I. Protocoles de routage

1) Fondamentaux du routage

Dans la partie précédente, nous avons vu la notion de réseau local. Plusieurs réseaux locaux sont reliés entre eux par un **routeur**.

Définition : un routeur peut être considéré _____

_____

_____.

Les routeurs les plus simples permettent de relier ensemble deux réseaux (ils possèdent alors 2 interfaces réseau). Les routeurs performants et 100% dédiés au routage sont capables de relier ensemble une dizaine de réseaux.

À gauche : une box ADSL est un routeur pour particuliers. Les multiples prises Ethernet servent de **switch.**

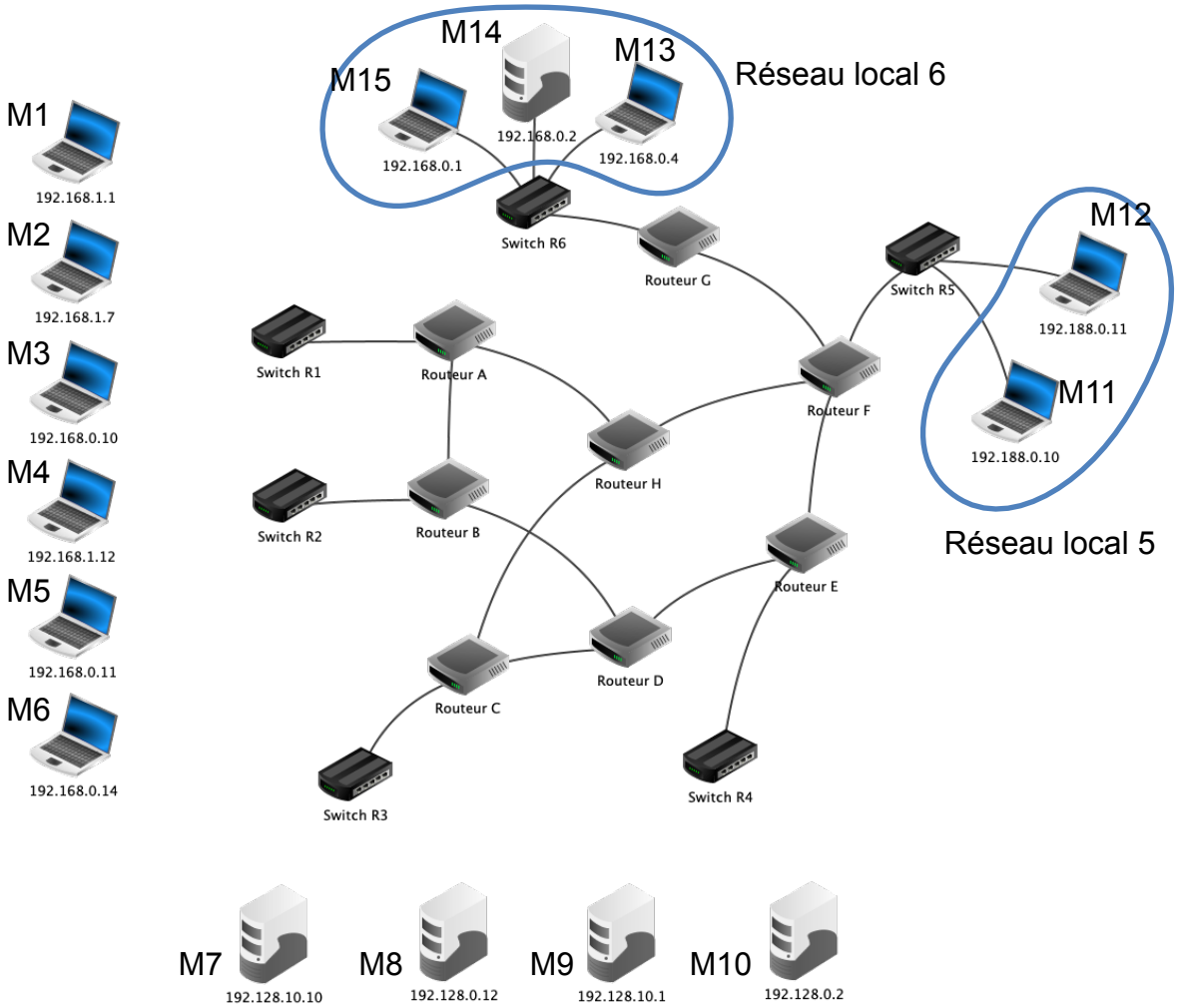
Ci-dessous : un routeur Huawei, routeur professionnel. Les multiples prises Ethernet servent à **relier d'autres routeurs**. Ce ne sont donc pas des switch.

Rem : N'importe quel ordinateur peut jouer le rôle de routeur du moment qu'il possède deux interfaces réseau.

Pour étudier les protocoles de routage, étudions l'interconnexion de réseaux page suivante.

Nous avons sur ce schéma les éléments suivants :

- 15 ordinateurs : M1 à M15
- 6 switch : R1 à R6
- 8 routeurs : A, B, C, D, E, F, G et H



Exercice **1**

Nous avons 6 réseaux locaux dans lequel chaque réseau local possède son propre switch. Leur masque de sous-réseaux est 255.255.255.0 (noté aussi /24).

- Expliquez pourquoi les machines M13, M14 et M15 appartiennent toutes au même réseau local.

_____

_____

- En utilisant la partie 0, entourez les machines appartenant au réseau local 1, 2, 3 et 4 et reliez chaque machine au switch (R1, R2, R3 et R4) correspondant par un câble.

Le but est bien sur de permettre à tout ce beau monde de communiquer !

Plusieurs cas se présentent :

- **Cas n°1 : M1 veut communiquer avec M2**

_____

_____

_____

❖ **Cas n°2 : M1 veut communiquer avec M6**

❖ **Cas n°3 : M1 veut communiquer avec M10**
Écrire la route que les paquets vont prendre.

❖ **Cas n°4 : M13 veut communiquer avec M10**
Donner deux routes que les paquets vont pouvoir suivre.

Quel serait l'intérêt de la route la plus courte en terme de liaisons ? À quoi pourrait servir la deuxième route ?

❐ — Exercice **2** —

Déterminer un chemin possible permettant d'établir une connexion entre les machines M4 et M14.

Des conclusions … et des questions :

- ❖ On a remarqué qu'il existe souvent plusieurs chemins possibles pour relier 2 ordinateurs : c'est très important pour la suite !
- ❖ Question 1 : comment les switch ou les routeurs savent-il où envoyer les paquets de données ?
- ❖ Question 2 : comment un routeur choisit-il un chemin par rapport à un autre ?

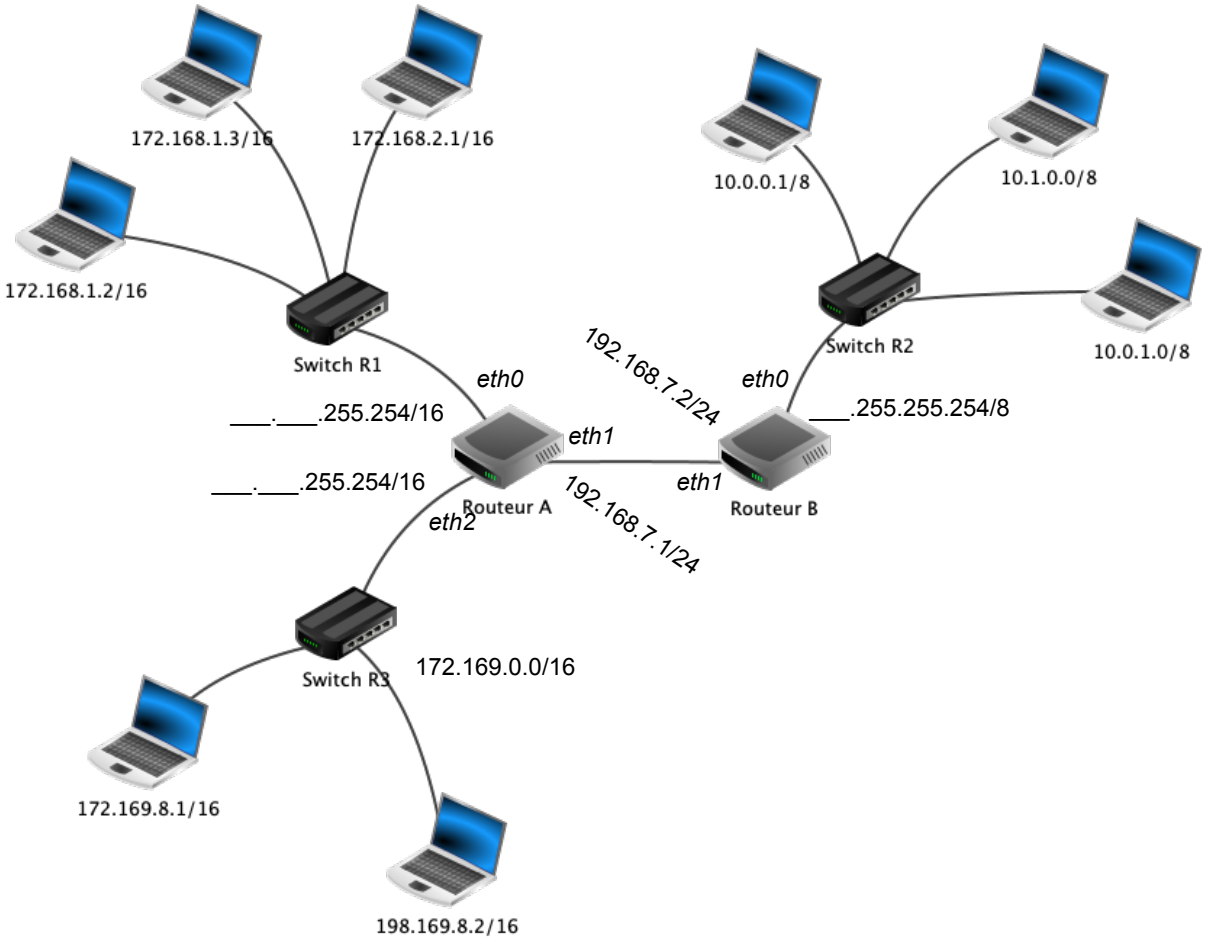
2) Établissement des tables de routage

- Dans le schéma précédent, M1 et M4 n'ont pas la même adresse réseau. Si M1 cherche à entrer en communication avec M4, le switch R1 va constater que M4 n'appartient pas au réseau local (grâce à son adresse IP) : R1 va donc envoyer le paquet de données vers le routeur A.
- Le routeur A doit à présent gérer la transmission du paquet. Pour réaliser cela, chaque routeur possède une table de routage.

Définition : _____

_____

Soit le schéma suivant :



Exercice 3

Étudiez attentivement le schéma ci-dessus.

- Le choix des adresses IP des machines a été fait au "hasard". Vérifiez que tout est cohérent pour les machines d'un même réseau local : adresses machines avec adresses réseaux. Des erreurs ont peut-être été commises pour les machines…
- Les switch possèdent l'adresse réseau (donc adresse machine = 0). Sur le schéma, complétez les adresses des switch R1 et R2.
- Complétez les adresses des routeurs afin de permettre à ceux-ci de dialoguer avec R1, R2 et R3.

Analyse du schéma :
Vous avez sans doute remarqué que nous avons 2 routeurs :

- le routeur A qui possède 3 interfaces réseau que l'on nomme eth0, eth1 et eth2. Les adresses IP liées à ces interfaces réseau sont : 172.168.255.254/16 (eth0), 172.169.255.254/16 (eth2) et 192.168.7.1/24 (eth1)
- le routeur B qui possède 2 interfaces réseau que l'on nomme eth0 et eth1. Les adresses IP liées à ces interfaces réseau sont : 10.255.255.254/8 (eth0) et 192.168.7.2/24 (eth1)

Dans la table de routage de A, nous devons donc enregistrer quelques informations :

- le routeur A est directement relié au réseau 172.168.0.0/16 par son interface eth0
- le routeur A est directement relié au réseau 172.169.0.0/16 par son interface eth2
- le routeur A est directement relié au réseau 192.168.7.0/24 par son interface eth1[1]

Le routeur A n'est pas directement relié au réseau 10.0.0.0/8 mais par contre il "sait" que les paquets à destination de ce réseau doivent être envoyé à la machine d'adresse IP 192.168.7.2/24 (c'est à dire le routeur B qui lui est directement relié au réseau 10.0.0.0/8)

On peut résumer tout cela avec le tableau suivant, appelé table de routage simplifiée de A :

Réseau	Moyen de l'atteindre	Métrique

❐ — Exercice **4** —

Déterminez la table de routage du routeur B. On partira du principe que la métrique est donnée par le nombre de câbles pour atteindre le réseau considéré, les voisins immédiats ne comptant pas.

Réseau	Moyen de l'atteindre	Métrique

Des conclusions … et des questions :

- En situation réelle, où les réseaux sont très complexes, chaque routeur aura une table de routage qui comportera des dizaines — voire des centaines — de lignes. En effet, chaque routeur devra savoir vers quelle interface réseau il faudra envoyer un paquet afin qu'il puisse atteindre sa destination.

[1] le réseau 192.168.7.0/24 est un peu particulier car il est uniquement composé des routeurs A et B

- On peut trouver dans une table de routage plusieurs lignes pour une même destination : en effet, comme nous l'avons vu précédemment, il peut exister plusieurs chemins possibles pour atteindre la destination.
- Dans le cas où il existe plusieurs chemins possibles pour atteindre la même destination, le routeur va choisir le "**chemin le plus court**".
- Question 1 : Comment choisir le chemin le plus court ?
  Pour choisir ce chemin le plus court, le routeur va faire appel à la **métrique** : plus la valeur de la métrique est petite, plus le chemin pour atteindre le réseau est "court".
- Question 2 : Comment choisir la métrique ?
- Question 3: Comment un routeur arrive à remplir sa table de routage ?

La réponse est simple pour les réseaux qui sont directement reliés au routeur (métrique = 0), mais comment cela se passe-t-il pour les autres réseaux (métrique supérieure à zéro) ?

Il existe deux méthodes :
- le routage statique : chaque ligne doit être renseignée "à la main". Cette solution est seulement envisageable pour des très petits réseaux de réseaux
- le routage dynamique : on utilise des protocoles qui vont permettre de "découvrir" les différentes routes automatiquement afin de pouvoir remplir automatiquement la table de routage.

3) Protocoles de routage : le protocole RIP

Il s'agit d'un processus décentralisé et dynamique : chaque routeur est en charge d'établir sa propre table de routage. Si l'un de ses voisin disparaît ou si un nouveau voisin apparaît, il doit recalculer une nouvelle table de routage.

Principe :
On cherche à minimiser le **nombre de sauts** effectués par un paquet avant d'arriver à destination.
Lorsque les routeurs suivent le protocole RIP :

- ❖ _____
- ❖ _____
- ❖ _____

Exemple :

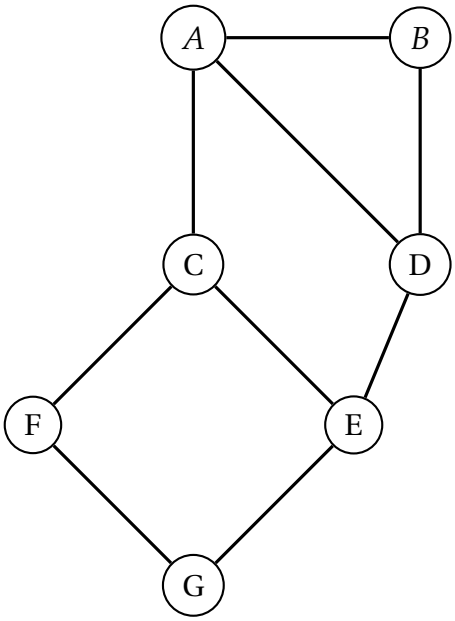
Dans cette partie, nous allons utiliser le protocole RIP pour établir les tables de routage d'un réseau simple. Celui-ci est constitué de 6 machines identifiées par les étiquettes A, B, . . ., G[2] que nous pouvons représenter dans le graphe page suivante.

Initialisation des tables de routage

Chaque routeur complète sa table de routage avec les routes qui concernent **ses voisins directs**.

Routeur C

Réseau (à atteindre)	Passerelle (Moyen de l'atteindre)	Métrique
A	—	0
E	—	0
F	—	0



Mise à jour des tables de routage

Chaque routeur demande les tables de routages de ses voisins.

- ❖ Si un de leurs voisins connait une destination qui leur est inconnue, alors ils l'ajoutent à leur table de routage.
- ❖ Si un de leurs voisins connait une destination qui leur est connue alors deux cas sont possibles :
  - ➡ soit il est plus avantageux de passer par ce voisin en terme de nombre de sauts. Le routeur met donc à jour sa table de routage pour utiliser ce voisin comme passerelle pour cette destination. Il met à jour également à jour la métrique correspondante.
  - ➡ sinon c'est que le chemin actuellement connu par le routeur est meilleur. On ne change rien.

[2] afin de simplifier la situation, nous n'utilisons pas les adresses IP.

❒ — Exercice **5** —

Mettre à jour successivement les tables de routage des routeurs dans l'ordre alphabétique : on utilisera la feuille prévue à cet effet. Si vous trouver une meilleure route, ne l'effacez pas : rayez-la proprement et rajoutez une entrée.

Une seule mise à jour était-elle suffisante ? _____

_____

Apparition d'un routeur

Supposons qu'un routeur R soit ajouté au réseau.
Il faut alors répercuter cette information dans les tables de routage :

- les routeurs voisins vont ajouter la destination R dans leurs tables de routage
- il est maintenant possible d'utiliser des routes utilisant R comme passerelle.

Puis, l'information se propage dans le réseau lors de la mise à jour des tables de routage.

❒ — Exercice **6** —

Ajouter au réseau le routeur H, relié aux routeurs D, E, et B.
Au stylo rouge, mettre à jour les tables de routage.

Disparition d'un routeur

Supposons maintenant que le routeur R soit retiré du réseau. Il faut alors répercuter cette informations dans les tables de routage :

- les routeurs voisins vont supprimer la destination R dans leurs tables de routage
- les routeurs voisins vont également supprimer les routes utilisant le routeur R comme passerelle.

Puis l'information se propage dans le réseau lors de la mise à jour de tables de routage.

Attention : Dans ce cas de figure il faut prendre garde à bien traiter les informations des tables de routages liées aux anciennes routes passant par le routeur supprimé, source d'incohérences

❒ — Exercice **7** —

Supprimer du réseau le routeur C. Mettre à jour les tables de routage.

Limitations techniques (à lire à la maison) :

Ce processus semble produire à terme une situation où les tables de routage sont stables et où les routes optimales sont utilisées (on parle de convergence). Chaque routeur du réseau doit cependant communiquer avec ses voisins de manière régulière afin de détecter les pannes ou les potentielles apparitions afin de retransmettre l'information aux autres routeurs du réseau.

À terme, l'intégralité des machines du réseau est présente dans les tables de routage de chaque routeur. Dans le cas du protocole RIP on impose une distance maximale de **15**, ce qui permet

également d'éviter des boucles de routage. Une durée de vie (TTL) sur les paquets TCP/IP permet également de limiter ces boucles de routages.

Toutefois, cela limite l'usage de ce protocole à des réseaux de petite taille. Ainsi, lorsque la taille du réseau augmente, on peut le hiérarchiser en différents domaines. Les destinations dans les tables de routage ne sont plus des machines mais des sous-réseaux.

Puisqu'on ne cherche à optimiser que le nombre de sauts, la route dans un réseau dont les tables de routage ont été générées par RIP n'est pas forcément la plus rapide, car il existe des liaisons de différentes vitesses (on parle de bande passante).

4) Protocoles de routage : le protocole OSPF

Il s'agit d'un également d'un décentralisé et dynamique. Toutefois, dans le protocole OSPF (Open Shortest Path First), la distance séparant deux machines est liée à la bande passante de la liaison qui les relie. Dans ce protocole, la distance séparant deux machines voisines est donnée par une formule du type :

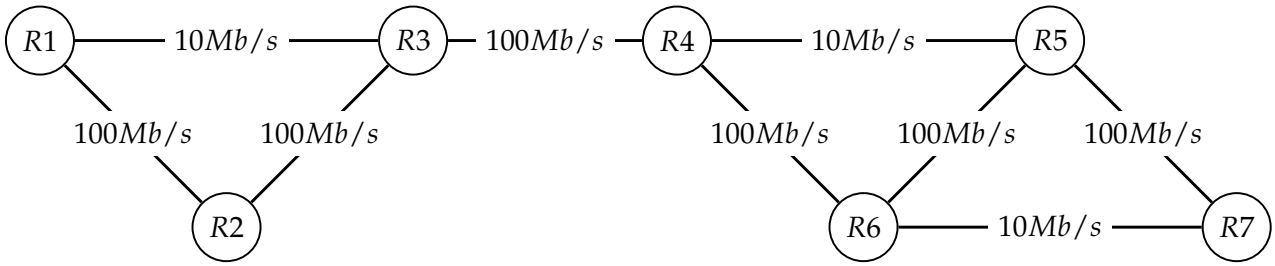
$$d(A,B) = \frac{10^8}{B}$$ avec B la bande passante de la liaison.

Il s'agit d'un protocole hiérarchique, les machines du réseau sont organisées en différentes zones.

- la zone backbone à laquelle toutes les autres zones sont connectées
- d'autres zones identifiées par un numéro unique.

Chaque zone communique avec la backbone à l'aide d'un routeur spécial nommé ABR (Area Border Routeur). Des routeurs présents dans une zone ne communiquent qu'avec les routeurs de cette zone.

Exemple : On donne la structure d'un réseau de routeurs OSPF. On a indiqué la bande passante de chaque liaison.
La zone 1 est constituée des routeurs R1, R2, et R3.
La zone 2 est constituée des routeurs R4, R5, R6, R7.
La backbone (zone 0) est constituée par les routeurs R3 et R4 : R3 et R4 sont des routeurs ABR.



Principe :
On cherche à minimiser la **vitesse de communication** entre deux machines du réseau.

- I

- _____

  _____

- _____

  _____.

Exemple :

Découverte des relations de voisinage

On met en place un processus de diffusion. À l'intérieur de chaque zone, chaque routeur envoie à travers toutes ses interfaces un message de type HELLO en multicast, qui contient toutes les informations que connait le routeur à propos du réseau. Les routeurs qui reçoivent un message de type HELLO renvoient en retour toutes les informations qu'ils connaissent à propos du réseau. Si un routeur ne répond pas, alors on efface la liaison concernée.

Après plusieurs cycles, tous les routeurs d'une zone connaissent l'état de toutes les liaisons d'une zone.

Calcul du plus court chemin

Chaque routeur connait l'intégralité de la structure de sa zone. Il peut donc calculer le plus court chemin qui le relie à chaque routeur de sa zone. Puis il remplit sa table de routage en fonction des résultats. Pour faire cela il utilise l'algorithme appelé **algorithme de Dijkstra**.

Échange d'information inter-zones

Les routeurs ABR s'échangent via la backbone la liste des routeurs présents dans leur zone, et complètent leur table de routage. Puis ils diffusent cette information à tous les routeurs de leur zone.

Algorithme de Dijkstra.

Étant donné un graphe étiqueté et pondéré par des réels positifs, l'algorithme de Dijkstra permet de calculer les plus courts chemins à partir d'une source vers tous les autres sommets du graphe.

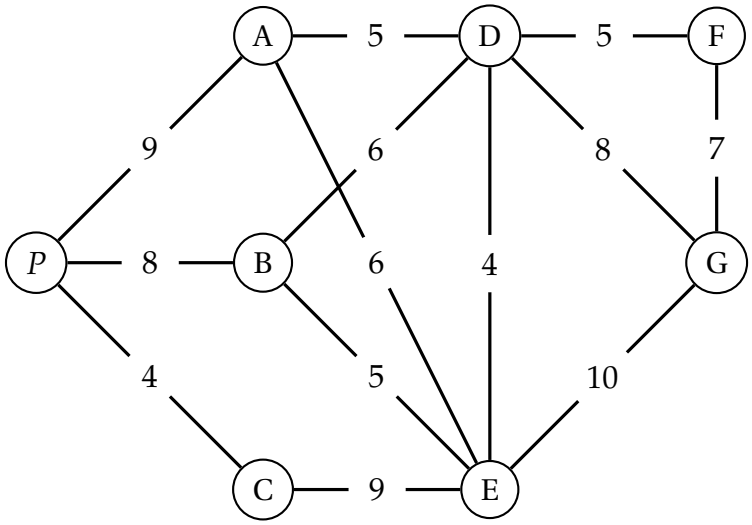
Initialement, les distances entre le sommet de départ et les autres sommets sont infinies. Le sous-graphe est le sommet de départ.

À chaque itération, on choisit un sommet en dehors du sous-graphe de distance minimale, et on l'ajoute au sous-graphe. Puis on met à jour les distances des sommets voisins du sommet ajouté : on conserve la plus faible distance. On continue ainsi tant que le sous-graphe ne recouvre pas l'intégralité du graphe.

Pas compris ? Regardez cette vidéo pour voir comment il convient de faire pour appliquer l'algorithme de Dijkstra à un graphe pondéré : https://www.youtube.com/watch?v=rI-Rc7eF4iw .

Exercice **8**

Déterminer dans le graphe ci-dessous le chemin de poids minimal reliant P à G.



P	A	B	C	D	E	F	G	Choix

II. Sécurisation des communications

1) Position du problème

L'idée de coder des messages pour les rendre illisible aux personnes non autorisées ne datent pas du début de l'ère de l'informatique. Toutes les civilisations de l'antiquité (Babyloniens, Grecs, Hébreux ou Romains) cherchaient déjà à sécuriser leurs communications importantes en codant les messages sensibles[3]. Toutes les sociétés suivantes, royaumes, théologies, démocraties ou totalitarisme ont mis au point des méthodes de codage de plus en plus sophistiquées.

Dans cette partie, nous nous intéresserons ici uniquement aux communications ayant lieu par l'intermédiaire d'un réseau informatique. Comme nous avons vu en Première, toute donnée (numérique, alphanumérique…) peut être encodée en binaire.

Nous chercherons donc uniquement à coder des suites de zéro et de un.

Alix et Bill cherchent à s'envoyer des messages par l'intermédiaire d'un réseau informatique tel que décrit dans la partie précédente. Alix et Bill désirent qu'une tierce personne (par exemple Clément) ne soit pas capable de lire leur conversation si par hasard celle-ci devait être interceptée.

Pour se faire, Alix va devoir crypter (on dit aussi *chiffrer*) son message. Toute personne qui ne possédera pas le **moyen de déchiffrer** ce message crypté se verra dans l'impossibilité de comprendre le contenu du message, même si elle a intercepté celui-ci. Il existe 2 grands types de chiffrement : le chiffrement symétrique et le chiffrement asymétrique.

Une méthode communément utilisée pour chiffrer un message consiste en l'utilisation d'une suite de caractère que l'on appelle **clé de chiffrement**.

Définition :

Une clé de chiffrement est un paramètre déterminant complètement **le résultat d'une opération de chiffrement**.

2) Le chiffrement symétrique

Propriété :

_____

_____

Nous allons étudier le cas du chiffrement symétrique au travers de l'exemple suivant. L'algorithme de chiffrement proposé ci-dessous n'est en aucun cas un algorithme réel mais il permet d'expliquer clairement le fonctionnement et l'intérêt du chiffrement symétrique.

Exemple :

Ainsi, dans le cas du chiffrement **symétrique**, la clé de chiffrement utilisée par Alix pour chiffrer son message est aussi celle utilisée par Bill pour le déchiffrer.

Alix veut coder le message "**100% Top Secret!**" :

[3] pour en savoir plus sur l'histoire du chiffrement, consultez la page Wikipédia.

1) Rappelez comment convertir un message texte en binaire. Quel encodage peut être utilisé dans ce cas ? _____

2) En utilisant https://tinyurl.com/y9t8gt7l (ou le 2D barcode), réalisez la conversion du message d'Alix en binaire. On aura un message de 16 octets, chaque octet étant séparé par un espace.

00110001 00110000 00110000 00100101 00100000 01010100 01101111 01110000 00100000 01010011 01100101 01100011 01110010 01100101 01110100 00100001

3) Nous allons à présent choisir une clé de chiffrement. Dans notre cas, nous choisirons **toto.** Codez **toto** en binaire. **01110100 01101111 01110100 01101111**

Pour chiffrer un message M_1, nous allons utiliser une fonction f qui permet de transformer notre message en un autre message M_2 illisible par Clément grâce à la clé C :

$$M_2 = f(M_1, C)$$

Travaillant en binaire, la fonction f est une opération binaire suivant les règles de la logique booléenne. On choisit pour f l'opération "Ou Exclusif" (XOR) **que l'on applique bit à bit**. Pour rappel, la table de vérité de l'opérateur XOR est donnée ci-contre :

E1	E2	S
0	0	0
0	1	1
1	0	1
1	1	0

La clé étant plus courte que le message, on reproduit la clé vers la droite autant de fois que nécessaire.

4) **a.** À la main, réalisez l'opération XOR sur le premier octet du message converti en binaire de la question 2.

---

b. On va utiliser un programme Python pour faire cette opération plus rapidement. Complétez la méthode statique **xor** calculant la table de vérité de l'opérateur **xor**, ainsi que la méthode **crypter** proposée sur bouillotvincent.github.io .
Quel est le message crypté en binaire, puis en ASCII ?

Maintenant, le message est prêt à être envoyé à Bill (on enverra la **version binaire**). Si Clément intercepte le message et cherche à le lire avec un éditeur de texte, il obtiendra la suite de caractère **E_DJT;T<N**.

Bill a maintenant reçu le message chiffré, il possède la clé (**toto**), il va donc pouvoir déchiffrer le message en appliquant un XOR entre le message chiffré et la clé. Il va donc appliquer exactement la même méthode que ci-dessus.

5) Au sein de la classe symmetricCrypto, complétez la méthode decrypter qui permet de décrypter un message binaire crypté.
6) Peut-on utiliser n'importe quelle opérateur booléen ? Pour justifier votre raisonnement, à l'aide d'une table de vérité, expliquez par exemple pourquoi la fonction OR ne peut pas fonctionner.

Vous avez du remarquer que nous avons bien retrouvé le code binaire d'origine et donc le message d'origine !

7) Utilisez la fonction décoder de la classe symmetricCrypto pour vérifier que le message reçu par Bill est bien le bon.

8) À l'aide de la classe symmetricCrypto, cryptez un mot **unique** de plusieurs lettres et utilisez une clé connue de vous seul. Donnez le code binaire obtenu (crypté) à votre voisin. Celui-ci doit essayer de retrouver votre mot sachant que vous avez utilisé un opérateur XOR. Y arrive-t-il sans la clé de chiffrement ? Donnez-lui ensuite la clé de chiffrement.

Remarques :

- La méthode la plus utilisée en matière de chiffrement symétrique se nomme AES (Advanced Encryption Standard). Cette méthode utilise une technique de chiffrement plus élaborée que ce qui a été vu ci-dessus, mais les grands principes restent identiques. En particulier, l'utilisation de la fonction XOR est présente.
- Le gros problème du chiffrement symétrique, c'est qu'il est nécessaire pour Alix et Bill de se mettre d'accord à l'avance sur la clé qui sera utilisée lors des échanges.

3) Le chiffrement asymétrique

On vient de voir que le chiffrement symétrique demande de se mettre d'accord sur une clé privée que l'on partage. Mais comment faire lorsque deux ordinateurs distants doivent réaliser une telle opération sur Internet ? Partager la clé en clair serait trop dangereux.

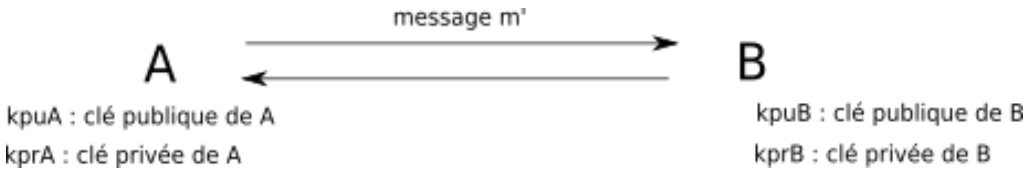
Dans le cas du chiffrement asymétrique Alix et Bill n'ont pas besoin de partager une "clé secrète" et peuvent tout de même communiquer de manière secrète !

Propriété :

- _____
- _____

En théorie, Alix possède une "clé privée" que l'on notera **kprA** et une "clé publique" que l'on notera **kpuA**. Alix ne devra jamais diffuser sa clé privée : elle devra rester strictement secrète.

En revanche sa clé publique pourra être connue de tout le monde sans aucun problème.
Bill possède également une "clé privée" que l'on notera **kprB** et une "clé publique" que l'on notera **kpuB**.



Si Alix désire envoyer un message m à Bill, elle va utiliser la clé publique de Bill afin de réaliser le chiffrement (m est chiffré en m'). Le message chiffré m' va ensuite pouvoir transiter entre Alix et Bill.

Une fois le message m' en sa possession, Bill va utiliser sa clé privée afin de pouvoir déchiffrer le message m' et ainsi obtenir le message m.
Le processus peut être résumé par le schéma ci-contre :

Si Clément intercepte le message m', il sera incapable de déterminer m à partir de m' sans la clé privée de Bill. C'est la théorie… mais cela semble impossible !

Activité débranchée : utilisation du chiffrement asymétrique pour échanger des informations

Le chiffrement asymétrique repose sur des problèmes très difficiles à résoudre dans un sens et faciles à résoudre dans l'autre sens. Dans l'activité débranchée, nous avons utilisé un problème appelé "ensemble dominant". Relié à la théorie des graphes, ce problème algorithmique a été prouvé comme étant NP-complet, donc complexe à résoudre !

L'exemple le plus utilisé en cryptographie est l'algorithme de chiffrement asymétrique RSA (du nom de ses 3 inventeurs : Rivest Shamir et Adleman). Il est particulièrement utilisé dans le domaine du commerce électronique. RSA se base sur la factorisation des très grands nombres premiers. Si vous prenez un nombre premier A (par exemple A = 16813007) et un nombre premier B (par exemple B = 258027589), il facile de déterminer C le produit de A par B (ici on a A x B = C avec C = 4338219660050123). En revanche si je vous donne C (ici 4338219660050123) il est très difficile de retrouver A et B. C est la clé publique, A (ou B) est la clé privée.

À ce jour, aucun algorithme n'est capable de retrouver A et B connaissant C dans un temps "raisonnable". Cela vient du fait que nous ne disposons pas d'algorithmes permettant de savoir rapidement si un nombre est premier, ce qui en fait un problème mathématique important encore aujourd'hui… Nous avons donc bien ici un problème relativement facile dans un sens (trouver C à partir de A et B) est extrêmement difficile dans l'autre sens (trouver A et B à partir de C). Les détails du fonctionnement de RSA sont étudiés en Maths Expertes et ne seront pas abordés ici.
Vous devez juste savoir qu'il existe un lien entre une clé publique et la clé privée correspondante, mais qu'il est impossible de trouver la clé privée de quelqu'un à partir de sa clé publique **en un temps raisonnable**.

Mais alors, quel est l'avantage des méthodes de chiffrement symétrique par rapport aux méthodes de chiffrement asymétrique ?

L'un des principaux problèmes et frein à l'utilisation des algorithmes de chiffrage asymétrique réside dans **ses faibles performances**. A titre d'exemple, un algorithme de cryptage symétrique permet de décrypter 4000 fois plus rapidement qu'un algorithme asymétrique.

4) Application au protocole HTTPS

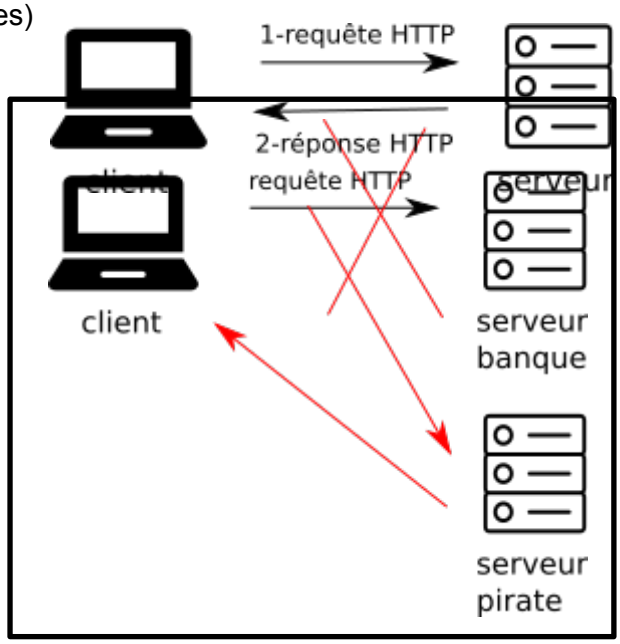
Nous allons maintenant voir une utilisation concrète de ces chiffrements symétriques et asymétriques au travers du protocole HTTPS.

a. Limitations du protocole HTTP

Avant de parler du protocole HTTPS, il convient de ré-expliquer le protocole HTTP et d'évoquer ses limites : lorsqu'un client effectue une requête HTTP vers un serveur, le serveur va répondre à cette requête (par exemple en envoyant une page HTML au client)[4].

Le protocole HTTP pose 2 problèmes en termes de sécurité informatique :

- Un individu qui intercepterait les données transitant entre le client et le serveur pourrait les lire sans aucun problème (ce qui serait problématique notamment avec un site de e-commerce au moment où le client envoie des données bancaires) ;
- grâce à une technique appelée le DNS spoofing, un serveur "pirate" peut se faire passer pour un site sur lequel vous avez l'habitude de vous rendre en toute confiance : imaginez que vous voulez consulter vos comptes bancaires en ligne, vous saisissez l'adresse web de votre banque dans la barre d'adresse de votre navigateur favori, vous arrivez sur la page d'accueil d'un site en tout point identique au site de votre banque, en toute confiance, vous saisissez votre identifiant et votre mot de passe.
  C'est terminé un "pirate" va pouvoir récupérer votre identifiant et votre mot de passe ! Pourquoi ? Vous avez saisi l'adresse web de votre banque comme d'habitude ! Oui, sauf que grâce à une attaque de type "DNS spoofing" vous avez été redirigé vers un site pirate, en tout point identique au site de votre banque. Dès vos identifiant et mot de passe saisis sur ce faux site, le pirate pourra les récupérer et se rendre avec sur le véritable site de votre banque. À noter qu'il existe d'autres techniques que le DNS spoofing qui permettent de substituer un serveur à un autre, mais elles ne seront pas évoquées ici.



HTTPS est donc la version sécurisée de HTTP, le but de HTTPS est principalement d'éviter les 2 problèmes évoqués ci-dessus. HTTPS s'appuie sur le protocole TSL (Transport Layer Security) anciennement connu sous le nom de SSL (Secure Sockets Layer)

[4] Si nécessaire, reprenez vos notes de Première de l'an dernier pour plus de détails.

b. *Comment chiffrer les données circulant entre le client et le serveur ?*

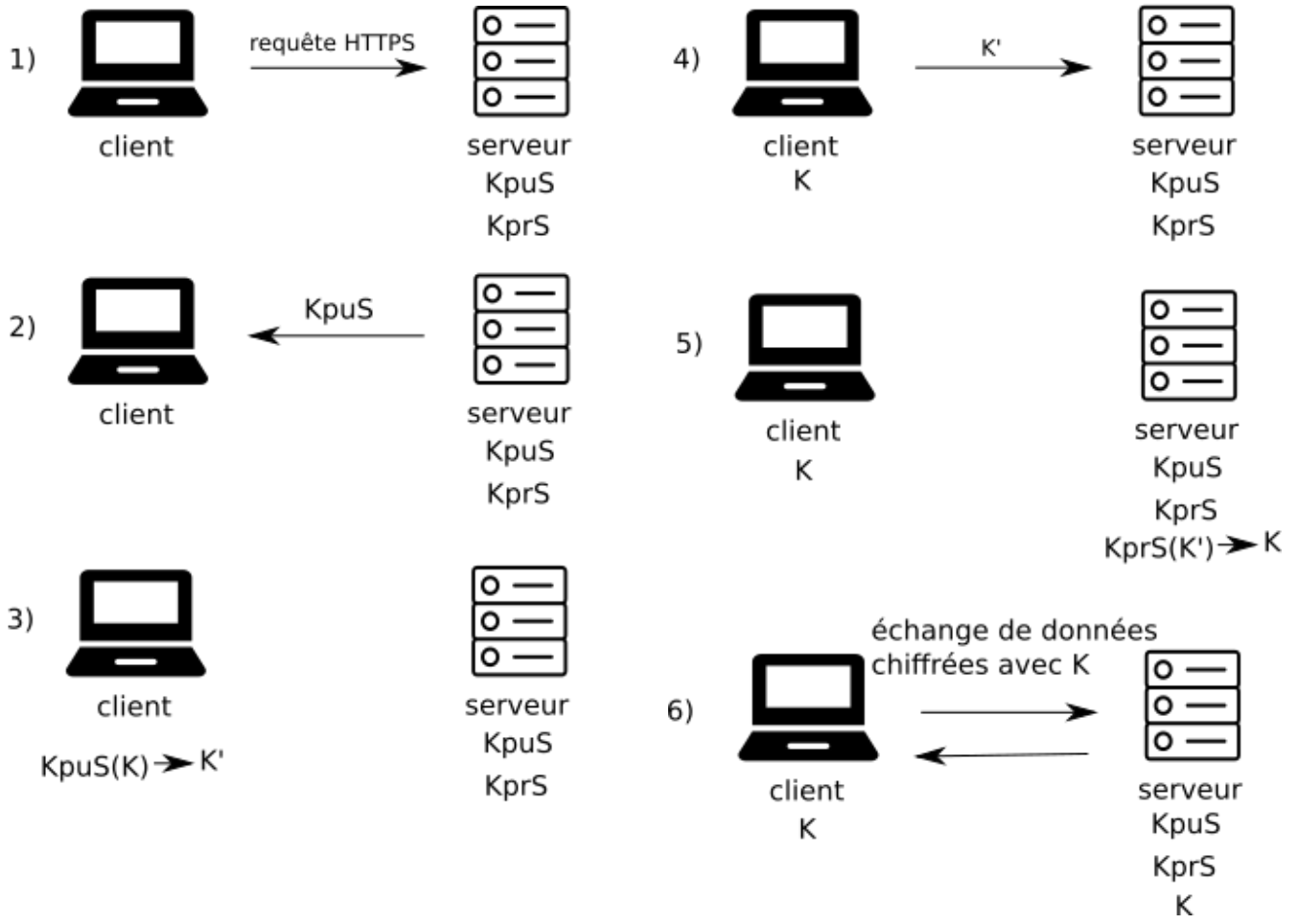
Exercice :
En utilisant les méthodes de chiffrement symétrique et asymétrique, imaginez une méthode permettant de chiffrer, en temps réel, les données circulant entre le client et le serveur. Vous expliquerez point par point votre méthode et serez amené à exposer votre proposition à la classe.

Réponse :
Pour assurer des communications en temps réel, celles-ci vont être chiffrées grâce à un chiffrement symétrique. Problème : comment échanger cette clé de chiffrement symétrique entre le client et le serveur ? Simplement en utilisant un chiffrement asymétrique avec une paire clé publique / clé privée !

Voici le déroulement des opérations :

- le client effectue une requête HTTPS vers le serveur, en retour le serveur envoie sa clé publique (KpuS) au client
- le client fabrique une **"clé symétrique" K** (qui sera utilisé pour chiffrer les futurs échanges), chiffre cette clé K avec KpuS et envoie la version chiffrée de la clé K au serveur ;
- le serveur reçoit la version chiffrée de la clé K et la déchiffre en utilisant sa clé privée (KprS). À partir de ce moment-là, le client et le serveur sont en possession de la clé de chiffrement symétrique K
- le client et le serveur commencent à échanger des données en les chiffrant et en les déchiffrant à l'aide de la clé K (chiffrement symétrique).

On peut résumer ce processus avec le schéma suivant :



Ce processus se répète à chaque fois qu'un nouveau client effectue une requête HTTPS vers le serveur et résout les problèmes d'interception des données.

Conclusion : comment éviter les attaques de type DNS Spoofing ?

Pour éviter tout problème, il faut que le serveur puisse justifier de son "identité" ("voici la preuve que je suis bien le site de la banque B et pas un site "pirate""). Pour se faire, chaque site désirant proposer des transactions HTTPS doit, périodiquement, acheter **un certificat d'authentification** auprès d'une autorité habilitée à fournir ce genre de certificats.

Nous n'entrerons pas dans les détails du fonctionnement des certificats, mais vous devez savoir que le serveur envoie **ce certificat au client en même temps que sa clé publique** (étape 2 du schéma précédent). En cas d'absence de certificat (ou d'envoi de certificat non conforme), le client stoppe immédiatement les échanges avec le serveur.

Il peut arriver de temps en temps que le responsable d'un site oublie de renouveler son certificat à temps (dépasse la date d'expiration) ou possède un certificat d'une autorité non reconnue par votre navigateur. Dans ce cas, le navigateur web côté client affichera une page de mise en garde comme ci-dessous :