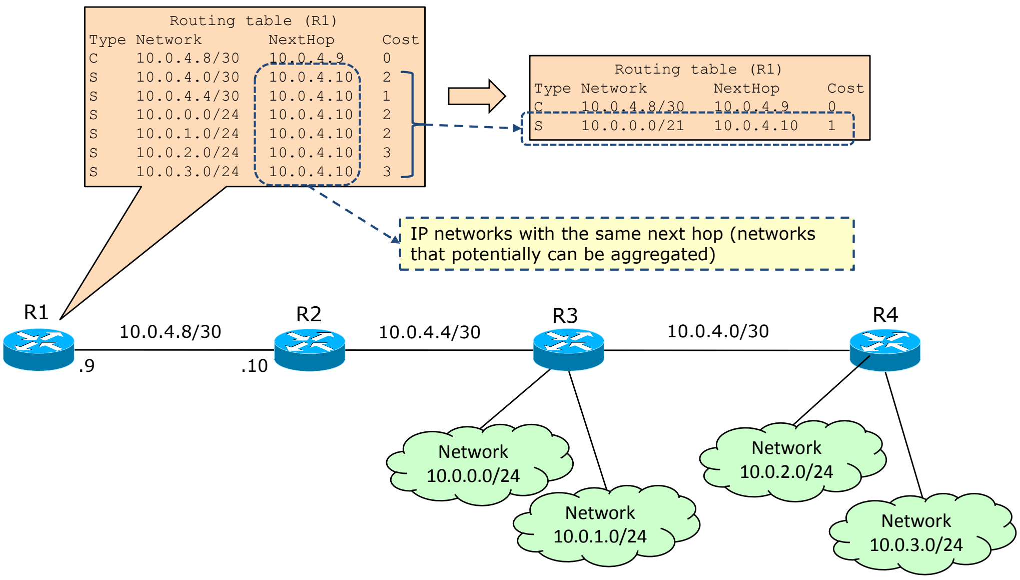
[6] For instance, we feel that aggregation of directly connected routes does not make sense. Those routes specify networks that need to be reached by using different directions (in fact, we can safely assume that they are reachable through different next hops, even if those addresses are associated to the same interface), hence we are not able to satisfy the first requirement for aggregating routes, i.e., the necesity to have the same next hop.



The router R2 in the previous figure has two routes: one for the address range 10.0.0.0/22 that points toward R3 and one for the address range 10.0.3.0/24 that points toward R1. We can now assume that the router R2 has an IP packet that has to be forwarded to the host 10.0.2.2: this address is included in the address range 10.0.0.0/22; hence it will be sent toward R3. Now, let us assume that R2 must forward a packet to the host 10.0.3.3: this host belongs to both the address ranges, i.e., this address is included in both the first (i.e., 10.0.3.0/24) and the second route (i.e., 10.0.0.0/22). As the two routes in fact point to different paths (actual, the first route points to R3, while the second points to R1), we need to define a criterion to determine which path has to be followed by the packet. In the IP world, the rule is that **the most specific route wins**. In this case, thus, the packet will be forwarded to R1.

The *most specific route* rule enables a much more efficient aggregation of the routes compared to the necessity to use only *exact* address ranges when defining the aggregation, because the aggregation of IP routes may actually use an address range that is bigger than the simple union of the involved routes.

Once again, a figure is useful to make this clearer.



In this example it is possible to directly aggregate in a single route all the destination sharing their next hop router (shown in figure), for example, replacing them with a single route to the address range

10.0.0.0/21. This addressing space includes the addresses from 10.0.0.0 to 10.0.7.255 and so gathers all the destinations that were previously known inside the single route. This address range includes also the destinations 10.0.4.8/30 that instead do not belong to the routes whose next hop is 10.0.4.10: this is not a problem because to these destinations there is a more specific route.

It is interesting to notice that the new address range 10.0.0.0/21 actually includes other destinations that would not be part of the simple union of the addressing spaces of the single routes, that is to say that the range of addresses from 10.0.4.12 to 10.0.7.255. This is normally not a problem because the destination are not in any way reachable in the network and thus what happens is that the packets heading to these destinations are forwarded in a direction but sooner or later a router in this path will notice that these destinations can not be reached and will discard the packets.

Referring to the example of the previous figure, the behaviors of the network receiving three hypothetical packets can be studied:

- Packet heading to the host 10.0.1.1: the destination address is included in the route 10.0.0.0/21 and it is thus correctly forwarded to R2.
- Packet heading to the host 10.0.4.10: the destination address is included in both the addressing space of the route 10.0.0/21 and the one of the route 10.0.4.8/30. In this case the most specific route is chosen and the packet is thus correctly forwarded by direct routing to the destination 10.0.4.10.
- Packet heading to the host 10.0.7.7: the destination is included in the route 10.0.0.0/21 and it will thus be forwarded to R2. Obviously, this destination does not exist in this network but this is of no importance: the router R2 may realize this (even though it could also have a route containing the address 10.0.7.7) but probably sooner or later a router will realize the non-existence of the host and discard the packet.

One of the problems that can appear using the aggregation technique with more extended routes (often indicate as *supernet routes*) compared with the original addressing space is the creation of *loops* in the forwarding of packets. Assume for instance that R1 in figure had a default route to the right, while R2 had a default route to the left. In this case a packet heading to the host 20.2.2.2 does not belong to any specific route and must thus be forwarding along the default route. At this point, then, R1 will send the packet to R2 that will send it back to R1 and so on as long as its life span has not been exceeded.

Generally, the loop problem can emerge when *supernets* are used: the network administrator has the responsibility to design the routing in a way that prevents that kind of behavior.

Finally, remember that the routing table is different on each router. Thus the way the aggregations can be handled depends from router to router on the way the various networks addressing spaces can be aggregated.

1.1.7. Default route

Extending the concept of aggregation, it is possible to examine how the router R1 lay also choose other address ranges to aggregate its routes: instead of specifying the address range 10.0.0.0/21 it may specify a default route that would be used to handle all the destinations that can be reached through a given next hop. In this case, the routing table of R1 would become:

```
Type   Network       Next Hop      Cost
C      10.0.4.8/30   10.0.4.9      0
S      0.0.0.0/0     10.0.4.10     1
```

The choice of an option instead of another is determined by the context and the preference of the operator. Generally, with a default route it is possible to aggregate all the routes to a given direction, replacing N routes that converge to the same next hop by a single one. Nevertheless it can be used only once (obviously, it is not possible to have *two* default routes on the same router).

In practice, the default route is commonly used when an user network is connected to Internet because it allows to reach all destinations on Internet without having to explicitly name each one of them.

1.1.8. Different costs routes

It is possible in a routing table to indicate two alternative routes with different cost, as in this example[7]:

```
Type   Network       Next Hop      Cost
S      20.2.2.0/24   30.3.3.3      2
S      20.2.2.0/24   40.4.4.4      3
```

In this case, the router will select the first route to the network 20.2.2.0/24 because its cost is smaller and the second one will be ignored. Nevertheless, in the case in which the first route becomes unavailable (for example the interface to the next hop 30.3.3.3 fails), the first route becomes useless and the second one will be used to have an alternative path to the same destination. This king of configuration is commonly named "*backup route*".

The use of such a configuration is however much limited. As seen in the theory, the static backup routes often do not function because the router is not able to notice failures of the non-directly connected networks (in some cases, no of these failures is noticed ; for example, the studied router may not notice that the interface 30.3.3.3 has been shut down.) and this may cause loops for packets. Actually, the custom is to use with much parsimony this function in case of static routing; in case of dynamic routing, this no longer necessary as the routing protocol foresees to recompute the addressing without the need for a "backup route".

Secondly, with this configuration topology, a additional level of ambiguity in the routes is introduced after the use of the aggregated routes because a packet can match more than one route with diverse costs. In case of ambiguity, the route choice algorithm will select first *(1)* the most specific route, then *(2)* the one will the smallest cost. This is equivalent to say that the routes with a bigger cost must have the same prefix length as the one with smaller cost else either they will never be selected 'if they refer to supernets) or they will always be selected (if they are more specific).

For instance, with the following routing table:

```
Type   Network       Next Hop      Cost
S      20.2.2.0/24   30.3.3.3      2
S      20.2.2.0/23   40.4.4.4      3
```

the second route will never be selected for the destinations 20.2.2.0/24 because the first one is more specific.

Vice-versa, with this configuration:

[7] Notice that this example has nothing to do with the topology used up to now. Actually, alternative routes with different costs have a meaning only when there are alternative paths to a given destination, which is not possible if the topology does not include any mesh.

```
Type    Network        Next Hop       Cost
S       20.2.2.0/24    30.3.3.3       2
S       20.2.2.0/25    40.4.4.4       3
```

the second route will always be selected for destinations 20.2.2.0/25 because it is more specific for these destinations, independently of the cost. The one case in which the cost is taken into account in the choice of the route is thus when both destination networks are identical but the routes differ by the cost.

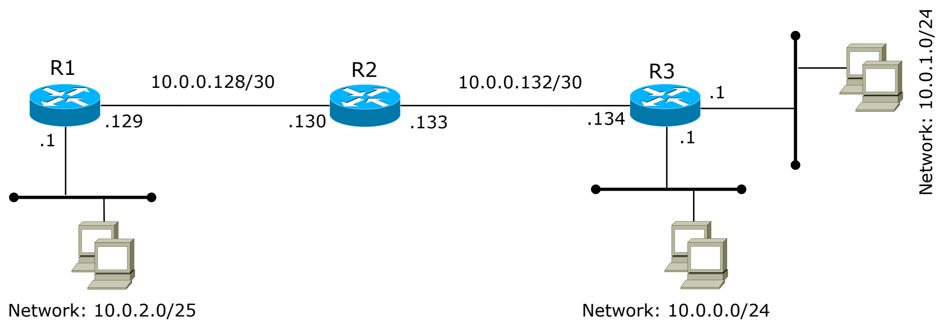
1.2. Deriving the routing table: methodology

Having pointed out the basic concepts of routing and of route aggregation, we can now propose a methodology for the definition of the routing table. Even though this methodology only aim is to create the routing table for each router, it can be used also to define an addressing plan that has a better aggregation capacity. Actually, as seen previously, the route aggregation is strongly dependent on how the IP networks have been assigned in the studied topology and thus the administrator will have to assign the addresses, as much as possible, in a way that favors the aggregability during routing.

The proposed methodology includes the following steps:

- Identify the list of the IP networks present in the given topology and type of the networks themselves (directly connected networks/ remote networks).
- Derive the spanning tree toward all IP networks found in the previous step.
- Determine the routing table, which includes a route for each destination (unless multiple paths with the same costs are available).
- Identify possible route aggregations.

The explanations will use the example in following figure.

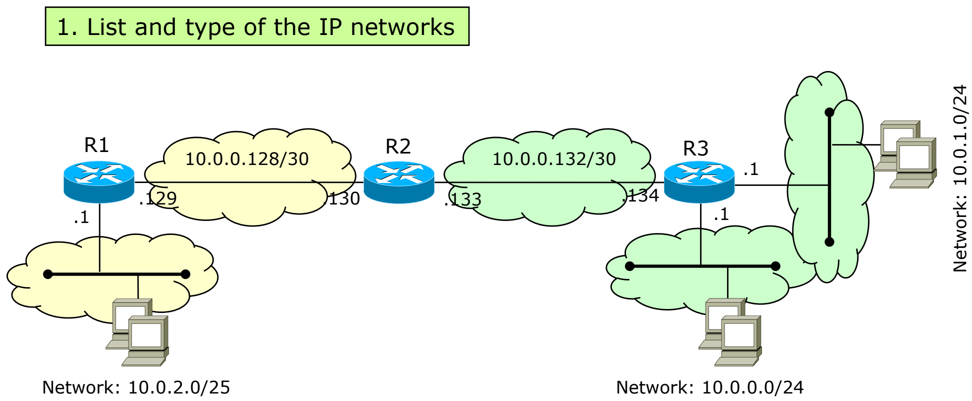


It is important to notice that the computing procedure of the routing table **must be repeated for each router**; for instance, in the previous network it will have to be done three times (because there are 3 routers). Actually, even if the chosen path by a router to a destination D is not completely independent of the path that the other routers will do to reach D (in other words, if R1 reaches the network 10.0.0.0/24 by sending the packets to R2, R2 can not reach the same destination by sending back the packets to R1), the spanning tree defines the best paths *from one point to all the possible destinations.* In consequence, changing the initial point from which the paths are computed (or the router whose routing table is being computed) will change the routing table and thus each router has to compute "autonomously" its own routing table.

1.2.1. List and type of the IP networks

During this step it is needed to simply point out which are the IP networks present in the studied topology, distinguishing between the directly connected IP networks (that is to say the ones reachable through direct routing by the studied router) and the remote networks (that is to say the ones reachable through indirect routing, which means that the packet is sent to the next router in the direction of the destination).

The result of the first step applied to the example is shown in figure.



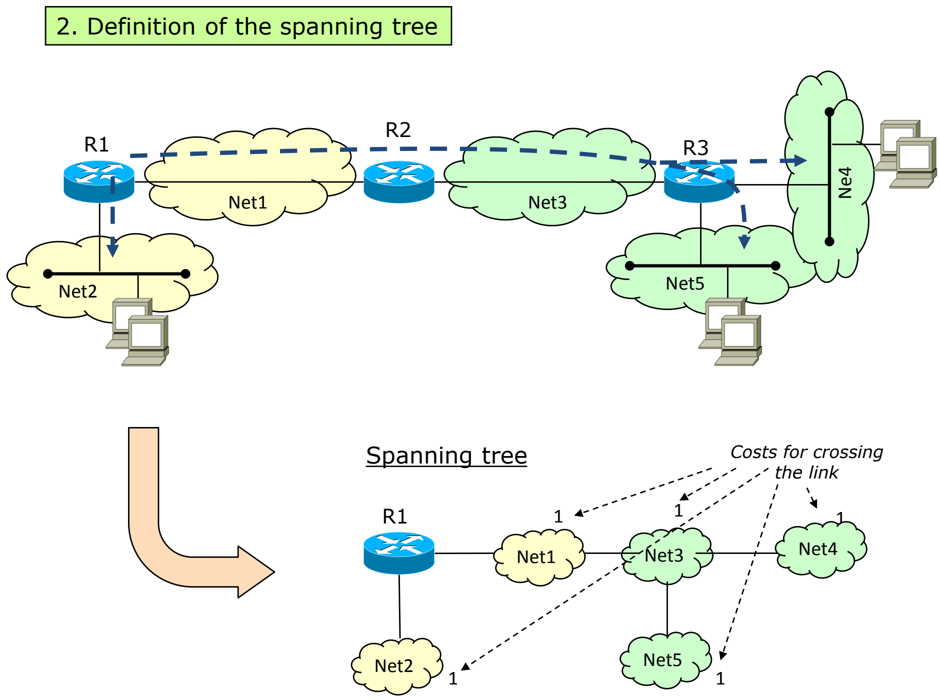
1.2.2. Derive the spanning tree

Given the list of the reachable destinations we needed to compute the *spanning tree* (in some cases called also *routing tree*), which is the set of paths that bring from a given router to all the available destinations, where each path is the one at the minimum cost.

In the case of a simple topology (as the one in figure) it is possible to use the innate human ability to determine the shortest paths. When confronted to more complex topologies it is possible to use algorithms for the computation of the *shortest path*, for example the algorithm of Dijkstra.

The result of this step will be the modeling of the topology in term of an acyclic graph (or a tree), representing the paths to reach all destinations in the network.

The result in the case of the example is shown in figure; in this case the result is particularly common due ti the lack of cycles in the original network that prevent the formation of multiple paths to a single destination.



Notice how import the **costs** of reaching the various networks are in this result. They are determined by the cost of going through the links (assumed unitary in the example network and shown with the value 1 in the spanning tree). For instance, the network *Net5* will be reachable with cost 3 from the router R1. Knowing the costs is mandatory in order to favor a path compared to an other one (and choose the best cost one) when multiple paths to the same destination exist.

Notice also that the **routers are not part of the topology graph**: the goal is to determine the best path to each destination and the intermediate routers do not give any additional information in the spanning tree. This is the reason why they can be omitted without loss.

1.2.3. Determine the routing table

Once the spanning tree has been derived, the routing table can be determined with a few easy steps. Each destination (or each IP network) must be written inside the routing table along the requested information (typology of the route, network/netmask, next hop, cost). The only possible source of problem is the difference between connected and remote networks with respect to the different value of the next hop in each case. Thus, for instance, for the network Net2 (directly connected) the routing table entry will be:

```
Type    Network        Next Hop       Cost
C       10.0.2.0/25    10.0.2.1       0
```

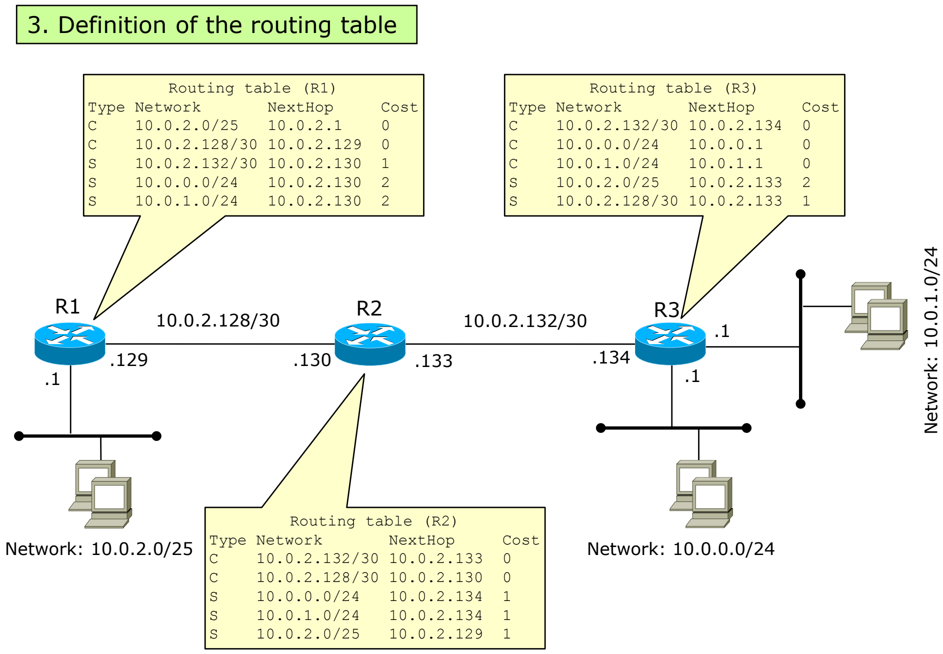
where the type of the route is "C" ("connected") and the next hop is the interface *of the router itself* that is used to reach these destinations in direct routing.

On the other hand, the entry in the routing table for the network Net5 (remote) will be:

```
Type    Network        Next Hop       Cost
S       10.0.0.0/24    10.0.2.130     2
```

where the type of the route is "S" ("static") and the next hop is the interface *of the next router* that is used to reach these destinations in indirect routing.

The result when applied to the example network is shown in figure. There is also the routing tables of all the routers in the studied topology.



Even if, as said previously, the cost is not particularly meaningful in the routing table, it is instrumental in the computation of the best path to reach this destination. Therefore notice that the cost of going through to reach the network Net1 is equal to zero, the one to reach the network Net3 is equal to 1 and the one to reach the network Net4 is equal to 2. The cost "S" of the neighbor of the Net 4 is thus irrelevant for the solution of the exercise and would be used only if they would be other reachable destinations than Net4.

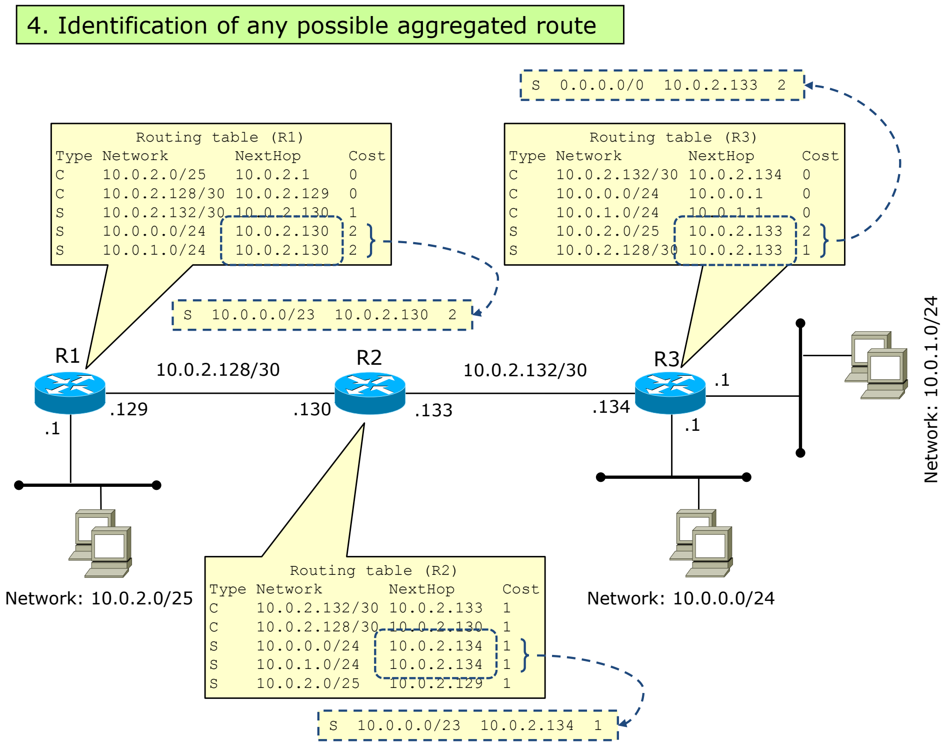
1.2.4. Identify possible route aggregations

The last step refers to the determination of route aggregations and is, in a way, subjective. The criterion that surely has to be satisfied is that the aggregatable routes share the same next hop and these routes must be related to networks *not* directly connected (because the directly connected routes can not be canceled from the routing table).

The non-objectivity of this operation stands in the number of routes that are aggregated in a given address range and in the used address range. The operator can make different choices depending on if he wants to favor the aggregation capability and thus uses supernets even really big (up to the default route), or if he wants to minimize the possible side effects limiting himself to the replacing if a given number of routes with a new address range that is exactly equivalent (that is to say the union of the original address ranges must be equal to the new address range).

The following figure shows both solutions: in the case of the routers R1 and R2 two networks /24 are aggregated with the equivalent /23 while in the case of the router R3 two remote networks (belonging to two non exactly aggregatable with a new address range address ranges) are replaced by a default route.

The resulting routing table is shown in next figure.



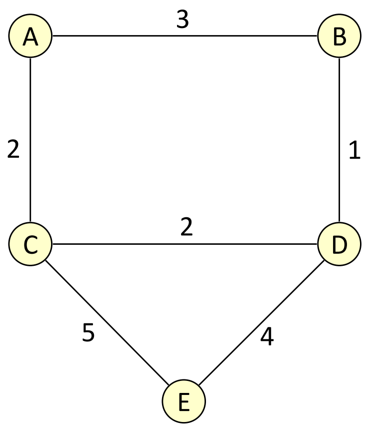
Part II.

Exercises

2. Exercises

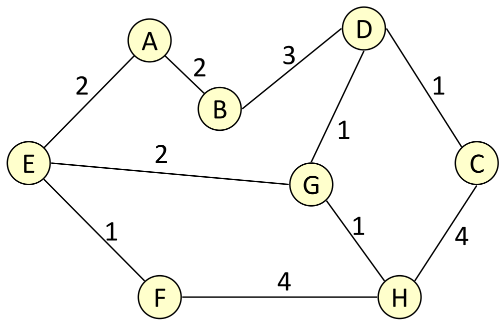
2.1. Exercise 1

Define the spanning tree of each node of the network in figure. In addition, write the routing table of each router in the format (destination router - next hop router).



2.2. Exercise 2

Define the spanning tree of each node of the network in figure. In addition, write the routing table of each router in the format (destination router - next hop router).

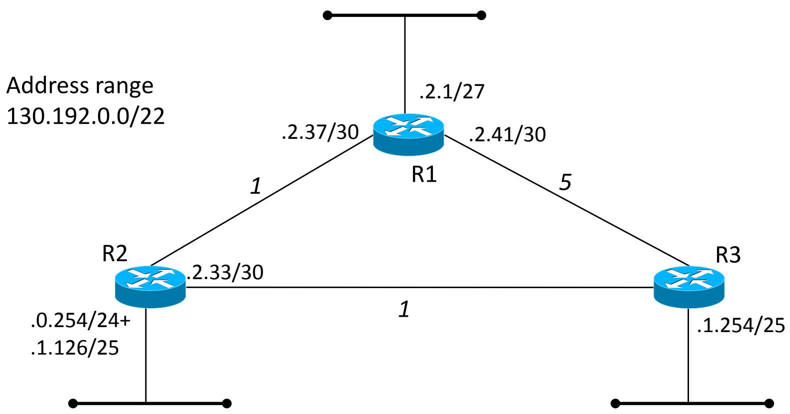


2.3. Exercise 3

Given the network in figure, determine the routing table of R1 by aggregating the routes in a way such as:

- the addressing spaces are exactly equivalent to the original ones
- (or) there are the fewest possible entries in the routing table

The numbers in italic on the network represent the going through cost of the link; assume unitary the costs not explicitly indicated in figure.

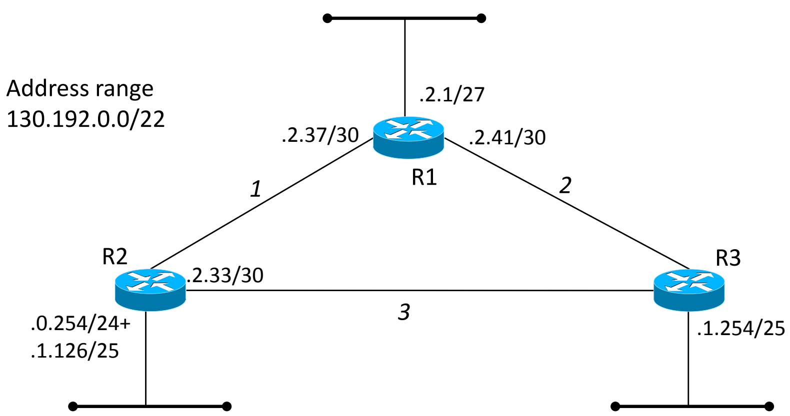


2.4. Exercise 4

Given the network in figure, determine the routing table of R1 by aggregating the routes in a way such as:

- the addressing spaces are exactly equivalent to the original ones
- (or) there are the fewest possible entries in the routing table

The numbers in italic on the network represent the going through cost of the link; assume unitary the costs not explicitly indicated in figure.

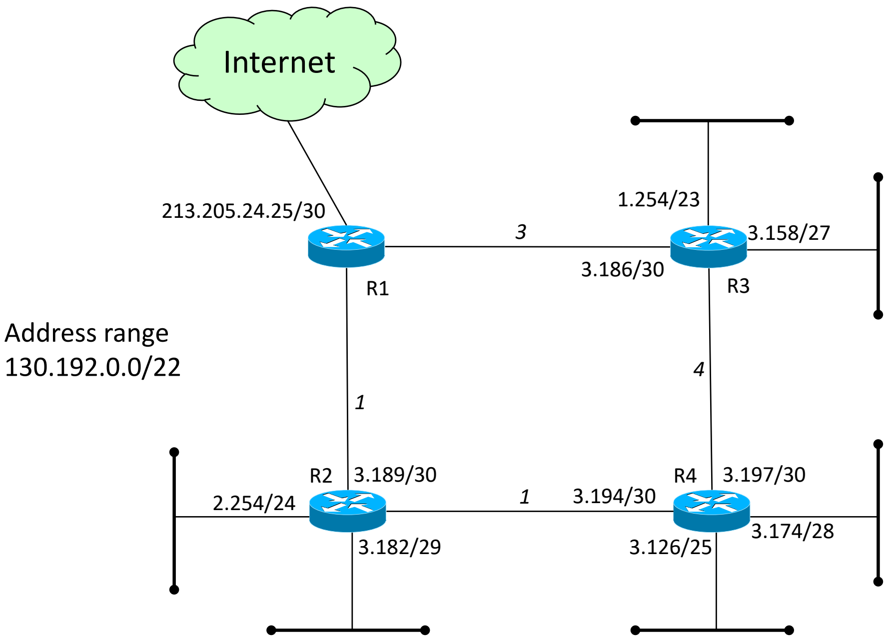


2.5. Exercise 5

Given the network in figure, determine the routing table of R1 by aggregating the routes in a way such as:

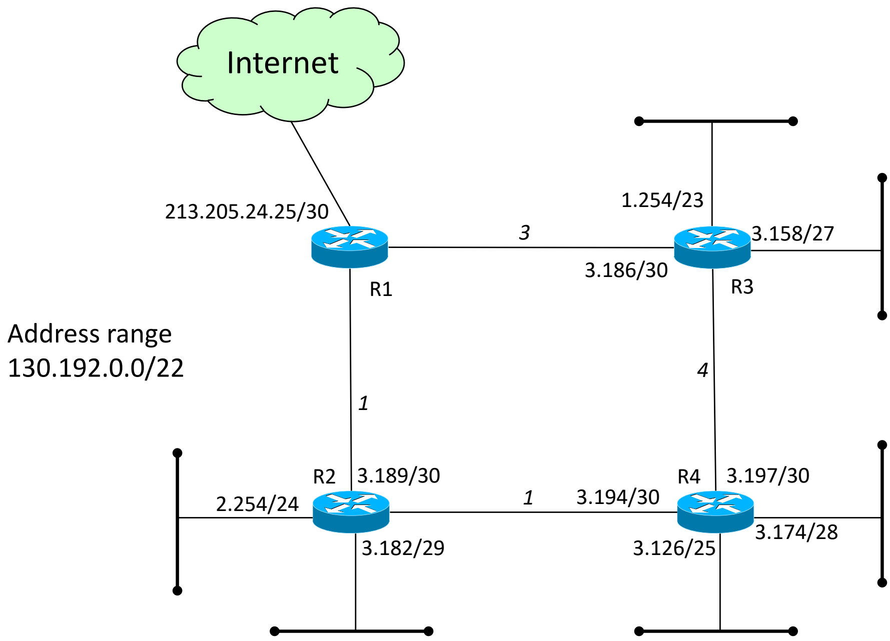
- the addressing spaces are exactly equivalent to the original ones
- (or) there are the fewest possible entries in the routing table

Numbers in italic represent the cost of the link; assume unitary costs when not explicitly indicated in the figure.



2.6. Exercise 6

Given the network of the previous exercise (indicated in figure), determine the routing table of R4 obtained when looking for the fewest possible entries in the routing table. Point out if the number of route is bigger or smaller than the one of routes in the routing table of the router R1 and explain why.

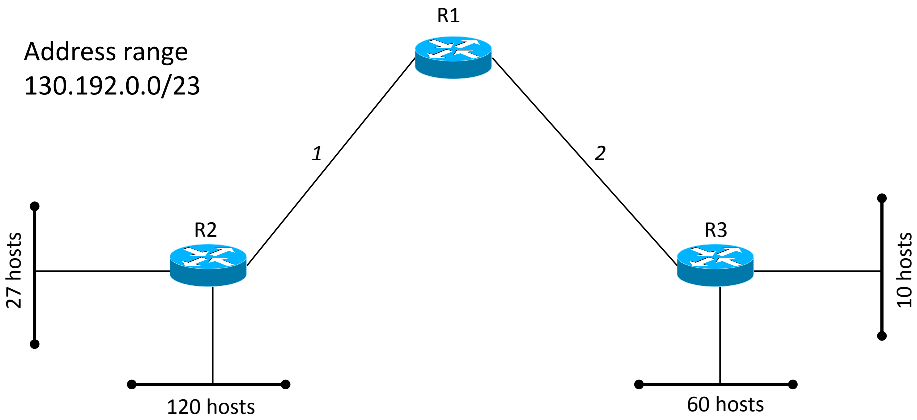


2.7. Exercise 7

Given the network in figure, define an addressing plan that:

- maximizes the aggregation of the routes on R1
- (or) minimize the number of addresses allocated to manage the network.

The numbers in italic on the network represent the going through cost of the link; assume unitary the costs not explicitly indicated in figure.

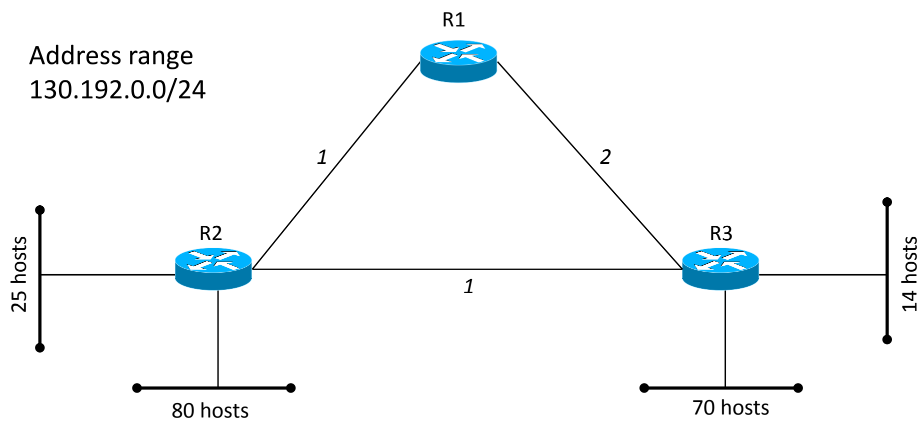


2.8. Exercise 8

Given the network in figure, define an addressing plan that:

- maximizes the aggregation of the routes on R1 (either in the case in which the addressing spaces are exactly equivalent to the original ones or in the case in which routes are managed through *supernets*)
- (or) minimize the number of addresses allocated to manage the network.

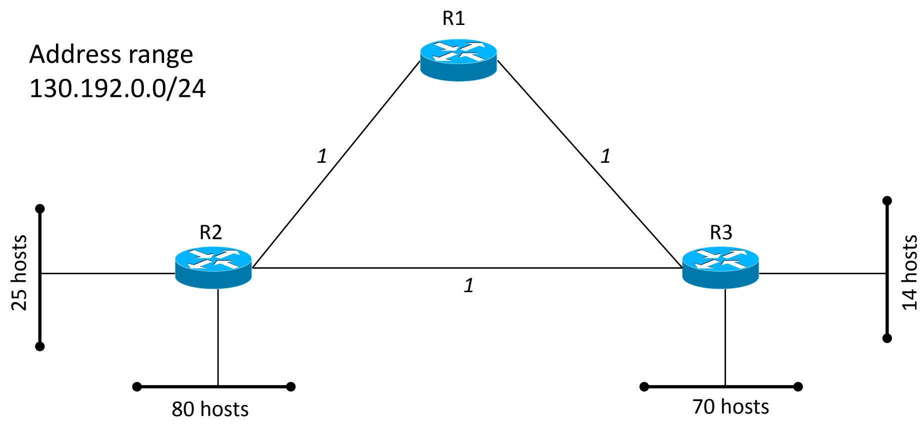
The numbers in italic on the network represent the going through cost of the link; assume unitary the costs not explicitly indicated in figure.



2.9. Exercise 9

Basing yourself on the previous exercise and given the network in figure (identical to the previous one except for the costs assigned to the links), define an addressing plan that maximizes the aggregation of the routes on R1, either in the case in which the addressing spaces are exactly equivalent to the original ones or in the case in which routes are managed through *supernets*.

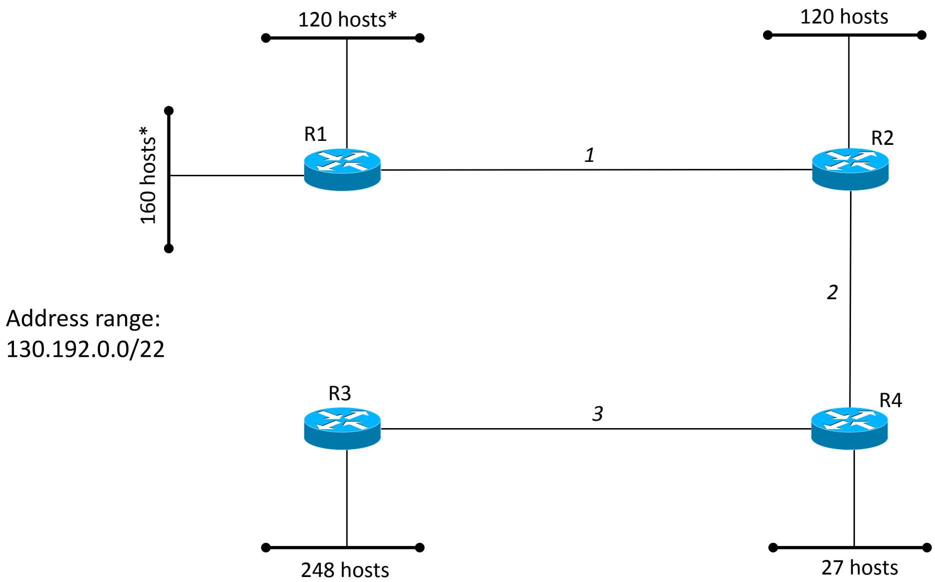
The numbers in italic on the network represent the going through cost of the link; assume unitary the costs not explicitly indicated in figure.



2.10. Exercise 10

Realize an addressing plan for the network in figure that maximizes the route aggregation on the router R2. In addition, write the routing table of each router, assuming that the aim is to maximize the route aggregation (supernets are accepted but not default routes).

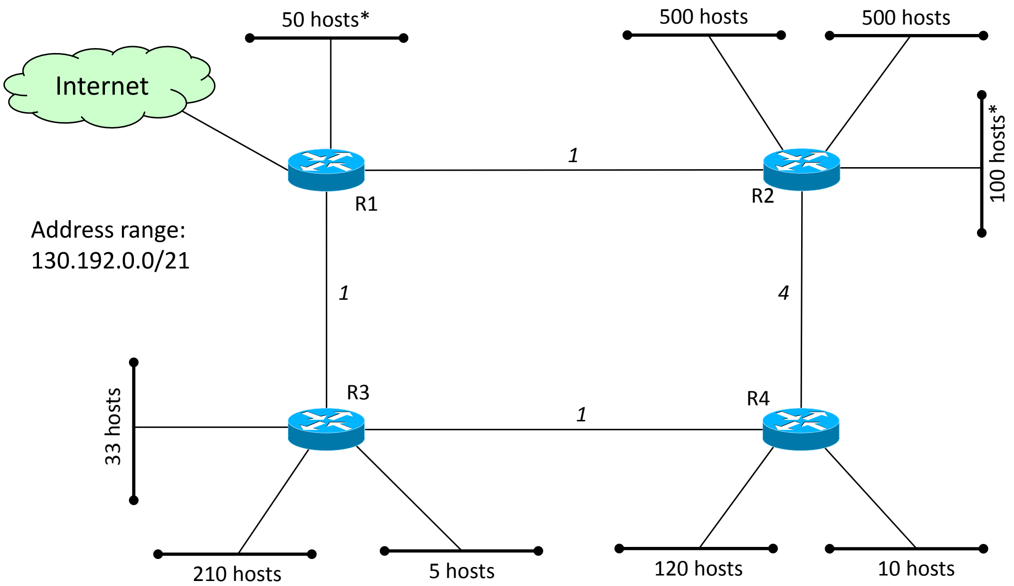
Notice that the networks indicated with an asterisk in figure are foreseen to receive the addition of hosts in the future.



2.11. Exercise 11

Realize an addressing plan for the network in figure that maximizes the route aggregation on the router R2. In addition, write the routing table of each router, assuming that the aim is to maximize the route aggregation (supernets are accepted but not default routes).

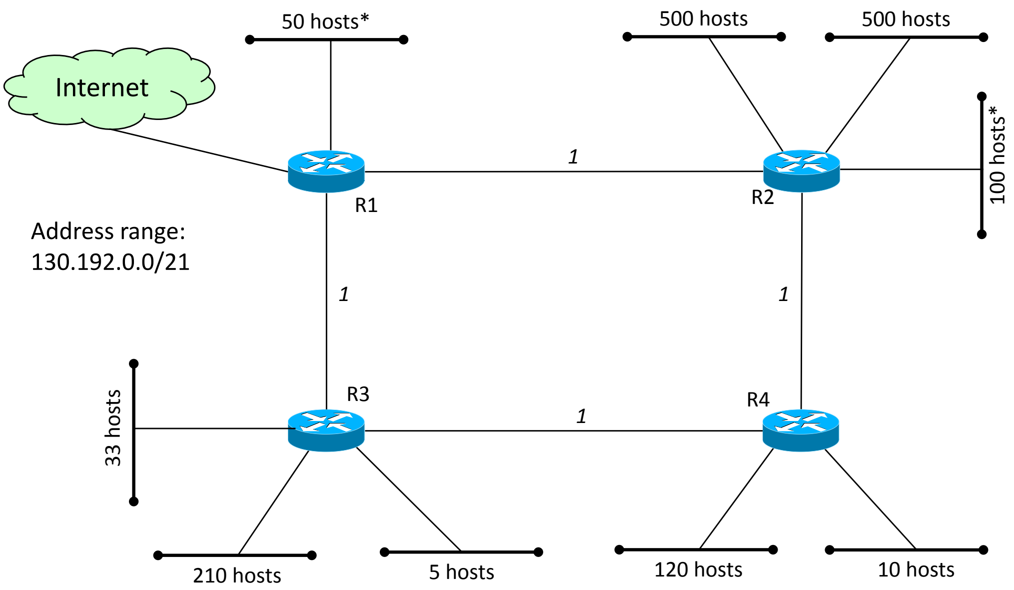
Notice that the networks indicated with an asterisk in figure are foreseen to receive the addition of hosts in the future.



2.12. Exercise 12

Given the same topology as in the previous exercise (but with different link costs), realize an addressing plan for the network in figure that maximizes the route aggregation on router R1. In addition, write the routing table of each router, assuming that the goal is to maximize the route aggregation (supernets are authorized)

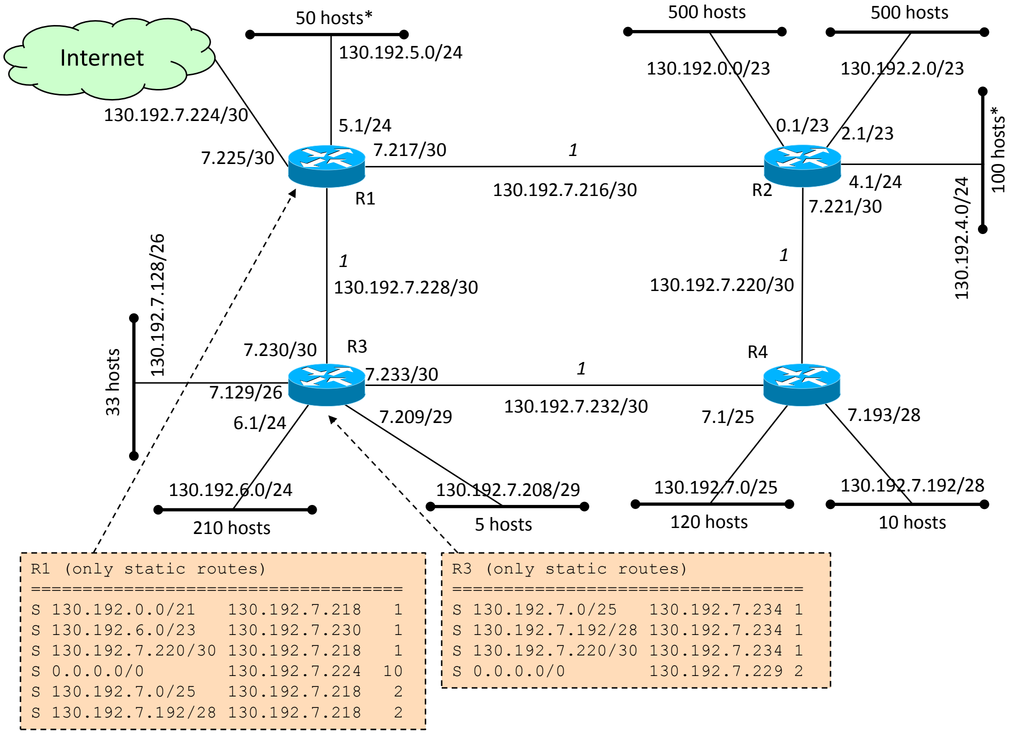
Notice that the networks indicated with an asterisk in figure are foreseen to receive the addition of hosts in the future.



2.13. Exercise 13

Given the topology of the previous exercise and the related routing tables of the devices, the network manager has decided to configure some additional static backup routes in order to be able to react automatically to some failures that may occur.

Particularly, the manager wants to warrant the router R1 the reachability of the LAN connected to R4 (reachable preferably through R3) even if the link R3-R4 fails (in this case, the path should go through R2). That is why the manager has modified the routing tables of R1 and R3 as shown in next figure (the directly connected routes are omitted for clarity):

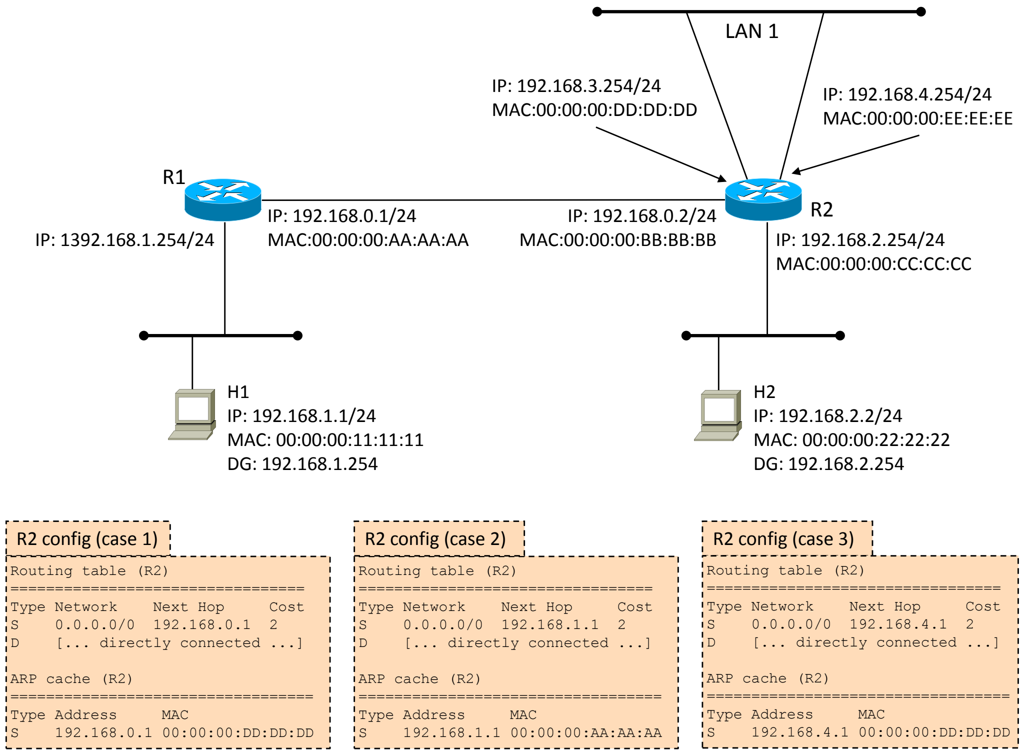


Determine if the modifications introduced by the manager are efficient; if not point out its problems and propose a way to improve it.

2.14. Exercise 14

Given the network in figure, the host H2 generates an IP packet for the host H1. The packet is correctly received by R2 which has to forward it to the destination. Given the different configurations of the router R2 at the routing table and ARP cache levels indicated in figure, determine the path followed by the packet in all three cases.

Finlay indicate if the solution would have been different if the network LAN1 had been realized using level 2 switches instead of a shared Ethernet infrastructure.



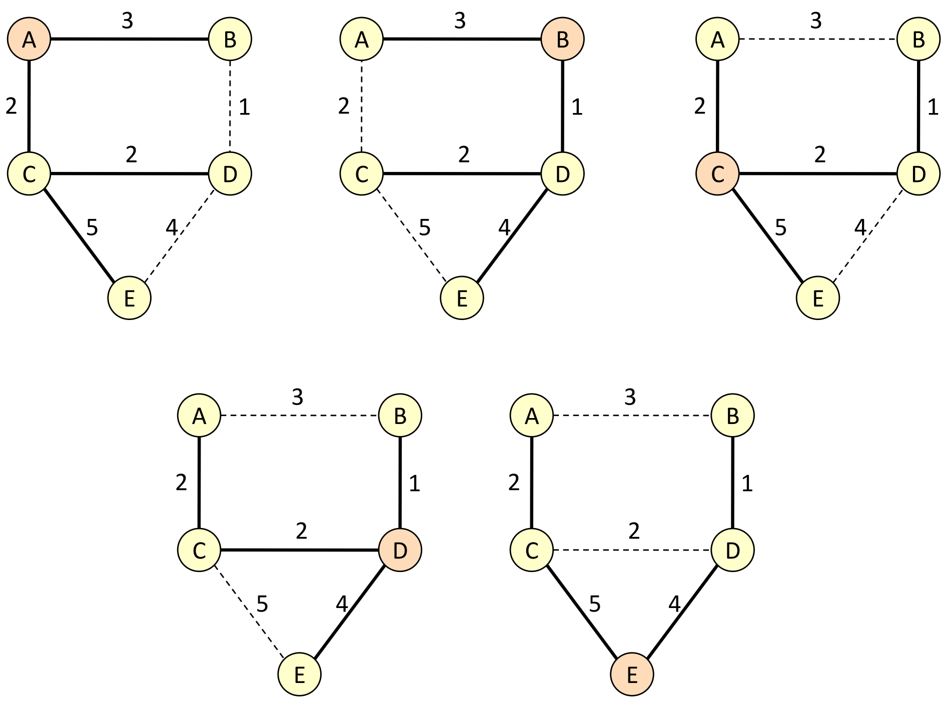
Part III.

Solutions

3. Solutions

3.1. Exercise 1

The spanning trees of the various nodes are reported in figure.



The one ambiguity is linked to the destinations that have more than one way with same cost (for instance the router A has two equivalent ways to the destination D). The schema chows only one of these solutions to be clearer, but both are reported on the following routing tables. To be more precise, in case of equivalent paths, the choice of one or the other (in a purely routing-oriented view) is completely arbitrary.

The tables for each router are the following:

Node E

Destination	Next-hop
A	C
B	D
C	C
D	D

Node A

Destination	Next-hop
B	B
C	C
D	B/C
E	C

Node B

Destination	Next-hop
A	A
C	D
D	D
E	D

Node C

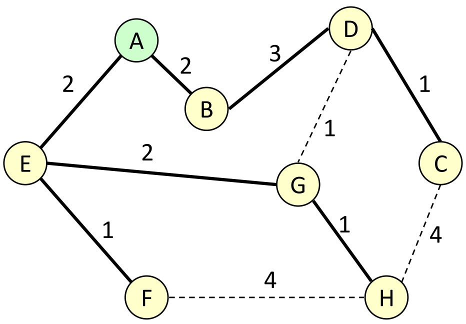
Destination	Next-hop
A	A
B	D
D	D
E	E

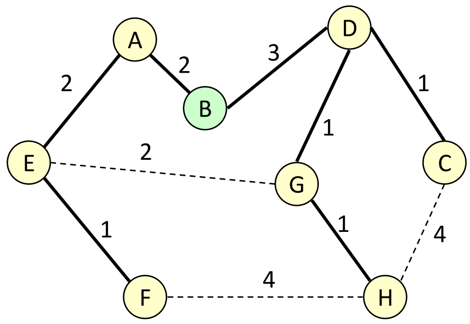
Node D

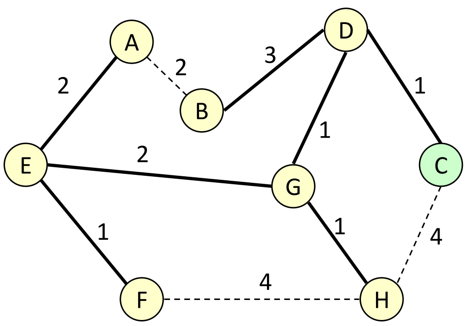
Destination	Next-hop
A	B/C
B	B
C	C
E	E

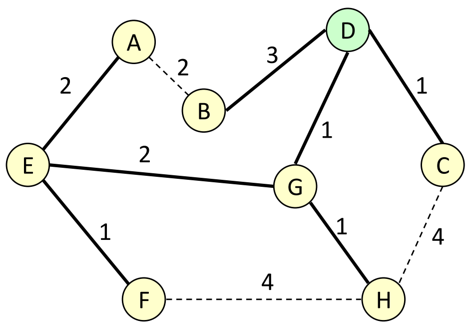
3.2. Exercise 2

The solution of this exercise is similar ton the one of previous exercise. The spanning trees are reported for a subset of the present routers but all the routing tables are written down.









Notice that some router have multiple equivalent paths to a given destination (for example C has two equivalent paths with cost 6 to A), but this does not appear in the routing table because both use the same next hop router (D).

Node A

Destination	Next-hop
B	B
C	B/E
D	B/E
E	E
F	E

Node B

Destination	Next-hop
A	A
C	D
D	D
E	A
F	A

Node C

Destination	Next-hop
A	D
B	D
D	D
E	D
F	D
G	D
H	D

Node D

Destination	Next-hop
A	B/G
B	B
C	C
E	G
F	G
G	G
H	G

Node E

Destination	Next-hop
A	A
B	A
C	G
D	G
F	F
G	G
H	G

Node F

Destination	Next-hop
A	E
B	E
C	E
D	E
E	E
G	E
H	E/H

Node G

Destination	Next-hop
A	E
B	D
C	D
D	D
E	E
F	E
H	H

Node H

Destination	Next-hop
A	G
B	G
C	G
D	G
E	G
F	G/F
G	G

3.3. Exercise 3

The routing tables of the router R1 are indicated in the next tables, depending on the criterion chosen to do the aggregations.

3.3.1. Routes with equivalent addressing space

Type	Destination network	Next hop	Cost
C	130.192.2.0/27	130.192.2.1	0
C	130.192.2.36/30	130.192.2.37	0
C	130.192.2.40/30	130.192.2.41	0

S	130.192.0.0/23	130.192.2.38	1
S	130.192.2.32/30	130.192.2.38	1

3.3.2. Routes with maximal aggregation

Type	Destination network	Next hop	Cost
C	130.192.2.0/27	130.192.2.1	0
C	130.192.2.36/30	130.192.2.37	0
C	130.192.2.40/30	130.192.2.41	0
S	0.0.0.0/0	130.192.2.38	1

3.4. Exercise 4

The routing tables of the router R1 are indicated in the next tables, depending on the criterion chosen to do the aggregations.

3.4.1. Routes with equivalent addressing space

Type	Destination network	Next hop	Cost
C	130.192.2.0/27	130.192.2.1	0
C	130.192.2.36/30	130.192.2.37	0
C	130.192.2.40/30	130.192.2.41	0
S	130.192.0.0/23	130.192.2.38	1
S	130.192.2.32/30	130.192.2.38	1
S	130.192.1.128/25	130.192.2.42	2

In this case, it is not possible to do much aggregation because the paths to reach the various networks are different (some go through R2 and others go through R3) and because of the restriction on the use of an equivalent addressing space.

3.4.2. Routes with maximal aggregation

Type	Destination network	Next hop	Cost
C	130.192.2.0/27	130.192.2.1	0
C	130.192.2.36/30	130.192.2.37	0
C	130.192.2.40/30	130.192.2.41	0
S	0.0.0.0/0	130.192.2.38	2
S	130.192.1.128/25	130.192.2.42	2

In this case, a default route is used in one direction (the one with the biggest number of single routes) to minimize the number of routes.

3.5. Exercise 5

The routing tables of the router R1 are indicated in the next tables, depending on the criterion chosen to perform the aggregations.

3.5.1. Routes with equivalent addressing space

Type	Destination network	Next hop	Cost
C	213.205.24.24/30	213.205.24.25	0
C	130.192.3.188/30	130.192.3.190	0
C	130.192.3.184/30	130.192.3.185	0
S	0.0.0.0/0	213.205.24.26	1
S	130.192.2.0/24	130.192.3.189	1
S	130.192.3.0/25	130.192.3.189	1
S	130.192.3.160/27	130.192.3.189	1
S	130.192.3.192/29	130.192.3.189	1
S	130.192.0.0/23	130.192.3.186	3
S	130.192.3.128/27	130.192.3.186	3

Please note that the route `130.192.3.160/27` includes also the address ranges `130.192.3.184/30` and `130.192.3.188/30` that are reachable through another next hop. However the above route are more specific, hence can be aggregated in a larger address range.

Notice that in this case the studied topology is connected to the Internet and thus it is consented to use a default route even in the case of an equivalent addressing space. Actually, one must assume that all the addresses not present in the studied topology are present in the Internet; the default route will thus be a route to these destinations.

3.5.2. Routes with maximal aggregation

Type	Destination network	Next hop	Cost
C	213.205.24.24/30	213.205.24.25	0
C	130.192.3.188/30	130.192.3.190	0
C	130.192.3.184/30	130.192.3.185	0
S	0.0.0.0/0	213.205.24.26	1
S	130.192.2.0/23	130.192.3.189	1
S	130.192.0.0/23	130.192.3.186	3
S	130.192.3.128/27	130.192.3.186	3

Notice that, the studied topology being connected to the Internet, the default route is basically used to reach these destinations and can not be used to aggregate the internal networks.

3.6. Exercise 6

This exercise is not really different from the previous ones. Notice that the default route is more useful for the router R4 than for the router R1 because this router reaches several internal destinations with the same next hop used to reach the Internet. In this case, these internal networks will be gathered directly inside the default route without the use of an explicit route.

The routing table of R4, created in order to maximize the efficiency of the aggregation, will be smaller than the one of R1 and will thus be:

Type	Destination network	Next hop	Cost
C	130.192.3.196/30	130.192.3.197	0
C	130.192.3.160/28	130.192.3.174	0
C	130.192.3.192/30	130.192.3.194	0
C	130.192.3.0/25	130.192.3.126	0
S	0.0.0.0/0	130.192.3.193	2
S	130.192.0.0/23	130.192.3.198	4
S	130.192.3.128/27	130.192.3.198	4

Notice that in this case the number of static routes decreases by two units (due to the better aggregation) while the number of routes related to directly connected networks, that can not be aggregated, is incremented (due to the bigger number of directly connected networks). The total sum is thus smaller: the total number of routes in the routing table of R4 is decreased by one unit compared to the one present in the router R1.

3.7. Exercise 7

3.7.1. Addressing done trying to maximize routes aggregation on R1

The first step to solve the exercise is to compute the required addresses to make the addressing plan. As the networks directly connected to R1 have no influence in the aggregability on this router, they are momentarily left out of the total computation of the necessary addresses for the two defined areas.

In the computation of the requested addresses, one can distinguish between the remote networks reachable from R1 through R2 (gathered in *Area 1*) and those reachable through R3 (gathered in *Area 2*) because, being reachable in diverse directions, they are obviously not aggregatable. One gets then:

- Area 1: 27 (+3) + 120 (+3) requested addresses= 32 + 128 allocated addresses= **160 addresses**
- Area 2: 60 (+3) + 10 (+3) requested addresses= 64 + 16 allocated addresses= **80 addresses**

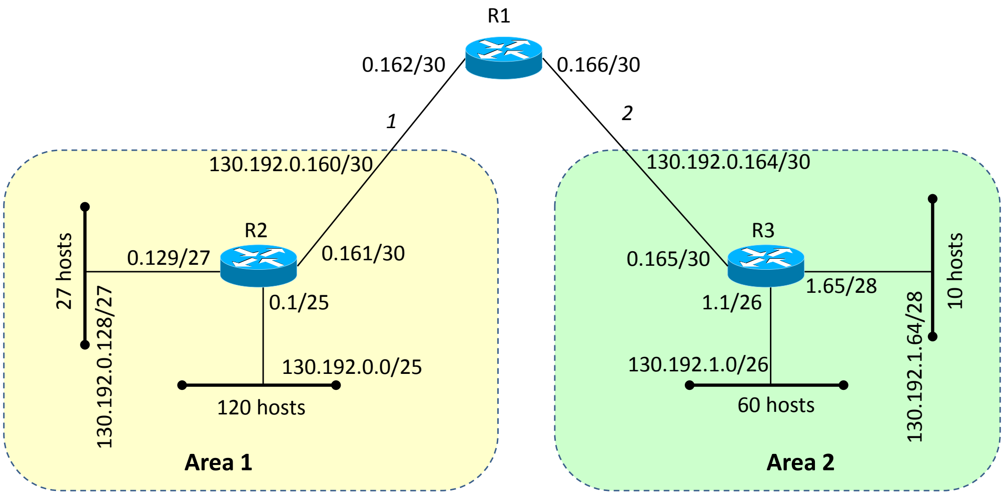
To maximize the aggregation, it is mandatory that the addressing space comprehensively allocated to the networks in Area 1 is disjoint from the addressing space comprehensively allocated to Area 2. In other words, to maximize routes aggregability it is appropriate to define some addressing spaces to the areas and inside them to recover then the address of the single networks. The addressing spaces of the areas will then be the "equivalent" address ranges that will be used to define the destinations of the aggregated routes.

The following address ranges are thus chosen:

- Area 1: **130.192.0.0/24**
- Area 2: **130.192.1.0/25**

The addressing space of the networks directly connected to R1 can be obtained either inside the address range of Area 1 or inside the address range of Area 2 (neither of them uses the whole address range that it has been allocated). In this exercise, it has been chosen to use part of the unused address range of Area 1 for these networks.

The resulting addressing plan is:

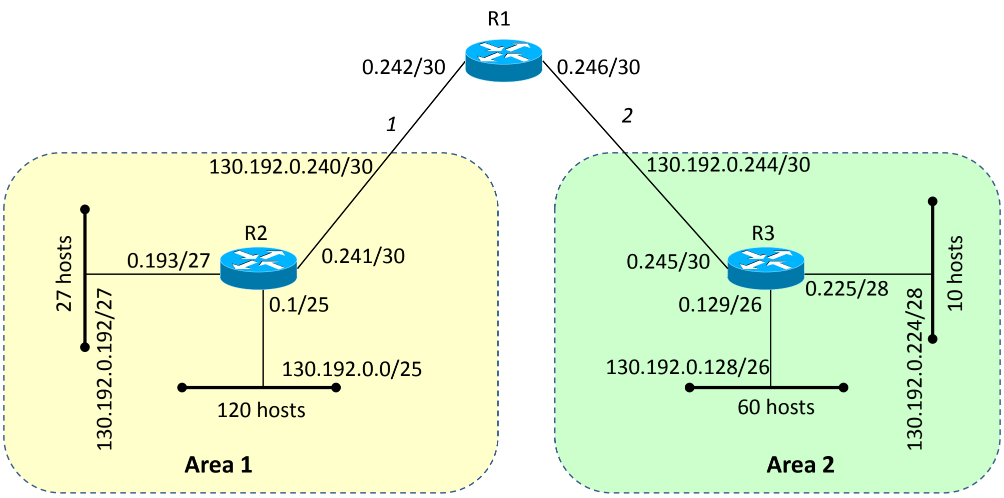


The corresponding routing table of R1 (without the use of a default route) is:

Type	Destination network	Next hop	Cost
C	130.192.0.160/30	130.192.0.162	0
C	130.192.0.165/30	130.192.0.166	0
S	130.192.0.0/24	130.192.0.161	1
S	130.192.1.0/25	130.192.0.165	2

3.7.2. Addressing done trying to minimize the number of allocated addresses

If the aim is however the minimization of the allocated address, the addressing plan is done in a more traditional way and may be the following:



Notice that in this case a /24 addressing space is enough to handle the whole topology, while in the previous case a /23 space was necessary.

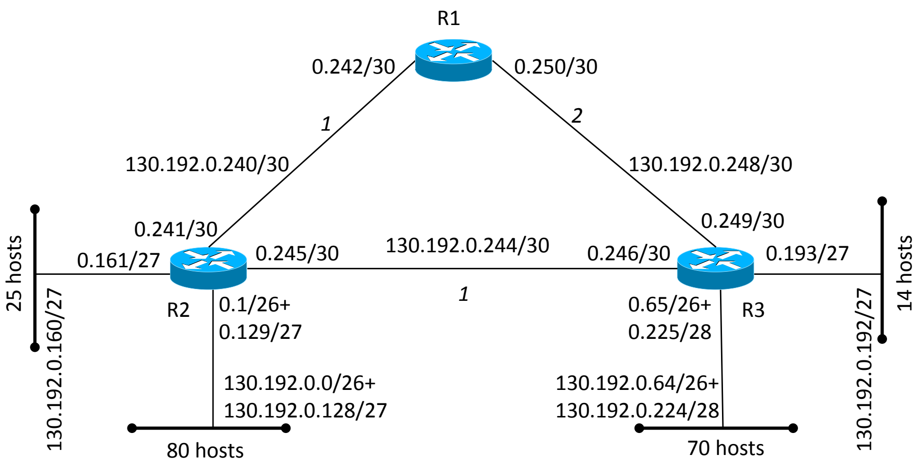
The corresponding routing table of R1 (without the use of a default route) is:

Type	Destination network	Next hop	Cost
C	130.192.0.240/30	130.192.0.242	0
C	130.192.0.244/30	130.192.0.246	0
S	130.192.0.0/24	130.192.0.241	1
S	130.192.0.128/26	130.192.0.245	2
S	130.192.0.224/28	130.192.0.245	2

3.8. Exercise 8

In this case both solutions, the one trying to maximize the aggregation and the one trying to minimize the allocated addresses, coincide. Actually, the addressing space is so reduced (130.192.0.0/24) that it is in any case mandatory to minimize the allocated addresses to be able to handle the studied topology. Hence, some physical networks (LAN) must be partitioned in more IP networks ti make the management possible.

The resulting addressing is shown in figure:



To write the table it is possible to use the fact that the destination connected to R3 may be reached with the same cost through the direct link R1-R3 or through the path R1-R2-R3. The second way is chosen because, with equal cost, it decreases the number of static routes present on the routing table.

In the case in which the aggregated addressing space is exactly equivalent to the original one, it is necessary to take into account that the network 130.192.0.252/30 is allocated to no one and thus it can not be part of the aggregation. That is why it is mandatory to write a set of lines in the routing table in a way not to include this space.

The resulting routing table will then be:

Type	Destination network	Next hop	Cost
C	130.192.0.240/30	130.192.0.242	0
C	130.192.0.248/30	130.192.0.250	0
S	130.192.0.0/25	130.192.0.241	2
S	130.192.0.128/26	130.192.0.241	2
S	130.192.0.192/27	130.192.0.241	2
S	130.192.0.224/28	130.192.0.241	2
S	130.192.0.244/30	130.192.0.241	1

In the case in which the aggregated addressing space may include even destinations not originally part of the studied topology, it is possible to aggregate all remote destinations in the route 130.192.0.0/24, obtaining an aggregation clearly more efficient than the previous one.

The resulting routing table will then be:

Type	Destination network	Next hop	Cost
C	130.192.0.240/30	130.192.0.242	0
C	130.192.0.248/30	130.192.0.250	0
S	130.192.0.0/24	130.192.0.241	2

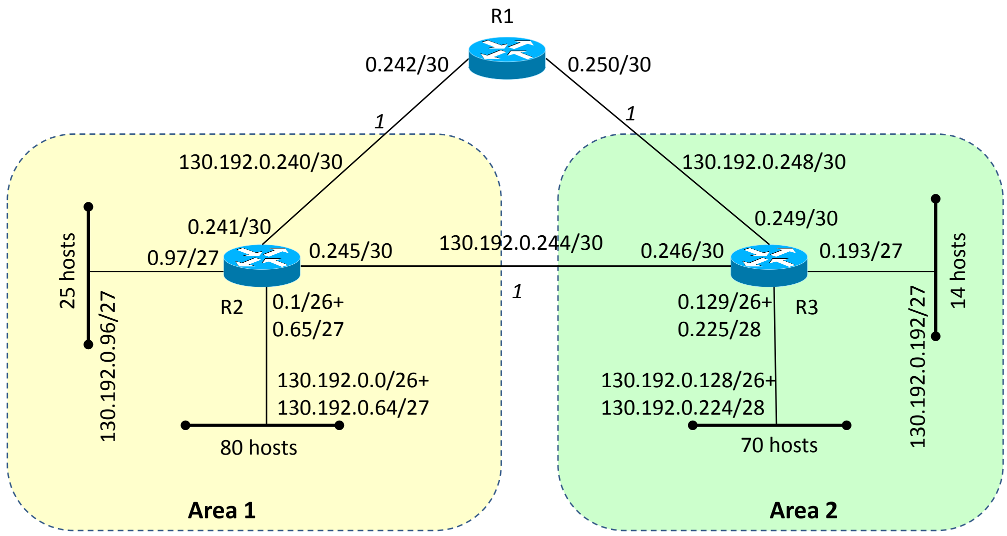
Notice that, in practice, it will be better to choose the second routing table, all the more because the network 130.192.252.0/30 is not physically used in the studied topology but it has been assigned to the network manager (the addressing space attributed to the manager is 130.192.0.0/24) and reasonably thus this network is nowhere else to be found.

3.9. Exercise 9

As the network is identical to the one of the previous exercise, there is no apparent difference at addressing level.

Nevertheless, the fact that the networks connected to R2 and R3 are reached through diverse next hop from R1 (due to different link costs) makes the situation much more complex because the networks are less aggregatable. That is why the possibility to position the contiguous IP addressing spaces in the same area is of instrumental importance for the aggregability. For instance, using the addressing level proposed in the previous solution the networks 130.192.0.0/26 and 130.192.0.64/26 will no longer be aggregatable in a single route 130.192.0.0/25 because they would have diverse next hops (one to Area 1 and the other to Area 2).

A possible new addressing level is indicated in figure:



In this case, the resulting routing tables will respectively be:

Addressing space equivalent to the one present in the topology:

Type	Destination network	Next hop	Cost
C	130.192.0.240/30	130.192.0.242	0
C	130.192.0.248/30	130.192.0.250	0
S	130.192.0.0/25	130.192.0.241	1
S	130.192.0.128/26	130.192.0.249	1
S	130.192.0.192/27	130.192.0.249	1
S	130.192.0.224/28	130.192.0.249	1
S	130.192.0.244/30	130.192.0.249	1

Maximization of the aggregation:

Type	Destination network	Next hop	Cost
C	130.192.0.240/30	130.192.0.242	0
C	130.192.0.248/30	130.192.0.250	0
S	130.192.0.0/25	130.192.0.241	1
S	130.192.0.128/25	130.192.0.249	1

3.10. Exercise 10

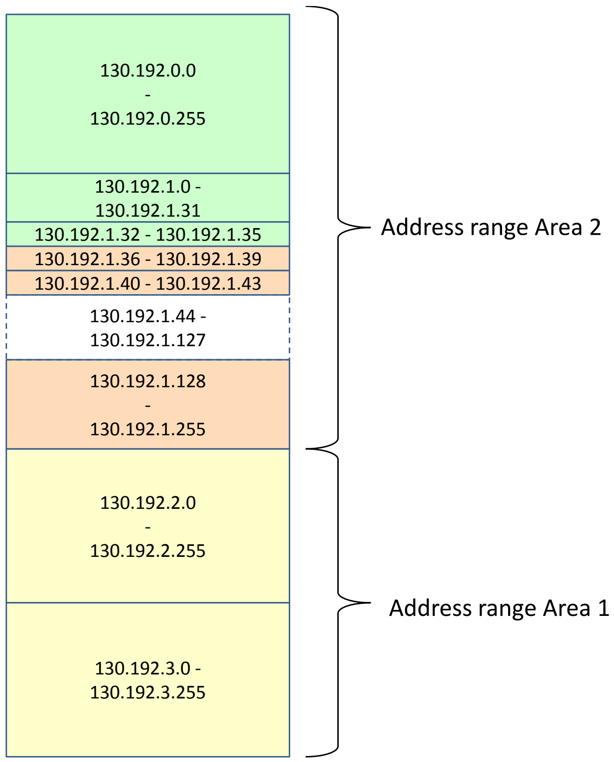
To determine the addressing level, the requested addresses to manage each network are computed first of all considering that the router R2 will see every remote destination through two next hops:

- Remote destinations through R1 (Area 1):
  - Network with 120 hosts*: 256 addresses are reserved (for future expansions)
  - Network with 160 hosts*: 256 addresses are reserved (for future expansions)
- Remote destinations through R4 (Area 2):
  - Network with 248 hosts: 256 addresses are reserved
  - Network with 127 hosts: 32 addresses are reserved
- networks directly connected to R2:
  - Network with 120 hosts: 128 addresses are reserved
  - Point-to-point network (R1-R2): 4 addresses are reserved
  - Point-to-point network (R2-R3): 4 addresses are reserved

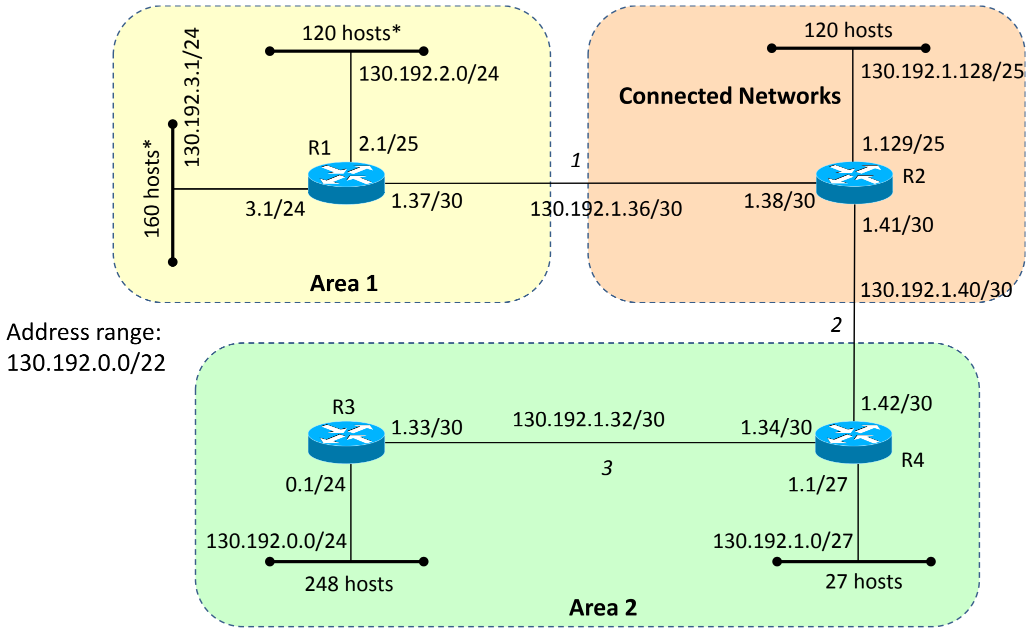
Trying to assign addresses in a way that facilitate the route aggregation on R2, we can proceed as follow:

- Network in Area 1: addressing space /23, completely used
- Network in Area 2: addressing space /23, not completely used (512 - 256 - 32 - 4 = 220 are still free)
- Directly connected networks: 128+4+4 = 136 addresses are necessary. They can be obtained from the /23 addressing space free addresses in the Area 2.

Next figure shows the allocation of the addresses in the assigned addressing space 130.192.0.0/22, with reference to the belonging area (seen from R2):



The resulting addressing plan is the following:



The routing tables, obtained by maximization of routes aggregation, consequently are:

R1

Type	Destination network	Next hop	Cost

C	130.192.1.36/30	130.192.1.37	0
C	130.192.2.0/24	130.192.2.1	0
C	130.192.3.0/24	130.192.3.1	0
S	130.192.0.0/23	130.192.1.38	2

R2

Type	Destination network	Next hop	Cost
C	130.192.1.36/30	130.192.1.38	0
C	130.192.1.40/30	130.192.1.41	0
C	130.192.1.128/25	130.192.1.129	0
S	130.192.0.0/23	130.192.1.42	2
S	130.192.2.0/23	130.192.1.37	1

R3

Type	Destination network	Next hop	Cost
C	130.192.1.32/30	130.192.1.33	0
C	130.192.0.0/24	130.192.0.1	0
S	130.192.0.0/22	130.192.1.34	3

R4

Type	Destination network	Next hop	Cost
C	130.192.1.32/30	130.192.1.34	0
C	130.192.1.40/30	130.192.1.42	0
C	130.192.1.0/27	130.192.1.1	0
S	130.192.0.0/24	130.192.1.33	3
S	130.192.0.0/22	130.192.1.41	2

3.11. Exercise 11

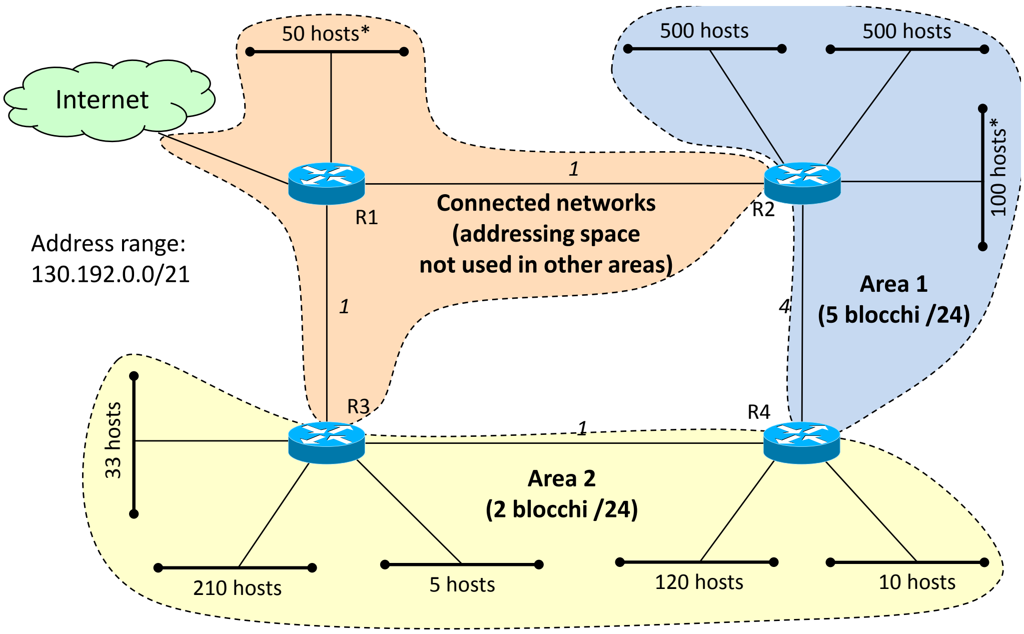
To determine the addressing plan, one has to compute the addresses required to handle each network considering that the router R1 will see each remote destination through two next hops:

- Remote destinations through R2 (Area 1):
  - Network with 500 hosts: 512 required addresses
  - Network with 500 hosts: 512 required addresses
  - Network with 100 hosts*: 256 required addresses (foreseen expansions)
  - Point-to-point network (R2-R4): 4 required addresses
- Remote destinations through R3 (Area 2):
  - Network with 33 hosts: 64 required addresses
  - Network with 210 hosts: 256 required addresses
  - Network with 5 hosts: 8 required addresses
  - Network with 120 hosts: 128 required addresses
  - Network with 10 hosts: 16 required addresses
  - Network with 500 hosts: 512 required addresses
  - Point-to-point network (R3-R4): 4 required addresses
- Networks directly connected to R1:
  - Network with 50 hosts*: 128 required addresses (foreseen expansions)
  - Point-to-point network (R1-R2): 4 required addresses
  - Point-to-point network (R1-R3): 4 required addresses
  - Point-to-point network (R1-Internet): 4 required addresses

It is obvious from that analysis that the number of requested addresses is enough given the assigned addressing space (1284 addresses in Area 1, 476 in Area 2 and 140 for the directly connected networks). Nevertheless, contrary to the previous exercise, the addressing space is not easy to partition into distinct address ranges for the two areas[1].

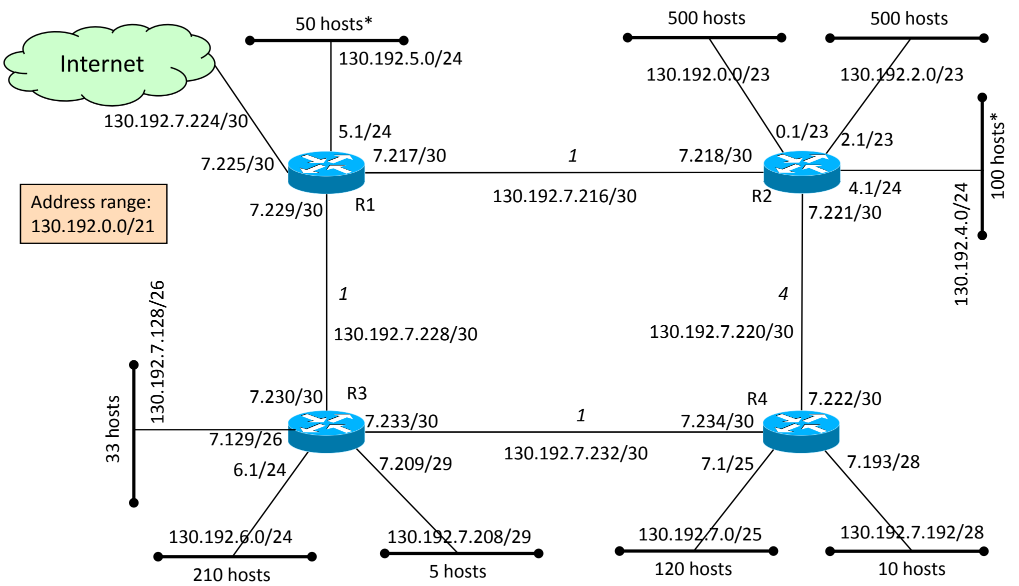
The following figure shows the distribution of the addressing spaces allocated inside the assigned address range 130.192.0.0/21, with reference to the areas they belong to (seen from R1):

[1] Remember that the addresses related to directly connected networks never are aggregatable and thus always appear as "explicit" networks inside the routing table. This nevertheless makes possible, when the supernets are allowed (which is the case here), that these addresses are obtained inside an addressing space unused by the other areas, because they are reachable through a more specific route.



A possible solution of this exercise is to handle the aggregation using supernets: in this case one can imagine to address all the networks in Area 1 with the address 130.192.0.0/21, using the the more specific route 130.192.6.0/23 for all the destinations in Area 2. Obviously this is possible if the addresses assigned in Area 2 can be aggregated (that is to say if the addresses 130.192.6.0/24 and 130.192.7.0/24 and not, for instance, 130.192.5.0/24 and 130.192.6.0/24 that cannot be aggregated together).

The resulting addressing level may thus be:



The routing tables, derived by maximizing the route aggregation (remember the use of the default route to reach all destinations on the Internet), are consequently:

R1

Type	Destination network	Next hop	Cost
C	130.192.7.216/30	130.192.7.217	0
C	130.192.7.224/30	130.192.7.225	0
C	130.192.7.228/30	130.192.7.229	0
C	130.192.5.0/24	130.192.5.1	0
S	130.192.0.0/21	130.192.7.218	1
S	130.192.6.0/23	130.192.7.230	2
S	130.192.7.220/30	130.192.7.218	1
S	0.0.0.0/0	130.192.7.224	2

Be careful about the point-to-point network between R2 and R4: the best path is through the router R2 but unfortunately it falls in the addressing space of the more specific route 130.192.6.0/23 that points in the wrong direction. That is why in this case it is required to address it alone with a more specific route.

R2

Type	Destination network	Next hop	Cost
C	130.192.7.216/30	130.192.7.218	0
C	130.192.7.220/30	130.192.7.221	0
C	130.192.0.0/23	130.192.0.1	0
C	130.192.2.0/23	130.192.2.1	0
C	130.192.4.0/24	130.192.4.1	0
S	0.0.0.0/0	130.192.7.217	2

The routing table of R2 is much more compressed than before because, due to the costs if the links in the studied topology, all destinations not directly connected are reached through the router R1.

R3

Type	Destination network	Next hop	Cost
C	130.192.7.228/30	130.192.7.229	0
C	130.192.7.232/30	130.192.7.233	0
C	130.192.7.208/29	130.192.7.209	0
C	130.192.7.128/26	130.192.7.129	0
C	130.192.6.0/24	130.192.6.1	0
S	130.192.7.0/25	130.192.7.234	1
S	130.192.7.192/28	130.192.7.234	1
S	130.192.7.220/30	130.192.7.234	1
S	0.0.0.0/0	130.192.7.229	2

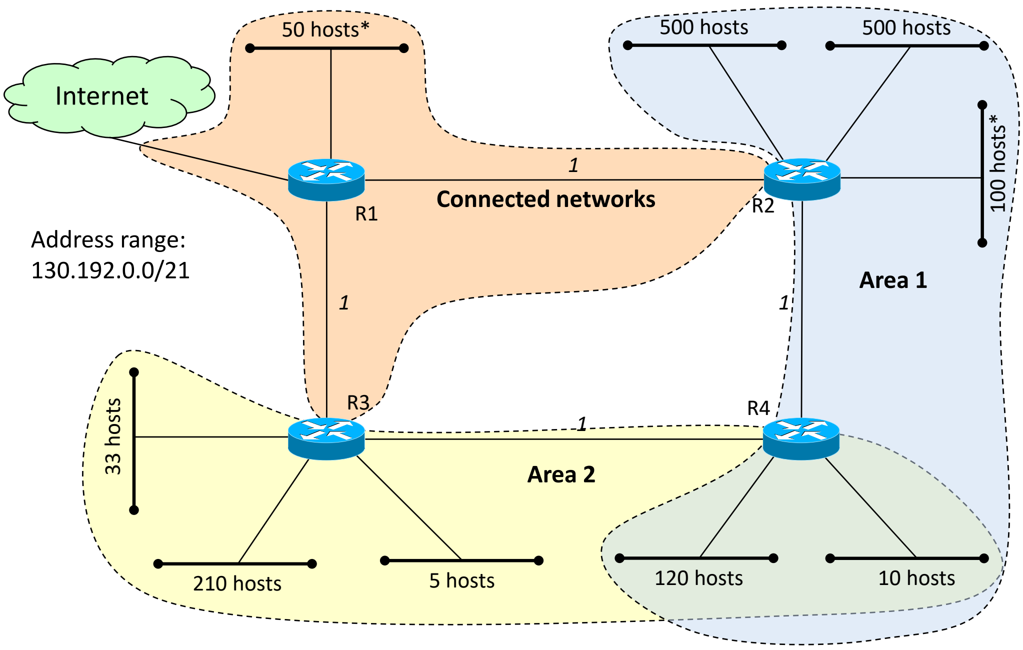
R4

Type	Destination network	Next hop	Cost
C	130.192.7.220/30	130.192.7.222	0
C	130.192.7.232/30	130.192.7.234	0
C	130.192.7.0/25	130.192.7.1	0
C	130.192.7.192/28	130.192.7.193	0
S	0.0.0.0/0	130.192.7.233	2

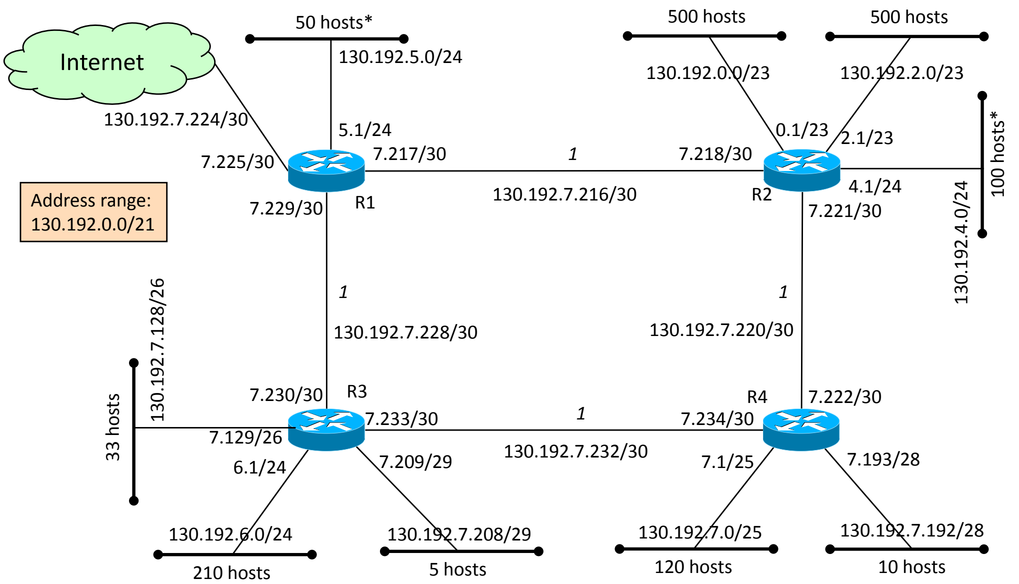
3.12. Exercise 12

The exercise is clearly similar to the previous one. The difference is to be found in the fact that, by changing the weights of the links, the best paths to reach the various destinations may be different compared to the previous case.

Particularly, concerning the LAN connected to R4, there exist two equivalent ways that can be used to best the aggregation from R1, as shown in next figure:



Unfortunately, this peculiar displacement of the IP networks does not allow a better aggregation compared to the one pointed out in the previous exercise and thus the addressing plan will be identical to the previous one:



In consequence, the routing tables, obtained by maximizing the route aggregation, are the following. Notice that in mean they are bigger that the ones of the previous exercise because there is no longer a link R2-R4 with a huge cost that, actually, prevented the emission of packets in that direction which allowed to aggregate a bigger number of destinations in the routes that were going the other ways.

R1

Type	Destination network	Next hop	Cost
C	130.192.7.216/30	130.192.7.217	0
C	130.192.7.224/30	130.192.7.225	0
C	130.192.7.228/30	130.192.7.229	0
C	130.192.5.0/24	130.192.5.1	0
S	130.192.0.0/21	130.192.7.218	2
S	130.192.6.0/23	130.192.7.230	2
S	130.192.7.220/30	130.192.7.218	1
S	0.0.0.0/0	130.192.7.224	2

R2

Type	Destination network	Next hop	Cost
C	130.192.7.216/30	130.192.7.218	0
C	130.192.7.220/30	130.192.7.221	0
C	130.192.0.0/23	130.192.0.1	0
C	130.192.2.0/23	130.192.2.1	0
C	130.192.4.0/24	130.192.4.1	0
S	130.192.7.232/30	130.192.7.222	1
S	130.192.7.192/28	130.192.7.222	1
S	130.192.7.0/25	130.192.7.222	1
S	0.0.0.0/0	130.192.7.217	2

R3

Type	Destination network	Next hop	Cost
C	130.192.7.228/30	130.192.7.229	0
C	130.192.7.232/30	130.192.7.233	0
C	130.192.7.208/29	130.192.7.209	0
C	130.192.7.128/26	130.192.7.129	0
C	130.192.6.0/24	130.192.6.1	0
S	130.192.7.0/25	130.192.7.234	1
S	130.192.7.192/28	130.192.7.234	1
S	130.192.7.220/30	130.192.7.234	1
S	0.0.0.0/0	130.192.7.229	2

R4

Type	Destination network	Next hop	Cost
C	130.192.7.220/30	130.192.7.222	0
C	130.192.7.232/30	130.192.7.234	0
C	130.192.7.0/25	130.192.7.1	0
C	130.192.7.192/28	130.192.7.193	0
S	130.192.7.216/30	130.192.7.221	2
S	130.192.0.0/22	130.192.7.221	1
S	130.192.4.0/24	130.192.7.221	1
S	0.0.0.0/0	130.192.7.233	2

3.13. Exercise 13

The solution proposed by the network manager raises a lot of problems.

Default route with cost 10 on the router R1

Probably the manager has increased the cost of the default route in order to make it chosen only in "last resort" when all other routes are not fit. Nevertheless, such a huge cost is absolutely useless because the cost of a route is a choice parameter only in presence of two routes that point to the same destination. When the routes point to different destinations (for example because they refer to diverse addressing spaces), the most specific wins.

Routes with an higher cost for the networks 7.0/25 and 7.192/28

The intent of the network manager was probably to define some backup routes for the networks 7.0/25 and 7.192/28 when the aggregated route 6.0/23 is unavailable.

Nevertheless, for the same reason pointed out in the previous point, these routes will *always* be chosen as the path to follow to reach the studied destinations. That is why, if one really wants to create backup routes, the configuration of the router R1 should be at least modified in a way that allows to define explicit routes for the networks 130.192.7.0/25 and 130.192.7.192/28, in a way such that these destinations will not be routed through the aggregated route 130.192.6.0/23, thus abound these two routes with likewise routes that should be used in case of backup.

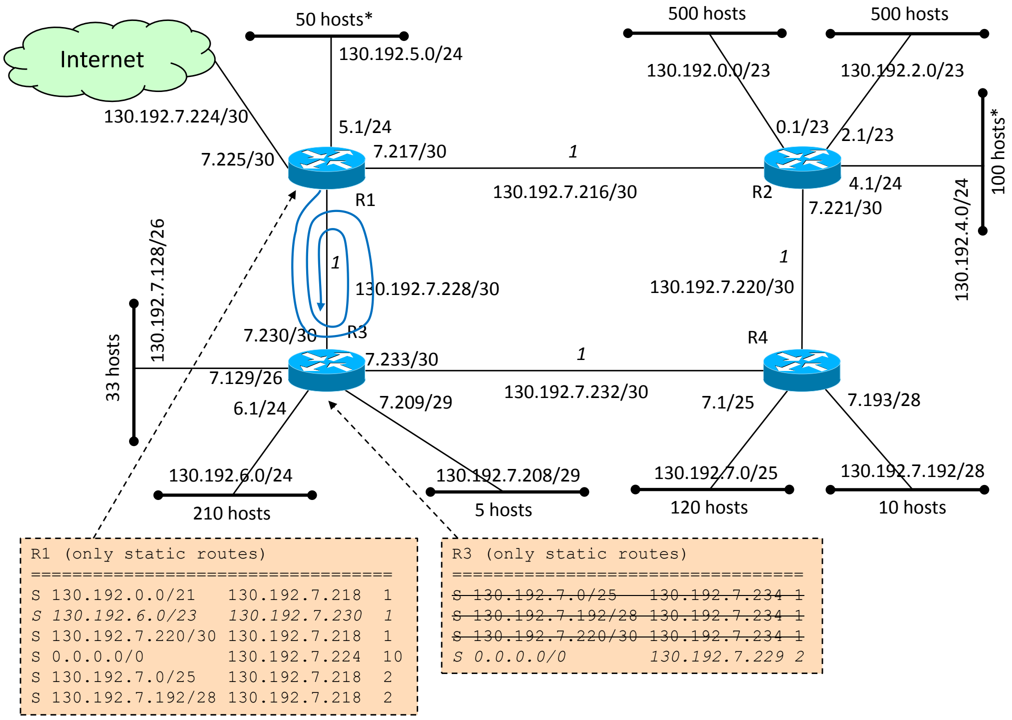
The routing table of R1 would become (omitting the direct routes):

Type	Destination network	Next hop	Cost
S	130.192.0.0/21	130.192.7.218	1
S	130.192.6.0/23	130.192.7.230	1
S	130.192.7.220/30	130.192.7.218	1
S	*130.192.7.0/25*	*130.192.7.230*	*1*
S	*130.192.7.192/28*	*130.192.7.230*	*1*
S	*130.192.7.0/25*	*130.192.7.218*	*2*
S	*130.192.7.192/28*	*130.192.7.218*	*2*
S	0.0.0.0/0	130.192.7.224	2

Robustness to the failure of R3-R4

The biggest problem of the solution is yet the fact that the "backup" routes for the networks 7.0/25 and 7.192/28 are not absolutely efficient because R1 is not absolutely able to react correctly to the failure of the link R3-R4. Actually, even if R3 realizes that there is a failure and that its routing table becomes the one indicated in next figure (all the routes that depend on the next hop 130.192.7.234 become invalid and this all the traffic heading to networks not directly connected would follow the default route that is the only one remaining), no one communicates to R1 the existence of this failure so the router can not know that it is required to use the route with higher cost to reach these destinations.

The result, shown in figure, would be that at reception of a packet destined to an host in one of these two networks the router R1 would continue to use the route with cost 1 to R3 but R3 would use the default route and send the packet back to R1 (the used routes are indicated in figure in *italic*). The packet would enter a loop and stay in it, going back and forth between R1 and R3, until its lifespan ends (the value of the field TTL in the IP header would become zero).



Unfortunately there exists no solution to this problem because it is intrinsic to the use of backup routes with static routing. So it is not possible to propose better solutions to the manager, except to use a dynamic routing protocol inside his network.

3.14. Exercise 14

3.14.1. Case 1

The router R2 detects that the packet must follow the default route and thus must be sent to the next hop 192.168.0.1. This address is directly reachable through its rightmost interface; besides the associated MAC address (address to which the frame at data-link level must be sent) is already known because it is in the ARP cache.

So the router R2 will send the packet to the router R1, however inserting the address 00:00:00:DD:DD:DD as destination Ethernet address. This frame will however be discarded because the MAC address present does not coincide with the one present in the interface.

Consequently, the packet generated by H2 will never reach destination and will be discarded by R1.

3.14.2. Case 2

The router R2 detect that the packet should follow the default route and thus must be sent to the next hop 192.168.1.1. Nevertheless, this address is not part if the IP networks directly connected to R2 and thus R2 has no idea of how to forward the packet to the next hop indicated in the routing table.

The fact that the MAC address related to this IP address is present in the ARP cache is no help because the router R2 will abort the sending of the packet before consulting the ARP cache. This table is actually consulted when it wants to discover the MAC address of a directly reachable host, condition that is not valid for the address 192.168.1.1.

Consequently, in this case again the packet generated by H2 will never reach destination and it will be discarded by R2.

3.14.3. Case 3

The router R2 detects that the packet should follow the default route and thus must be sent to the next hop 192.168.4.1. This address is directly reachable through its Ethernet interface with address 192.168.4.254; moreover its associated MAC address (address to which the frame at level data-link must be sent) is already known because it is in the ARP cache.

So the router R2 will forward the packet to LAN1, inserting the address 00:00:00:DD:DD:DD as destination Ethernet address. This frame will then be received by the router R2 itself but in the interface 192.168.3.254, because the present destination MAC address coincide with the one present in the interface[2].

At this point, the packet will be examined by R2 again that will retake the same decision already applied previously. Consequently, in this case again the packet generated by H2 will never reach destination but will cycle on the LAN1 until its lifespan (IP TTL) will end and will finally be discarded by R2.

[2]Notice that the next hop address contained in the routing table (that is 192.168.4.1) is absolutely unlinked to the effective address of the interface that receive that packet in the LAN (that is 192.168.3.254).

3.14.4. LAN 1 implemented as switched LAN

The technology used to implement LAN1 ("shared" or "switched") has no influence in the solution of the exercise.