Les réseaux permettent de connecter différentes machines entre elles. Dans ce chapitre, on s'intéresse aux **protocoles de routage** permettant de déterminer le moyen que l'on a de faire transiter une information d'une machine émettrice vers une machine destinatrice.

1 Vocabulaire, adresse IP

Un **réseau** est un ensemble de **machines** (ordinateurs personels, serveurs, imprimantes, etc.) pouvant communiquer les unes avec les autres en s'envoyant **des paquets** (des bouts de messages découpés). Afin de connecter entre elles ces machines, plusieurs solutions sont possibles :

— chaque machine est connectée directement via une certaine **interface** (ethernet, wi-fi, etc.) à chaque autre machine du réseau, on parle alors de **réseau local** ;

— en passant par un sous-réseau voisin, à l'aide d'un **routeur** servant de **passerelle**.

Exemples de réseaux locaux : le réseau du lycée, un réseau domestique, le réseau d'une petite entreprise.

Exemples de connexion entre plusieurs réseaux : votre box internet sert de routeur connectant votre réseau domestique au réseau internet, des routeurs permettent de connecter le réseaux créé par une antenne 4G à internet, internet est lui même constitué d'une multitudes de réseaux locaux interconnectés (inter-network).

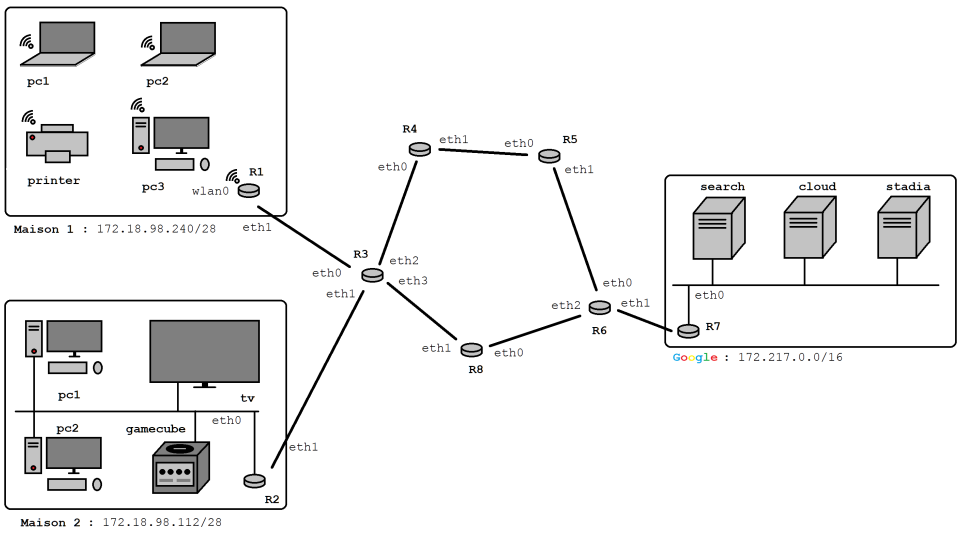
Les machines constituant un réseau sont de différentes natures. On distingue :

— les **serveurs** proposant des services aux clients ;

— les **clients** utilisant les serveurs ;

— les **routeurs** faisant le lien entre différent sous-réseaux et assurant la bonne diffusion des informations au sein du réseau.

Voyons un exemple de réseau :



Les machines client de ce réseau sont les ordinateurs, l'imprimante, la gamecube, la télé.

Les machines serveur sont `search`, `cloud` et `stadia`.

Les routeurs sont `R1`, ..., `R8`.

Les réseaux locaux sont `Maison 1`, `Maison 2`, `Google` mais aussi les liens entre les différents routeurs, par exemple entre `R1` et `R3`, entre `R6` et `R8` etc.

Les différentes machines de `Maison 1` sont connectées entre elles via une interface wi-fi. Celles de `Maison 2` par une interface filaire (par exemple ethernet).

Le routeur `R1` servira de passerelle entre `Maison 1` et les autres sous-réseaux.

La première chose à considérer lorsque l'on veut envoyer un paquet, c'est un moyen de retrouver le bon destinataire : sur un réseau, chaque machine est identifiée par une adresse unique, appelée **adresse IP**. Dans la version 4 (ipv4), celle-ci est constituée de 32 bits, généralement écrits sous la forme de 4 nombres compris entre 0 et 255.

Exemple : 192.58.12.13

Une adresse IP se décompose en deux parties de tailles variables :

— une première partie identifiant l'adresse du sous-réseau dans lequel se situe la machine ;

— une seconde identifiant l'adresse de la machine au sein de son sous-réseau.

Pour préciser cela, on utilise un **masque de sous-réseau** qui peut être donné sous la même forme ou sous la forme d'un entier compris entre 0 et 32. Ce masque permet d'identifier des sous réseaux dans lesquels l'IP de toutes les machines commenceront par les mêmes bits.

Un masque, écrit en binaire, doit être de la forme 1...10...0. Les 1 correspondant aux positions des bits communs à toutes les adresses disponibles du sous réseau et les 0 aux positions des bits définissant l'adresse spécifique de la machine sur le sous-réseau.

Reprenons notre réseau exemple et attribuons des IP fictives aux différentes machines :

Machine	**ip**
Maison 1	172.18.98.240/28
pc1	172.18.98.241/28
pc2	172.18.98.242/28
pc3	172.18.98.243/28
printer	172.18.98.244/28
Maison 2	172.18.98.112/28
pc1	172.18.98.113/28
pc2	172.18.98.114/28
gamecube2	172.18.98.115/28
tv	172.18.98.116/28
Google	172.217.0.0/16
search	172.217.0.1/16
cloud	172.217.0.14/16
stadia	172.217.255.112/16

Sous-réseau	**ip**
R1-R3	10.1.1.0/30
R2-R3	10.1.2.0/30
R3-R4	10.2.1.0/30
R3-R8	10.2.2.0/30
R4-R5	10.3.1.0/30
R8-R6	10.4.1.0/30
R5-R6	10.4.2.0/30
R6-R7	10.5.1.0/30

Remarque : Les routeurs étant toujours présents sur plusieurs sous-réseaux à la fois, ils ont plusieurs adresses ip (une par sous-réseau connecté). En fait, une adresse ip n'est pas attribuée à une machine mais à une carte réseau et les routeurs en ont par conséquent plusieurs.

Analysons l'ip de `pc1` dans le sous-réseau `Maison 1` :

$$172.18.98.241/28$$

Appliquons tout d'abord le masque de sous-réseau. Pour cela, il faut écrire l'ip en binaire :

$$10101100.00010010.01100010.11110001$$

Le masque de sous réseau nous informe que cette ip se décompose en une partie réseau de 28 bits et une partie machine sur $32 - 28 = 4$ bits.

$$10101100.00010010.01100010.11110001$$

Si on remplace, les bits de la partie machine par des 0, on trouve l'adresse du sous-réseau dans laquelle elle se trouve :

$$10101100.00010010.01100010.11110000$$
$$= 172.18.98.240$$

On retrouve bien l'adresse ip de `Maison 1`.

De plus, on a une information importante sur le sous-réseau `Maison 1` : puisque la partie machine des adresses ip est de taille 4, on dispose à l'intérieur du sous-réseau de 2^4 adresses différentes.

Cependant, toutes ces adresses ne sont pas disponibles pour des machines. En effet, deux adresses sont toujours attribuées par défaut dans un sous-réseau :

— La première (partie machine 0...0) correspond à **l'adresse du sous-réseau** comme on l'a vu sur `pc1` et `Maison 1`.

— La dernière (partie machine 1...1) correspond à **l'adresse de diffusion** (ou *broadcast*) utilisée lorsqu'on veut atteindre sans distinction toutes les machines du sous-réseau.

De manière générale, pour un sous-réseau de masque k on peut donc connecter $2^{32-k} - 2$ machines différentes.

Donner l'adresse du sous réseau dans lequel ce trouve la machine d'ip

$$172.18.98.67/29$$

Combien de machines peut-on connecter sur ce sous-réseau ?

On veut envoyer un paquet à toutes les machines de ce sous-réseau. Quelle adresse ip doit-on utiliser ?

On s'intéresse dans ce chapitre aux routeurs. On en distingue deux types :

— **les routeurs d'accès**, connecté à des clients ou à des serveurs et servant à envoyer ou recevoir des paquets à partir d'un réseau local ;

— **les routeurs internes**, n'étant connectés qu'à d'autres routeurs et servant à faire transiter ces paquets.

L'ensembles des liens (en fait des sous-réseaux) entre les différents routeurs d'un réseau est appelé **la topologie du réseau**. C'est une donnée fondamentale pour savoir comment faire transiter les paquets.

2 Table de routage

On considère un réseau composés de plusieurs sous-réseaux reliés par des routeurs. Lorsqu'un paquet doit être transmis d'une machine à une autre, il y a deux possibilités :

— Si les deux machines sont dans un même réseau local, le paquet peut être transmis directement par l'interface commune.

— Sinon, le paquet est transmis à un sous réseau adjacent via un des routeurs (la passerelle).

Ainsi, le paquet circule de sous-réseaux en sous-réseaux et finit (on l'espère !) par arriver à destination. Se pose alors la question suivante : comment un routeur décide du sous réseaux auquel il doit transmettre l'information ?

Pour décider de la prochaine destination d'un paquet, chaque routeur garde en mémoire une **table de routage**, c'est à dire un tableau associant à chaque destination connue le sous-réseau voisin par lequel l'information va devoir transiter.

Reprenons notre exemple et donnons des tables de routages possibles pour chaque routeur :

Table de routage de R1				
Destination	**ip**	**Passerelle**	**ip**	**Interface**
Google	172.217.0.0/16	R3	10.1.1.2/30	eth1
Maison1	172.18.98.240/28	*	*	wlan0
Maison2	172.18.98.112/28	R3	10.1.1.2/30	eth1

Table de routage de R2				
Destination	**ip**	**Passerelle**	**ip**	**Interface**
Google	172.217.0.0/16	R3	10.1.2.2/30	eth1
Maison1	172.18.98.240/28	R3	10.1.2.2/30	eth1
Maison2	172.18.98.112/28	*	*	eth0

Table de routage de R3				
Destination	**ip**	**Passerelle**	**ip**	**Interface**
Google	172.217.0.0/16	R8	10.2.2.2/30	eth3
Maison1	172.18.98.240/28	R1	10.1.1.1/30	eth0
Maison2	172.18.98.112/28	R2	10.1.2.1/30	eth1

Table de routage de R4				
Destination	**ip**	**Passerelle**	**ip**	**Interface**
Google	172.217.0.0/16	R5	10.3.1.2/30	eth1
Maison1	172.18.98.240/28	R3	10.2.1.1/30	eth0
Maison2	172.18.98.112/28	R3	10.2.1.1/30	eth0

Table de routage de R5				
Destination	**ip**	**Passerelle**	**ip**	**Interface**
Google	172.217.0.0/16	R6	10.4.2.2/30	eth1
Maison1	172.18.98.240/28	R4	10.3.1.1/30	eth0
Maison2	172.18.98.112/28	R4	10.3.1.1/30	eth0

Table de routage de R6				
Destination	**ip**	**Passerelle**	**ip**	**Interface**
Google	172.217.0.0/16	R7	10.5.1.2/30	eth1
Maison1	172.18.98.240/28	R8	10.4.1.2/30	eth2
Maison2	172.18.98.112/28	R8	10.4.1.2/30	eth2

Table de routage de R7				
Destination	**ip**	**Passerelle**	**ip**	**Interface**
Google	172.217.0.0/16	*	*	eth0
Maison1	172.18.98.240/28	R6	10.5.1.1/30	eth1
Maison2	172.18.98.112/28	R6	10.5.1.1/30	eth1

Table de routage de R8				
Destination	**ip**	**Passerelle**	**ip**	**Interface**
Google	172.217.0.0/16	R6	10.4.1.1/30	eth0
Maison1	172.18.98.240/28	R3	10.2.2.1/30	eth1
Maison2	172.18.98.112/28	R3	10.2.2.1/30	eth1

En suivant les informations de la table de routage, décrire la route d'un paquet envoyé du `pc1` de `Maison 1` à destination de `stadia`.

Définir le plus court chemin d'un point à un autre dans un réseau n'est pas quelque chose d'évident. Prenons un exemple en dehors de ce contexte pour nous en rendre compte : celui d'un GPS devant déterminer un itinéraire entre deux localisations.

Usuellement, il est possible de choisir dans les paramètres du GPS la manière dont celui ci calcule l'itinéraire. Cela peut être :

— le trajet le plus court (en kilomètre) ;

— le trajet le plus rapide (en heures).

Mais on pourrait aussi en imaginer d'autres :

— le trajet le moins polluant (en kg de CO2 émis) ;

— le trajet le moins cher (en €).

De plus, il est possible que certaines routes, pourtant présentes sur le réseau routier, soient momentanément indisponibles (travaux, accidents, embouteillages). Les GPS peuvent prendre ces informations en compte en temps réel.

Apparaît dans cet exemple une notion fondamentale en réseau : la métrique. Il s'agît, étant donné un ensemble de positions, d'une mesure de la distance séparant chaque couple de points.

Pour reprendre l'exemple précédent, on a mis en évidence quatre métriques pertinentes pouvant être utilisées par un GPS :

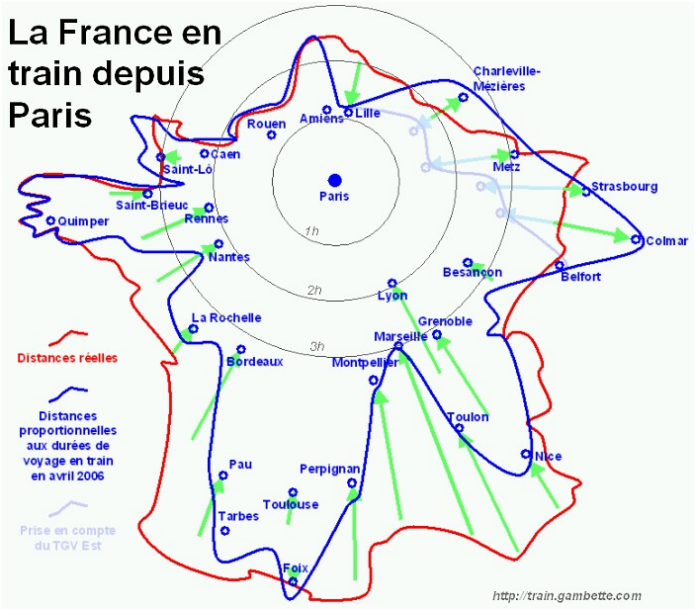
— la longueur ;

— le temps ;

— l'émission de CO_2 ;

— le coût.

En fonction de la métrique choisie le plus court chemin entre deux points, c'est à dire le chemin minimisant la distance parcourue **du point de vue de la métrique**, n'est pas forcément le même.



Mettons en évidence une autre difficulté dans l'élaboration des tables de routages par une différence majeure entre le GPS et les résaux.

Une métrique étant choisie, le GPS détermine le plus court chemin entre deux points, mais comment s'y prend-il ? Le GPS dispose d'une carte routière, contenant plus ou moins d'informations, qu'il analyse via des **algorithmes de parcours de graphe** pour obtenir ce chemin.

C'est un confort que les routeurs n'ont pas forcément. Ils fonctionnent en effet, pour des raisons de maintenabilité des réseaux, **de manière décentralisé** : un routeur n'a accès qu'à ses propres informations et éventuellement aux informations de ses voisins directs. En particulier et contrairement à un GPS, il n'a pas de carte complète du réseau sur lequel exécuter un algorithme de plus court chemin.

Chaque routeur doit donc adopter un comportement individuel permettant, *in fine*, l'emprunt par les données du plus court chemin selon la métrique choisie.

La métrique choisie et les algorithmes utilisés par les routeurs leur permettant de construire et de mettre à jour leur table de routage sont des informations définissant le **protocole de routage** du réseau.

Nous allons en détailler deux : le protocole RIP (*Routing Information Protocol*) et le protocole OSPF (*Open Shortest Path First*).

3 Protocole RIP

Dans le protocole RIP, la métrique utilisée est le nombre de sauts minimum entre deux sous réseaux. Sans détailler comment on a fait pour obtenir ces distances, les voici sur notre exemple de réseau les distances entre les différents routeurs :

	R1	R2	R3	R4	R5	R6	R7	R8
R1	0	2	1	2	3	3	4	2
R2		0	1	2	3	3	4	2
R3			0	1	2	2	3	1
R4				0	1	2	3	2
R5					0	1	2	3
R6						0	1	1
R7							0	2
R8								0

Dans le protocole RIP, une table de routage contient les informations suivantes :

— les destinations connues ;

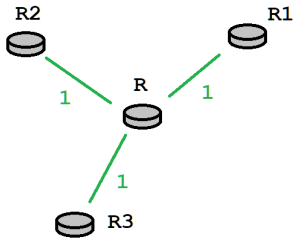
— les passerelles à emprunter pour se rendre à ces destination ;

— les interfaces à utiliser pour joindre les passerelles ;

— les distances (en nombre de sous-réseaux traversés) jusqu'aux destinations.

Les destinations et les distances sont donnés sous forme d'adresses ip. Afin de simplifier les notations, on donnera plutôt les noms des machines (uniques dans l'exemple). De plus, on ne précisera pas l'interface.

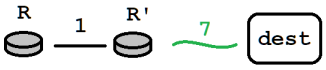
Afin de construire ces tables de routage, le protocole spécifie que les routeurs procèdent de la manière suivante :
On commence par une phase d'initialisation dans laquelle chaque routeur identifie ces voisins immédiats (qui ont un sous-réseau communs avec lui). Ces voisins sont à une distance de 1. Chaque routeur dispose donc initialement d'une liste restreinte de **sous-réseaux atteignables** ainsi que sa distance à ses sous réseaux. Cette liste appelée **vecteur distance** va s'étoffer par la suite.



Destination	Passerelle	Dist.
$R1$	*	1
$R2$	*	1
$R3$	*	1

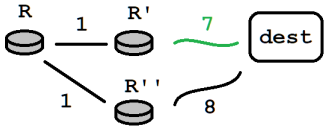
Pour poursuivre le protocole, un routeur – disons R – envoie des **demandes RIP** à ses voisins. Lorsqu'un routeur voisin – disons R' – reçoit la demande RIP de R, il y répond en envoyant un accusé de réception contenant la liste des réseaux qu'il peut atteindre ainsi que leurs distance. Cette liste étant maintenant connue de R, celui-ci peut mettre à jour sa table :

— Si une nouvelle destination est atteignable par R', il la rajoute à sa table en précisant que sa passerelle sera R' et que sa distance sera celle de R' plus un.



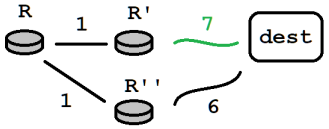
Destination	Passerelle	Dist.
R'	*	1
dest	R'	8

— Si une destination connue (via R' ou un autre voisin R'') est atteignable par R' via une route plus courte, il modifie sa table comme précédemment.



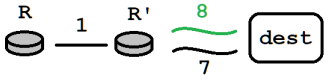
Destination	Passerelle	Dist.
R'	*	1
R''	*	1
dest	R''	9
dest	R'	8

— Si une destination connue est atteignable par R' via une route plus longue, il y a deux possibilités :
  — si l'ancienne route ne passait pas par R' mais par un autre voisin R'', alors la route est moins efficace : il l'ignore ;



Destination	Passerelle	Dist.
R'	*	1
R''	*	1
dest	R''	7

— — si l'ancienne route passait déjà par R', cela signifie qu'il y a eu un problème avec cette route : il la met à jour comme précédemment.



Destination	Passerelle	Dist.
R'	*	1
R''	*	1
dest	R'	8
dest	R'	9

C'est en appliquant ce protocole que l'on a obtenu la table de routage donnée plus haut.

Détection des pannes : Si un routeur ne reçoit pas de réponse à une demande RIP (par défaut au bout de 3 minutes), il considère que le routeur auquel il l'a envoyé est en panne. Dans ce cas, il lui attribue une distance de 16 qui est interprétée dans le protocole comme une distance infinie.

Cela signifie en particulier que les réseaux dans lesquels on utilise RIP ne peuvent pas être de trop grande taille. **Leur diamètre** (distance maximale entre deux points du réseau) **ne peut excéder 15**.

Cela signifie aussi que les routeurs auront une vision cohérente du réseau une fois que tous les routeurs auront effectué 15 demandes à tous leurs voisins. On parle de **délai de convergence**.

4 Protocole OSPF

Contrairement au protocole RIP, le protocole OSPF est adapté aux plus grands réseaux. Ces principales caractéristiques sont les suivantes :

— Il prend en compte la **bande passante** (c'est à dire le débit maximum des transferts de données) des différentes liaisons de communication des réseaux traversés. Il favorisera donc les trajets **les plus rapides**.

— Il est organisé **par zones** (ensembles de sous-réseaux), desquelles leurs routeurs ont une vision d'ensemble de la topologie. Les différentes zones étant reliées entre elles par une zone centrale : la **backbone**

Définissons dans un premier temps la métrique du protocole :

Chaque sous-réseau liant entre eux deux routeurs dispose d'une **bande passante** représentant la quantité maximale de données pouvant être échangées par unité de temps. On l'exprime en **bits par seconde** notés bit/s, $bit.s^{-1}$ ou encore bps.

La bande passante d'une liaison dépend majoritairement de la technologie utilisée. Voici un tableau donnant les bandes passantes des technologies usuelles (on distingue les BP montantes, de l'utilisateur au fournisseur d'accès, des BP descendante, du fournisseur à l'utilisateur) :

Technologie	**BP descendante**	**BP montante**
Modem	56 kbit/s	48 kbit/s
Bluetooth	3 Mbit/s	
Ethernet	10 Mbit/s	
Wi-Fi	11 Mbit/s à 10 Gbit/s	
ADSL	13 Mbit/s	1 Mbit/s
4G	100 Mbit/s	50 Mbit/s
Satellite	50 Mbit/s	1 Mbit/s
FastEthernet	100 Mbit/s	
FFTH (fibre)	10 Gbit/s	

Plus précisément, on associe un coût à chaque liaison :

$$\text{coût} = \lceil 10^8/BP \rceil$$

où $\lceil x \rceil$ désigne la partie entière supérieure de x (on arrondit à l'entier supérieur). Le 10^8 est une valeur par défaut qui peut être modifiée et qui donne un poids de 1 (plus petit poids possible) aux liasons FastEthernet ou plus rapides. En particulier, cette métrique ne distingue pas les liasons FastEthernet des liasons FFTH (fibre).

Exemples :

— une liaison de 100 Mbits/s a un coût de $\lceil 10^8/100.10^6 \rceil = 1$;

— une liaison de 50 Mbits/s a un coût de $\lceil 10^8/50.10^6 \rceil = 2$;

— une liaison de 33 Mbits/s a un coût de $\lceil 10^8/33.10^6 \rceil = 3$;

— etc.

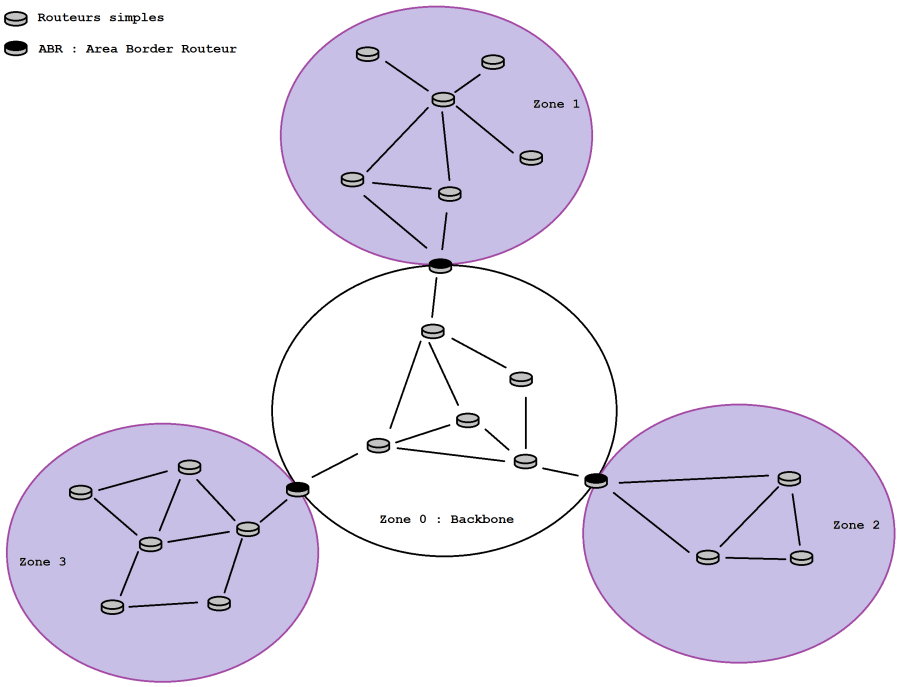
— une liaison de 1 Gbit/s a un coût de $\lceil 10^8/10^9 \rceil = \lceil 0.1 \rceil = 1$

Maintenant que la métrique est définie, détaillons le protocole :

Un réseau dont le protocole est OSPF est organisé en zones (zone 1, zone 2, etc.) qui fonctionnent de manière indépendantes et qui sont liées entre elles par une zone particulière (zone 0) appelées **backbone**.

À l'intérieur de chaque zone, chaque routeur connaît la topologie de la zone, qui comprend l'ensemble des liens entre chaque paire de routeurs de la zone ainsi que leur bande passante.

De plus, dans chaque zone, un routeur spécifique appelé **ABR** (*Area Border Routeur*) est connecté à la fois à sa zone et à la backbone. Par ce routeur transiteront tous les paquets devant entrer ou sortir de la zone.



Phase d'initialisation : À l'intérieur d'une zone donnée, les routeurs commencent par s'échanger des informations sur leurs liaisons. Cela fonctionne comme dans le protocole RIP sauf qu'il n'y a aucun calcul a faire. Le but est simplement de faire en sorte que chaque routeur connaisse la topologie de sa zone.

Par exemple, dans la zone 2 de l'exemple, cela pourrait donner une table des liaisons :

Liaison	BP	Coût
R1-R2	100 Mbit/s	1
R1-R3	10 Mbit/s	10
R2-R3	100 Mbit/s	1
R2-R4	25 Mbit/s	4
R3-R4	100 Mbit/s	1

Routage intra-zone : Une fois cette table connue de tous les routeurs de la zone, chacun applique **l'algorithme de Dijkstra** permettant de transformer le graphe en arbre dont les chemins réalisent toujours les distances les plus courtes de la racine vers les feuilles.

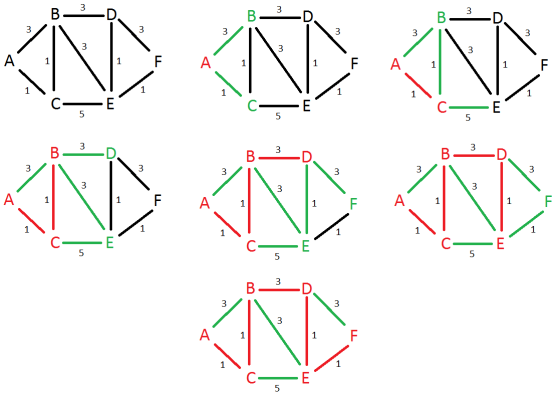
Algorithme de Dijkstra, version visuelle : On va classer les sommets et les liens du graphe en 3 catégories : inaccessibles, accessibles, ou fixés.

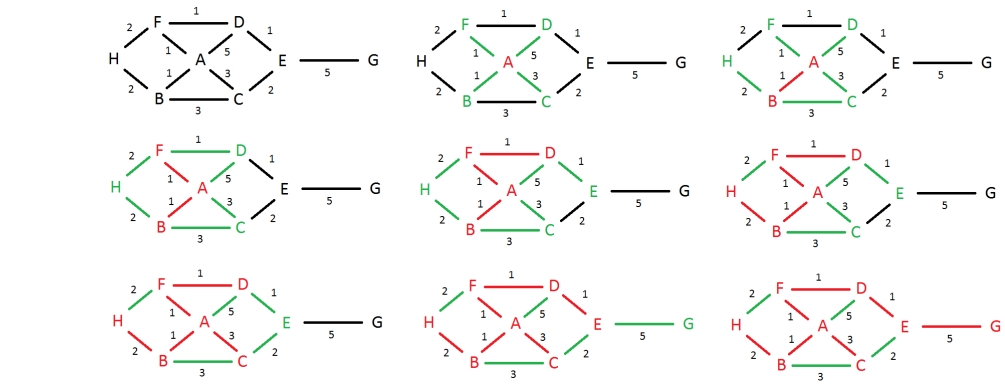
Initialement, tous les sommets sont inaccessibles sauf le sommet de départ qui est fixé.

Tant que tous les sommets ne sont pas fixés :

— Tout sommet voisin ou lien d'un sommet fixé devient accessible.

— Le sommet accessible le plus proche du sommet de départ en ne passant que par des sommets fixés devient fixé ainsi que le lien utilisé (si on a plusieurs choix possibles n'importe lequel peut être choisi).





Ici on a représenté l'arbre obtenu par R3 après application de Dijkstra. En fait vu la topologie, l'arbre est le même quelque soit le routeur. Les routeurs peuvent construire leurs tables de routage :

Table de R1		
Dest.	Inter.	Coût
R2	*	1
R3	R2	2
R4	R2	3

Table de R3		
Dest.	Inter.	Coût
R1	R2	2
R2	*	1
R4	*	1

Table de R2		
Dest.	Inter.	Coût
R1	*	1
R3	*	1
R4	R3	2

Table de R4		
Dest.	Inter.	Coût
R1	R3	3
R2	R3	2
R3	*	1

Routage inter-zone : À ce stade, dans chaque zone (y compris la backbone) chaque routeur dispose de sa table de routage **intra-zone**. Il va chercher à la compléter pour l'ensemble du réseau en passant par les ABR.

Les ABR connaissant les meilleurs chemins de deux zones, ils transmettent l'information aux autres routeurs, qui peuvent alors mettre à jour leurs tables de routage : si un routeur R5 de la backbone apprends que R1 peut joindre R4 avec un coût de 3, il ajoute à sa table de routage la destination R4 suivant la même passerelle que sa route vers R1 et en ajoutant 3 à son coût.

Exercices 181, 182, sujet 0 exercice 5

5 Commandes système

Sur linux, on peut accèder à certaines informations de routage via les commandes `ip`, `ping` et `traceroute`.

La commande `ip` sert a afficher des information reseau concernant la machine :

```
ip addr
    #affiche les interfaces de la machine

ip route
    #affiche la table de routage de la machine
```

```
truc@machine:~ $ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:4f:41:81 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::a00:27ff:fe4f:4181/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast master lxcbr0 state UP group default qlen 1000
```

```
    link/ether 08:00:27:5b:b3:66 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.101/28 brd 192.168.56.255 scope global eth1
       valid_lft forever preferred_lft forever

truc@machine:~ $ ip route
default via 192.168.74.254 dev eno16777736  metric 2
192.168.74.0/24 dev eno16777736  proto kernel  scope link  src 192.168.74.128
```

La commande `ping` envoie des paquets à l'adresse specifiee (sous forme d'ip ou d'URL) et affiche les temps de réponse.

```
ping <ip cible>
    #affiche en continue les temps de reponse de l'adresse

ping <ip cible> -c <nb de tentatives>
    #precise le nombre de paquets a envoyer

ping <ip cible> -t <TTL>
    #precice le TTL (duree de vie du paquet en sauts)
```

Le TTL (*Time To Live*) d'un paquet est sa durée de vie. Il décroît de 1 à chaque saut dans le réseau. Attribuer un TTL aux paquets transmis permet d'assurer qu'aucun paquet circule indéfiniment sur le réseau en cas de routage défaillant.

Remarque : Si le TTL est plus petit que la distance séparant la machine de l'adresse spécifiée (en nombre de sauts), alors la commande donnera le dernier routeur contacté.

```
truc@machine:~ $ ping www.google.fr
PING www.google.fr (216.58.209.227) 56(84) bytes of data.
64 bytes from par10s29-in-f227.1e100.net (216.58.209.227): icmp_seq=1 ttl=109 time=50.10 ms

truc@machine:~ $ ping www.google.fr -t 3
PING www.google.fr (216.58.209.227) 56(84) bytes of data.
From 10.216.10.49 (10.216.10.49) icmp_seq=1 Time to live exceeded
```

La commande `traceroute` sert à afficher la route suivie par un paquet jusque l'adresse spécifié.

```
traceroute <ip_cible>
    #montre le chemin suivi par un paquet envoye vers l'ip

traceroute -n <ip_cible>
    #n'affiche que les ip et pas de noms

sudo traceroute -I <ip_cible>
    #change la nature du paquet pour qu'il soit moins bloque
```

Remarque 1 : Lorsque des astérisques s'affichent, cela signifie qu'un routeur n'a pas répondu et dans ce cas, le paquet est envoyé sur une autre route.

Remarque 2 : Cette commande fonctionne en envoyant des `ping` de TTL croissants. En particulier si un changement a lieu sur le réseau entre deux de ces `ping`, la route donnée ne sera pas juste.

```
truc@machine:~ $ traceroute fr.wikipedia.org
traceroute to rr.knams.wikimedia.org (145.97.39.155), 30 hops max, 38 byte packets
 1  80.67.162.30 (80.67.162.30)  0.341 ms  0.300 ms  0.299 ms
 2  telehouse2-gw.netaktiv.com (80.67.170.1)  5.686 ms  1.656 ms  0.428 ms
 3  giga.gitoyen.net (80.67.168.16)  1.169 ms  0.704 ms  0.563 ms
 4  62.4.73.27 (62.4.73.27)  2.382 ms  1.623 ms  1.297 ms
 5  ge5-2.mpr2.cdg2.fr.above.net (64.125.23.86)  1.196 ms ge9-4.mpr2.cdg2.fr.above.net (64.125.23.102)  1.290 ms ge5-1.
    mpr2.cdg2.fr.above.net (64.125.23.82)  30.297 ms
 6  so-5-0-0.cr1.lhr3.uk.above.net (64.125.23.13)  41.900 ms  9.658 ms  9.118 ms
 7  so-7-0-0.mpr1.ams5.nl.above.net (64.125.27.178)  23.403 ms  23.209 ms  23.703 ms
 8  64.125.27.221.available.above.net (64.125.27.221)  19.149 ms so-0-0-0.mpr3.ams1.nl.above.net (64.125.27.181)  19.378
    ms 64.125.27.221.available.above.net (64.125.27.221)  20.017 ms
 9  PNI.Surfnet.ams1.above.net (82.98.247.2)  16.834 ms  16.384 ms  16.129 ms
10  af-500.xsr01.amsterdam1a.surf.net (145.145.80.9)  21.525 ms 20.645 ms  24.101 ms
11  kncsw001-router.customer.surf.net (145.145.18.158)  20.233 ms 16.868 ms  19.568 ms
12  gi0-24.csw2-knams.wikimedia.org (145.97.32.29)  23.614 ms  23.270 ms  23.574 ms
13  rr.knams.wikimedia.org (145.97.39.155)  23.992 ms  23.050 ms 23.657 ms
```