

## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



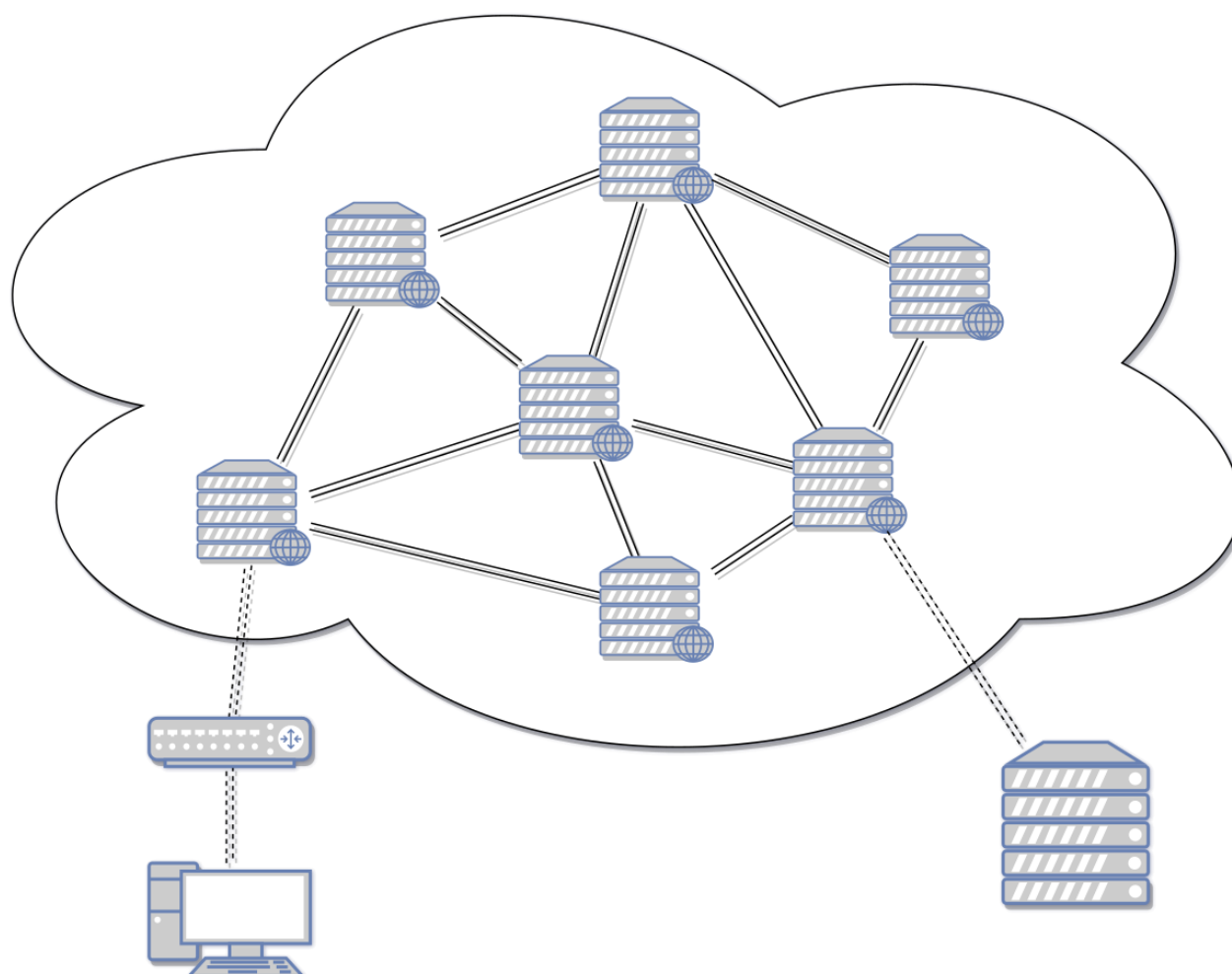
Nom :

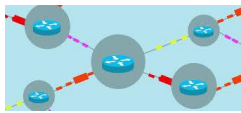
Prénom :

Date :

## Table des matières

1	Les SoC (System On a Chip) .....	2
2	Les processus .....	5
3	Les protocoles de routage.....	14
4	La sécurisation des communications .....	25





## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



## 1 Les SoC (System On a Chip)

Qu'est-ce qu'un « System on a Chip » ?

On pourrait définir l'expression « Système sur une puce » en une phrase :

Un « **système sur une puce** », souvent désigné dans la littérature scientifique par le terme anglais « system on a chip » (d'où son abréviation SoC), est un système complet embarqué sur une seule puce ("circuit intégré"), pouvant comprendre de la mémoire, un ou plusieurs microprocesseurs, des périphériques d'interface, ou tout autre composant nécessaire à la réalisation de la fonction attendue.



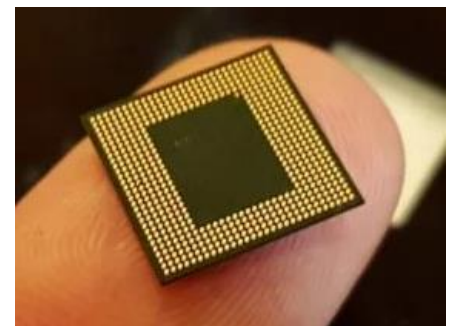
Afin de mieux comprendre les caractéristiques d'une architecture basée sur un SOC, nous allons la comparer à celle d'un ordinateur standard de type PC.

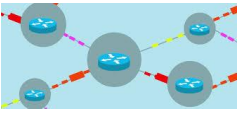
En effet, un PC se compose de plusieurs éléments essentiels: le processeur, la RAM, la carte graphique, les cartes réseau, les disques durs... Chacun de ces composants est ensuite monté sur un élément central: la carte mère.

Le SoC quant à lui est un peu comme cette carte mère, mais pour nos appareils mobiles. En effet, son architecture regroupe une grande variété de puces telles que le CPU (processeur), le GPU (carte graphique), les dispositifs réseau tels que le Wi-Fi ou la 5G ainsi que bien d'autres microéléments.

On entend souvent dire que les téléphones portables (smartphones) sont de véritables ordinateurs, ce qui est vrai. On peut s'interroger sur la taille d'un smartphone par rapport à la taille d'un PC (la carte mère d'un PC mesure environ 25 cm sur 30 cm, soit bien plus qu'un smartphone). Pourtant on doit obligatoirement trouver dans un smartphone les mêmes composants que dans un PC : CPU, RAM, carte graphique et interfaces réseau (Wifi et Bluetooth dans le cas d'un smartphone) !

La solution ? Placer tous ces composants dans une puce unique d'une centaine de mm<sup>2</sup> : le SoC



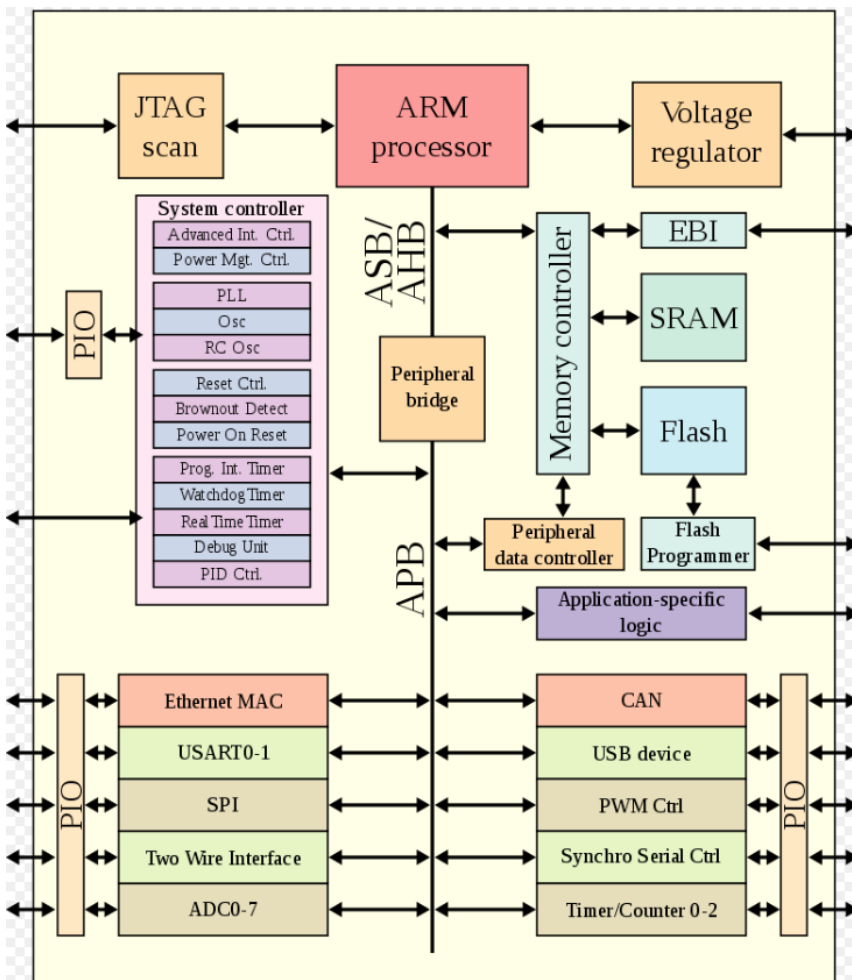


## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



Le SoC est un composant regroupant l'ensemble des puces intégrées dans un appareil mobile. Ainsi, on les retrouve par exemple dans nos téléphones, tablettes ou même nos montres connectées. Nommé généralement « processeur mobile », celui-ci est la pierre angulaire permettant le bon fonctionnement de nos smartphones.



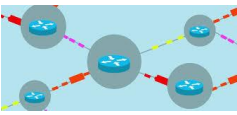
Voici par exemple le schéma simplifié du SoC ARM.

Il intègre sur une seule puce : microprocesseur, processeur graphique (GPU), DSP, FPU, SIMD, et contrôleur de périphériques. Ceux-ci sont présents dans la majorité des smartphones et tablettes.

ARM propose des architectures qui sont vendues sous licence de propriété intellectuelle aux concepteurs. Ils proposent différentes options dans lesquelles les constructeurs peuvent prendre ce qui les intéresse pour compléter avec leurs options propres ou de concepteurs tiers.

ARM propose ainsi pour les SoC les plus récents les microprocesseurs Cortex (Cortex-A pour les dispositifs portables de type smartphones et tablettes, Cortex-M pour le couplage à un microcontrôleur, Cortex-R pour les microprocesseurs temps réel), des processeurs graphiques (Mali), des bus AMBA sous licence libre, ainsi que les divers autres composants nécessaires à la composition du SoC complet. Certains

constructeurs, tels que Nvidia, préfèrent produire leur propre processeur graphique, d'autres, comme Samsung, préfèrent prendre dans certains cas un processeur graphique de prestataire tiers ou d'ARM selon les modèles, et d'autres, comme Apple, modifient certains composants du microprocesseur en mélangeant plusieurs architectures processeur ARM (l'Apple A6 par exemple, mixe les technologies de microprocesseur Cortex-A9 et Cortex-A15).

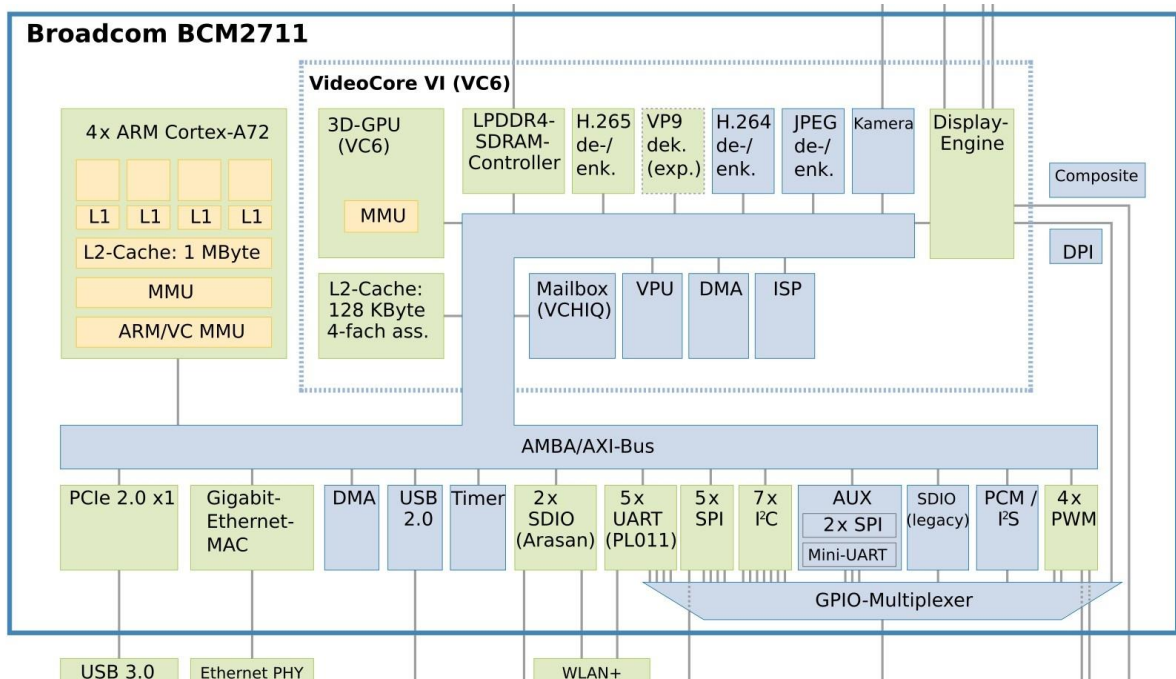


## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



Ci-dessous le schéma du SoC Broadcom bcm2711 qui équipe notamment le nano ordinateur Raspberry PI 4 modèle B :



Oltre leur taille, les Soc ont d'autres avantages par rapport aux systèmes "classiques" (carte mère + CPU + carte graphique...) :

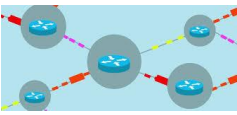
- Les SoC sont conçus pour consommer beaucoup moins d'énergie qu'un système classique (à puissance de calcul équivalente).
- Cette consommation réduite permet dans la plupart des cas de s'affranchir de la présence de système de refroidissement actif comme les ventilateurs (voir l'image du PC ci-dessus). Un système équipé de SoC est donc silencieux.
- Vu les distances réduites entre, par exemple, le CPU et la mémoire, les données circulent beaucoup plus vite, ce qui permet d'améliorer les performances. En effet, dans les systèmes "classiques" les bus (si nécessaire voir le cours de première : Modèle d'architecture de von Neumann à propos des bus) sont souvent des "goulots d'étranglement" en termes de performances à cause de la vitesse de circulation des données.

En revanche, là où un ordinateur équipé d'une carte mère permet de faire évoluer les composants individuellement, l'extrême intégration du SoC présente l'inconvénient de n'autoriser aucune mise à jour du matériel.

## A faire vous-même I

Recherchez les principales caractéristiques des SoC(s) « Broadcom BCM2711 » et « Broadcom BCM2837 » puis comparez les performances des modèles Raspberry Pi respectifs. Vous pourrez vous aider du support de cours intitulé « Découverte\_NanoOrdi\_V2.pdf ».





## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



## 2 Les processus

### 2.1 Qu'est-ce que c'est ?

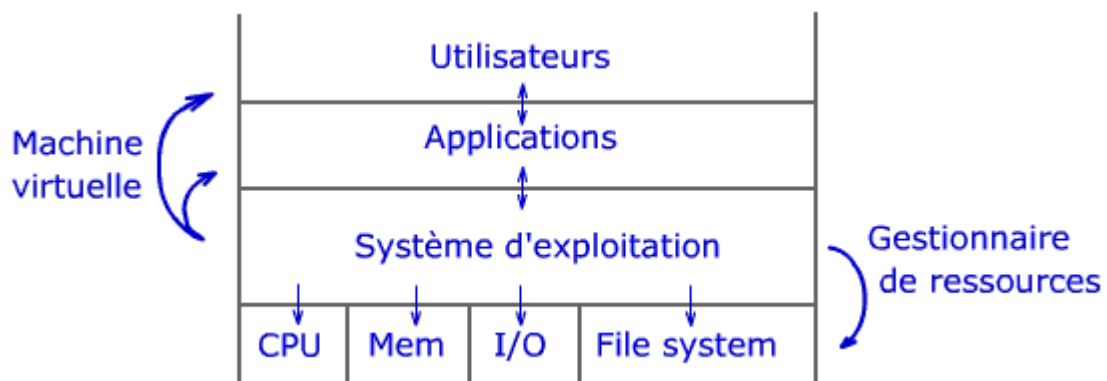
Pour bien comprendre le concept de processus informatique, il est utile au préalable de revenir sur le concept de système d'exploitation.

Le système d'exploitation automatique des ressources de l'ordinateur, communément appelé "système d'exploitation" ou Operating System O.S. et parfois "système opératoire", est généralement représenté comme une couche logicielle placée au-dessus du matériel (« hardware »).

Au système d'exploitation est associée une interface utilisateur (texte ou graphique) capable d'interpréter une série de commandes de base. Cette interface ne fait (théoriquement) pas partie du système d'exploitation mais l'étude des commandes donne un bon aperçu des fonctions qu'il procure. La convivialité du système dépend de la richesse et de la simplicité de l'interface homme/machine.

Le système d'exploitation constitue une machine virtuelle qui, pour les applications, substitue des composants logiciels aux composants matériels. Il est la plateforme pour laquelle sont construites les applications.

Pour ce qui est du développement des applications, l'OS propose une interface de programmation appelée API (Application Program Interface). Elle est une sorte de boîte à outils à laquelle les développeurs recourent pour construire leurs applications. Ces API procurent une vue uniforme et simplifiée des ressources de la machine. Cela permet aux applications de faire abstraction des particularités du matériel en dissimulant sa diversité et sa complexité.




Le système d'exploitation reçoit des demandes des programmes d'application et des utilisateurs. Il y donne suite en leur allouant les ressources du système :

- le CPU
- la mémoire
- les périphériques
- le système de fichier (y compris parfois, le réseau)

Un **programme** écrit à l'aide d'un langage de haut de niveau (on parle de "code source") est traduit en langage machine afin de pouvoir être exécuté par un ordinateur comme une suite statique d'instructions (cf. le support intitulé « Découverte\_NanoOrd\_V2.pdf »).



Lycée d'enseignement général et technologique international Victor Hugo COLOMIERS		
	<div>Architectures matérielles et Systèmes d'Exploitation</div> <div>Cours &amp; Activité Pratique</div>	

Un **processeur** est l'agent qui exécute les instructions d'un programme

Un **processus** est un programme en cours d'exécution. Un programme peut avoir plusieurs exécutions simultanées. Pour faire tourner un processus il faut donc, non seulement chercher le code et les données mais il faut aussi lui réserver un espace mémoire, le gérer les accès aux ressources, la sécurité etc. C'est un des rôles du système d'exploitation.

## 2.2 Les états d'un processus

Tous les systèmes d'exploitation "modernes" (Linux, Windows, macOS, Android, iOS...) sont multitâches. Le temps processeur est partagé par plusieurs processus qui semblent tourner simultanément. Mais pour être précis, le partage se fait dans un mode "chacun son tour". Pour gérer ce "chacun son tour", les systèmes d'exploitation attribuent des "états" au processus.

Les différents états sont:

- Lorsqu'un processus est en train de s'exécuter (qu'il utilise le microprocesseur), on dit que le processus est dans l'état "élu".
- Un processus qui se trouve dans l'état élu peut demander à accéder à une ressource non disponible instantanément (par exemple lire une donnée sur le disque dur). Le processus ne peut pas poursuivre son exécution tant qu'il n'a pas obtenu cette ressource. En attendant de recevoir cette ressource, il passe de l'état "élu" à l'état "bloqué"
- Lorsque le processus finit par obtenir la ressource attendue, celui-ci peut potentiellement reprendre son exécution. Mais comme nous l'avons vu ci-dessus, les systèmes d'exploitation permettent de gérer plusieurs processus "simultanément", mais un seul processus peut se trouver dans un état "élu" (le microprocesseur ne peut "s'occuper" que d'un seul processus à la fois). Quand un processus passe d'un état "élu" à un état "bloqué", un autre processus peut alors "prendre sa place" et passer dans l'état "élu". Le processus qui vient de recevoir la ressource attendue ne va donc pas pouvoir reprendre son exécution immédiatement, car pendant qu'il était dans l'état "bloqué" un autre processus a "pris sa place". Un processus qui quitte l'état bloqué ne repasse pas forcément à l'état "élu". Il peut, en attendant que "la place se libère" passer dans l'état "prêt" (cela signifie : « j'ai obtenu ce que j'attendais, je suis prêt à reprendre mon exécution dès que la place sera libre »).

Le passage de l'état "prêt" à l'état "élu" constitue l'opération "d'élection". Le passage de l'état « élu » à l'état « bloqué » est l'opération de "blocage". Un processus est toujours créé dans l'état "prêt". Pour se terminer, un processus doit obligatoirement se trouver dans l'état "élu".

Lycée d'enseignement général et technologique international Victor Hugo COLOMIERS		
	Architectures matérielles et Systèmes d'Exploitation Cours & Activité Pratique	

On peut résumer tout cela avec le diagramme suivant :

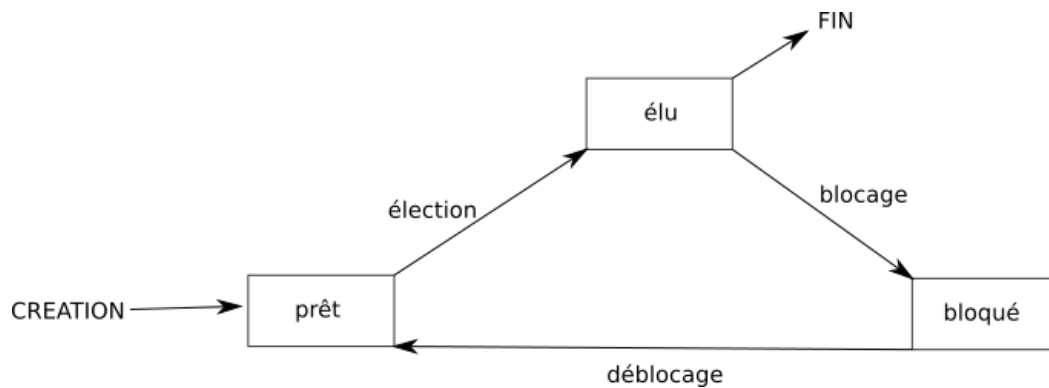


Diagramme réalisé par David ROCHE

Le "chef d'orchestre" qui attribue aux processus leur état "élu", "bloqué" ou "prêt" est le système d'exploitation. On dit que le système gère l'ordonnancement des processus (tel processus sera prioritaire sur tel autre...)

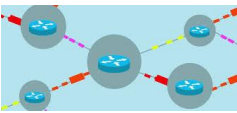
Un processus qui utilise une ressource R doit la "libérer" une fois qu'il a fini de l'utiliser afin de la rendre disponible pour les autres processus. Pour libérer une ressource, un processus doit obligatoirement être dans un état "élu".

## 2.3 Création d'un processus

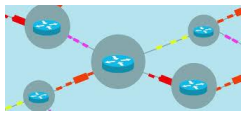
Un processus peut créer un ou plusieurs processus à l'aide d'une commande système ("fork" sous les systèmes de type Unix). Imaginons un processus A qui crée un processus B. On dira que A est le père de B et que B est le fils de A. B peut, à son tour créer un processus C (B sera le père de C et C le fils de B). On peut modéliser ces relations père/fils par une structure arborescente (cf. le support de cours intitulé « SDD\_V1.pdf »).

Si un processus est créé à partir d'un autre processus, comment est créé le tout premier processus ?

Avec un système d'exploitation comme Raspbian (Linux), au moment du démarrage de l'ordinateur, un tout premier processus (appelé « processus 0 » ou encore « Swapper ») est créé à partir de "rien" (il n'est le fils d'aucun processus). Ensuite, ce processus 0 crée un processus souvent appelé "init" ("init" est donc le fils du processus 0). À partir de "init", les processus nécessaires au bon fonctionnement du système sont créés (par exemple les processus "crond", "inetd", "getty",...). Puis d'autres processus sont créés à partir des fils de "init".







## Architectures matérielles et Systèmes d'Exploitation



### Cours & Activité Pratique

b. Utilisation des commandes permettant de visualiser les processus.

Après avoir ouvert un terminal, tapez la commande suivante : `ps -aef`. Quel est le processus de PID 1 ? A-t-il plusieurs fils ? Quel est le propriétaire des dix premiers processus ? des vingt premiers ? Dans quel dossier est situé le programme responsable du processus de PID 1 ? Quelles sont les permissions du programme ?

Le processus de PID 1 est root

Oui il a plusieurs fils

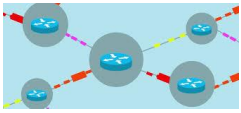
Le propriétaire des 10 premiers processus et des vingt premiers est le processus est le PID 2

Le dossier responsable du PID 1 est /sbin/init splash

c. Les processus en « temps réel ». Après avoir ouvert un terminal, tapez la commande suivante :

```
pi@raspberrypi:~$ top
```

Pour en savoir plus sur la commande, faites un « man top » ou consultez la page <http://debian-facile.org/doc:systeme:top>



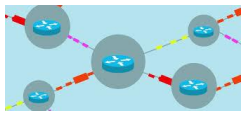
## Architectures matérielles et Systèmes d'Exploitation



### Cours & Activité Pratique

- d. En utilisant la commande `top` dans un terminal, observez ce qui se passe au niveau des processus quand vous ouvrez ou supprimez un onglet dans le navigateur web « chromium ».

Quand un onglet est ouvert, un nouveau processus est crée  
Les processus sont fini quand le navigateur est fermé



## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



- e. Suppression d'un processus. Il est possible de supprimer un processus en utilisant la commande « kill ». Sa syntaxe est la suivante :

Kill [options] <pid> [...liste pid...]

Ouvrez 2 terminaux, placez-les l'un à côté de l'autre. Dans l'un des 2 terminaux, exécutez la commande top comme ceci :

```

pi@raspberrypi: ~
Fichier Édition Onglets Aide
top - 08:56:39 up 16 min, 2 users, load average: 0,31, 0,16, 0,10
Tasks: 163 total, 1 running, 162 sleeping, 0 stopped, 0 zombie
%Cpu(s): 6,8 us, 3,4 sy, 0,0 ni, 89,8 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 1994,3 total, 913,1 free, 447,5 used, 633,7 buff/cache
MiB Swap: 2148,0 total, 2148,0 free, 0,0 used. 1344,7 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 2132 pi        20   0 607320 172924 129156 S   6,6   8,5   0:10.54 chromium
 2172 pi        20   0 432180 135712 114556 S   3,0   6,6   0:02.82 chromium
 1069 root       20   0 121292 41584  27340 S   1,0   2,0   0:04.43 Xorg
 2456 pi        20   0 509796 152084 98752 S   0,3   7,4   0:01.44 chromium
 2565 pi        20   0 10644 3624 3104 R   0,3   0,2   0:00.39 top
    1 root       20   0 34480 9096 7296 S   0,0   0,4   0:03.07 systemd
    2 root       20   0 0 0 0 S   0,0   0,0   0:00.00 kthreadd
    3 root       0 -20 0 0 0 I   0,0   0,0   0:00.00 rcu_gp
    4 root       0 -20 0 0 0 I   0,0   0,0   0:00.00 rcu_par_gp
    6 root       0 -20 0 0 0 I   0,0   0,0   0:00.00 kworker/0+
    7 root       20   0 0 0 0 I   0,0   0,0   0:00.00 kworker/u+
    8 root       0 -20 0 0 0 I   0,0   0,0   0:00.00 mm_percpu+
    9 root       20   0 0 0 0 S   0,0   0,0   0:00.14 ksoftirqd+
   10 root       20   0 0 0 0 I   0,0   0,0   0:00.21 rcu_sched
   11 root       20   0 0 0 0 I   0,0   0,0   0:00.00 rcu_bh
   12 root      rt   0 0 0 0 S   0,0   0,0   0:00.00 migration+
   13 root       20   0 0 0 0 I   0,0   0,0   0:00.49 kworker/0+
   14 root       20   0 0 0 0 S   0,0   0,0   0:00.00 cpuhp/0
   15 root       20   0 0 0 0 S   0,0   0,0   0:00.00 kdevtmpfs
   16 root       0 -20 0 0 0 I   0,0   0,0   0:00.00 netns

pi@raspberrypi:~$

```

Fermez votre navigateur Web et observez le résultat dans le terminal exécutant top.

Ouvrez votre navigateur Web et observez le résultat dans le terminal exécutant top.

Notez le PID des processus liés au fonctionnement du navigateur.

Utilisez la commande « kill » afin de supprimer le (ou les) processus lié(s) au fonctionnement du navigateur. Que constatez-vous ? Qu'est-ce que cela signifie pour le programme « chromium » ?

Le navigateur se ferme

Lycée d'enseignement général et technologique international Victor Hugo COLOMIERS		
	<b>Architectures matérielles et Systèmes d'Exploitation</b>  Cours & Activité Pratique	

## 2.4 Interblocage (« deadlock »)

*Qu'est-ce que c'est ?*

L'interblocage est un problème de blocage qui peut survenir lorsque les processus obtiennent des accès exclusifs aux ressources.

*Exemple*

Soit 2 processus P1 et P2, soit 2 ressources R1 et R2. Initialement, les 2 ressources sont "libres" (utilisées par aucun processus). Le processus P1 commence son exécution (état élu), il demande la ressource R1. Il obtient satisfaction puisque R1 est libre, P1 est donc dans l'état "prêt". Pendant ce temps, le système a passé P2 à l'état élu : P2 commence son exécution et demande la ressource R2. Il obtient immédiatement R2 puisque cette ressource était libre. P2 repasse immédiatement à l'état élu et poursuit son exécution (P1 lui est toujours dans l'état prêt). P2 demande la ressource R1, il se retrouve dans un état bloqué puisque la ressource R1 a été attribuée à P1 : P1 est dans l'état prêt, il n'a pas eu l'occasion de libérer la ressource R1 puisqu'il n'a pas eu l'occasion d'utiliser R1 (pour utiliser R1, P1 doit être dans l'état élu). P2 étant bloqué (en attente de R1), le système passe P1 dans l'état élu et avant de libérer R1, il demande à utiliser R2. Problème : R2 n'a pas encore été libéré par P2, R2 n'est donc pas disponible, P1 se retrouve bloqué.

Résumons la situation à cet instant : P1 possède la ressource R1 et se trouve dans l'état bloqué (attente de R2), P2 possède la ressource R2 et se trouve dans l'état bloqué (attente de R1)

Pour que P1 puisse poursuivre son exécution, il faut que P2 libère la ressource R2, mais P2 ne peut pas poursuivre son exécution (et donc libérer R2) puisqu'il est bloqué dans l'attente de R1. Pour que P2 puisse poursuivre son exécution, il faut que P1 libère la ressource R1, mais P1 ne peut pas poursuivre son exécution (et donc libérer R1) puisqu'il est bloqué dans l'attente de R2. Bref, la situation est totalement bloquée !

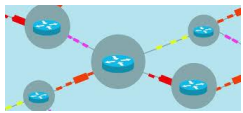
Cette situation est qualifiée d'interblocage (« deadlock » en anglais).

Il existe plusieurs solutions permettant soit de mettre fin à un interblocage (cela passe par l'arrêt d'un des processus fautifs) ou d'éviter les interblocage, mais ces solutions ne seront pas étudiées ici.

*Conditions nécessaires pour l'interblocage*

Pour qu'une situation d'interblocage ait lieu, les quatre conditions suivantes doivent être remplies (Conditions de Coffman) :

- L'exclusion mutuelle. A un instant précis, une ressource est allouée à un seul processus.
- La détention et l'attente. Les processus qui détiennent des ressources peuvent en demander d'autres.
- Pas de préemption. Les ressources allouées à un processus sont libérées uniquement par le processus.
- L'attente circulaire. Il existe une chaîne de deux ou plus processus de telle manière que chaque processus dans la chaîne requiert une ressource allouée au processus suivant dans la chaîne.



## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



## A faire vous-même III

- a. Imaginez la situation d'interblocage qui pourrait survenir lors de l'utilisation d'un SGBD. Supposez pour cela deux processus  $P_1$  et  $P_2$  qui demandent des accès exclusifs aux enregistrements  $R_1$  et  $R_2$  d'une base de données. Dans quelles conditions peut-il survenir un interblocage ? Expliquez la situation. Vous pouvez vous aider d'un schéma.

Il peut y avoir une situation d'interblocage si

- $P_1$  qui possède  $R_1$  demande l'accès à  $R_2$  qui est possédé par  $P_2$
- et que  $P_2$  demande l'accès à  $R_1$

$P_1$  et  $P_2$  seront alors en situation d'interblocage

- b. Circulation routière : imaginez la situation d'interblocage qui pourrait survenir lorsque des automobilistes empruntent le croisement de deux routes à double sens, dépourvu de signalisation. Est-ce que les quatre conditions de Coffman sont remplies ? Expliquez la situation. Vous pouvez vous aider d'un schéma.

- L'exclusion mutuelle : chaque route n'est "attribuée" à une seule voiture

- La détention et l'attente : chaque voiture attend pour changer de voie (tourner ou aller tout droit)


- Pas de préemption : chaque route est libérée uniquement par chacune des voitures respectives

- L'attente circulaire :  $V_1$  attend  $V_2$  (priorité à droite),  $V_2$  attend  $V_3$ ,  $V_3$  attend  $V_4$ ,  $V_4$  attend  $V_1$ , etc...

les conditions de Coffman sont remplies

  : voitures



Lycée d'enseignement général et technologique international Victor Hugo COLOMIERS		
	<b>Architectures matérielles et Systèmes d'Exploitation</b>  Cours & Activité Pratique	

## 3 Les protocoles de routage

Afin de bien profiter de cette activité, il est vivement conseillé de relire les concepts fondamentaux sur lesquels reposent les réseaux informatiques. N'hésitez pas à consulter les supports utilisés en classe de première :

- « Cours\_Decouverte\_Réseaux\_V1.pdf »
- « TP\_Decouverte\_Réseaux\_V1.pdf »

### 3.1 Introduction

Le routage est le mécanisme par lequel des chemins sont sélectionnés dans un réseau pour acheminer les données d'un expéditeur jusqu'à un ou plusieurs destinataires. Le routage est une tâche exécutée dans de nombreux réseaux, tels que le réseau téléphonique, les réseaux de données électroniques comme Internet, et les réseaux de transports.

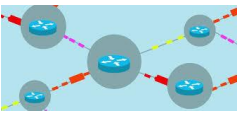
Pour effectuer le routage, on considère deux types de machines ou composants du réseau : **les routeurs**, qui servent d'intermédiaire dans la transmission d'un message, et les hôtes qui émettent ou reçoivent les messages.

Le réseau Internet peut être vu comme un réseau de réseaux : il résulte de l'interconnexion de réseaux par des routeurs. Un routeur permet de relier ensemble plusieurs réseaux locaux. Un routeur est composé d'un nombre plus ou moins important d'interfaces réseau (cartes réseau). Les routeurs les plus simples que l'on puisse rencontrer permettent de relier ensemble deux réseaux (ils possèdent alors 2 interfaces réseau), mais il existe des routeurs capables de relier ensemble une dizaine de réseaux. N'importe quel ordinateur peut jouer le rôle de routeur (à partir du moment où il possède au moins 2 interfaces réseau), mais on rencontre souvent des "machines" dédiées.

Le routage est un processus décentralisé, c'est-à-dire que chaque routeur possède des informations sur son voisinage. Chaque routeur maintient une liste des réseaux connus, chacun de ces réseaux étant associé à un ou plusieurs routeurs voisins à qui le message peut être passé. Cette liste s'appelle **la table de routage**.

### 3.2 Qu'est-ce qu'une route ?

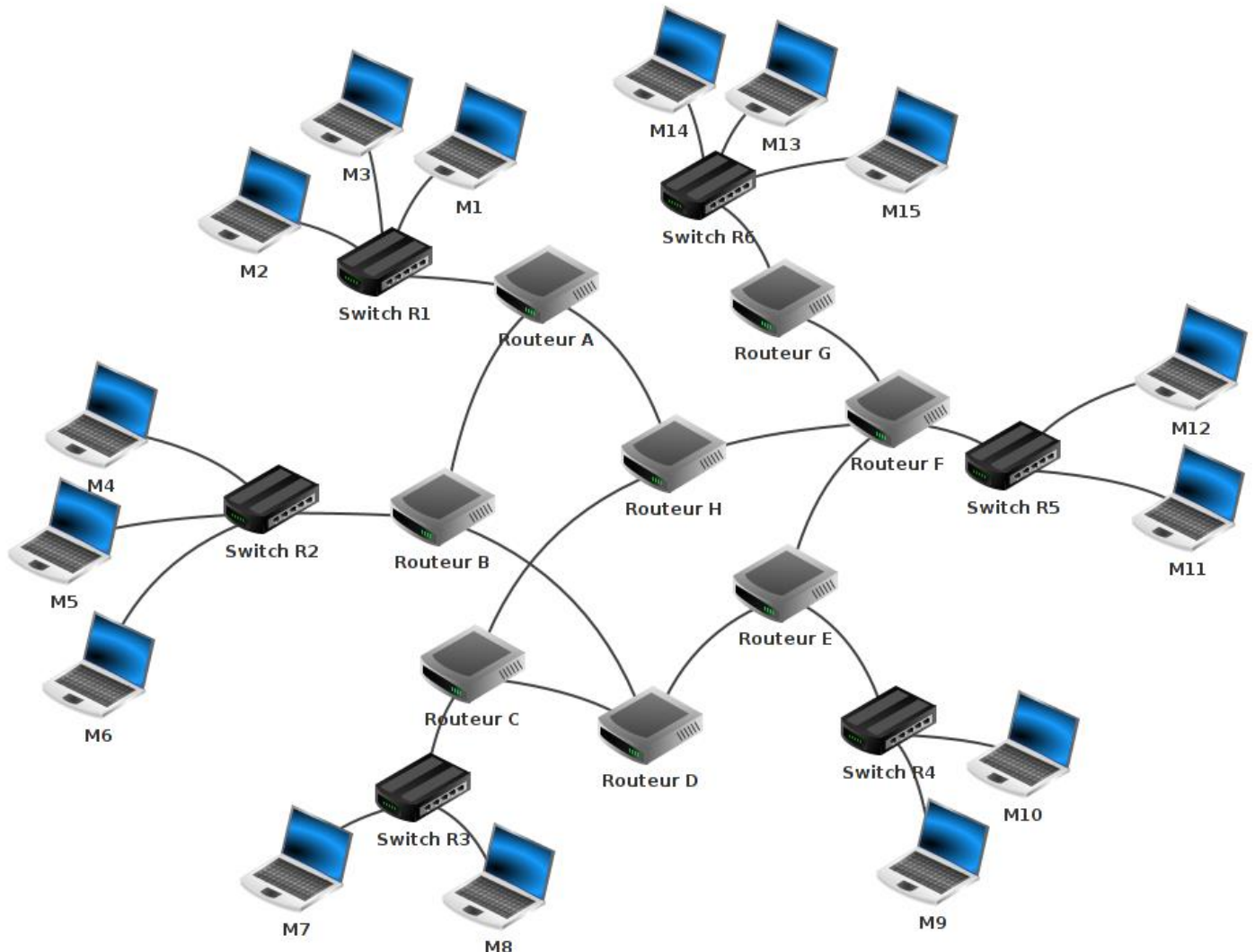
Dans un premier temps, considérons le schéma de configuration suivant :



## Architectures matérielles et Systèmes d'Exploitation



### Cours & Activité Pratique



Nous distinguons sur ce schéma les éléments suivants :

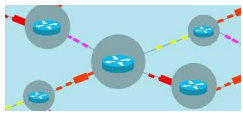
- 15 ordinateurs : M1 à M15
- 6 switches : R1 à R6
- 8 routeurs : A, B, C, D, E, F, G et H

L'analyse superficielle du schéma révèle la présence de six réseaux locaux. Chaque réseau local possède son propre switch.

Les ordinateurs M1, M2 et M3 appartiennent au réseau local 1. Les ordinateurs M4, M5 et M6 appartiennent au réseau local 2. Nous pouvons synthétiser tout cela ainsi :

- réseau local 1 : Switch R1, Machines M1, M2 et M3
- réseau local 2 : Switch R2, Machines M4, M5 et M6

## A Faire Vous-même IV



## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



Complétez la liste ci-dessus avec les réseaux locaux 3, 4, 5 et 6

Réseau local 3: Switch R3, Machines M7,M8  
Réseau local 4: Switch R4, Machines M9,M10  
Réseau local 5: Switch R5, Machines M11,M12  
Réseau local 6: Switch R6, Machines M13,M14,M15

## A Faire Vous-même V

Analysons quelques exemples de communication entre deux machines.

Cas n°1 : M1 veut communiquer avec M3

Le paquet est envoyé de M1 vers le switch R1, R1 "constate" que M3 se trouve bien dans le réseau local 1, le paquet est donc envoyé directement vers M3. On peut résumer le trajet du paquet par :

M1→R1→M3

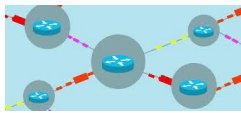
a. cas n°2 : M1 veut communiquer avec M6. Analysez cette communication puis donner le trajet du paquet.

Le paquet est envoyer de M1 vers le switch R1  
R1 envoie le paquet vers le routeur A  
Le routeur A envoie le paquet vers le routeur B  
Le routeur B envoie le paquet vers le switch R2  
Le switch R2 envoie le paquet vers la machine M6

M1 -> R1 -> A -> B -> R2 -> M6

Cas n°3 : M1 veut communiquer avec M9

M1 → R1 → Routeur A → Routeur B → Routeur D → Routeur E → R4 → M9



## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



b. Existe-t-il d'autres trajets possibles ? Proposez-en un.

M1 -> R1 -> A -> H -> F -> E -> R4 -> M9

c. Cas n°4 : M13 veut communiquer avec M9. Analysez cette communication puis donnez deux trajets possibles du paquet. Selon vous, quelle situation pourrait provoquer l'utilisation d'un deuxième trajet ?

M13 -> R6 -> G -> F -> E -> R4 -> M9

M13 -> R6 -> G -> F -> H -> C -> D -> E -> R4 -> M9

Le 2eme trajet peut etre utiliser si la connexion entre F et E ne marche plus

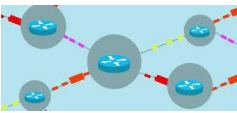
d. Cas n°5 : Déterminer deux chemins possibles permettant d'établir une connexion entre la machine M4 et M14.

M4 -> R2 -> B -> D -> E -> F -> G -> R6 -> M14

M4 -> R2 -> B -> A -> H -> F -> G -> R6 -> M14

## 3.3 L'acheminement des paquets

Comment les switches ou les routeurs procèdent-ils pour amener les paquets à bon port ?



## Architectures matérielles et Systèmes d'Exploitation

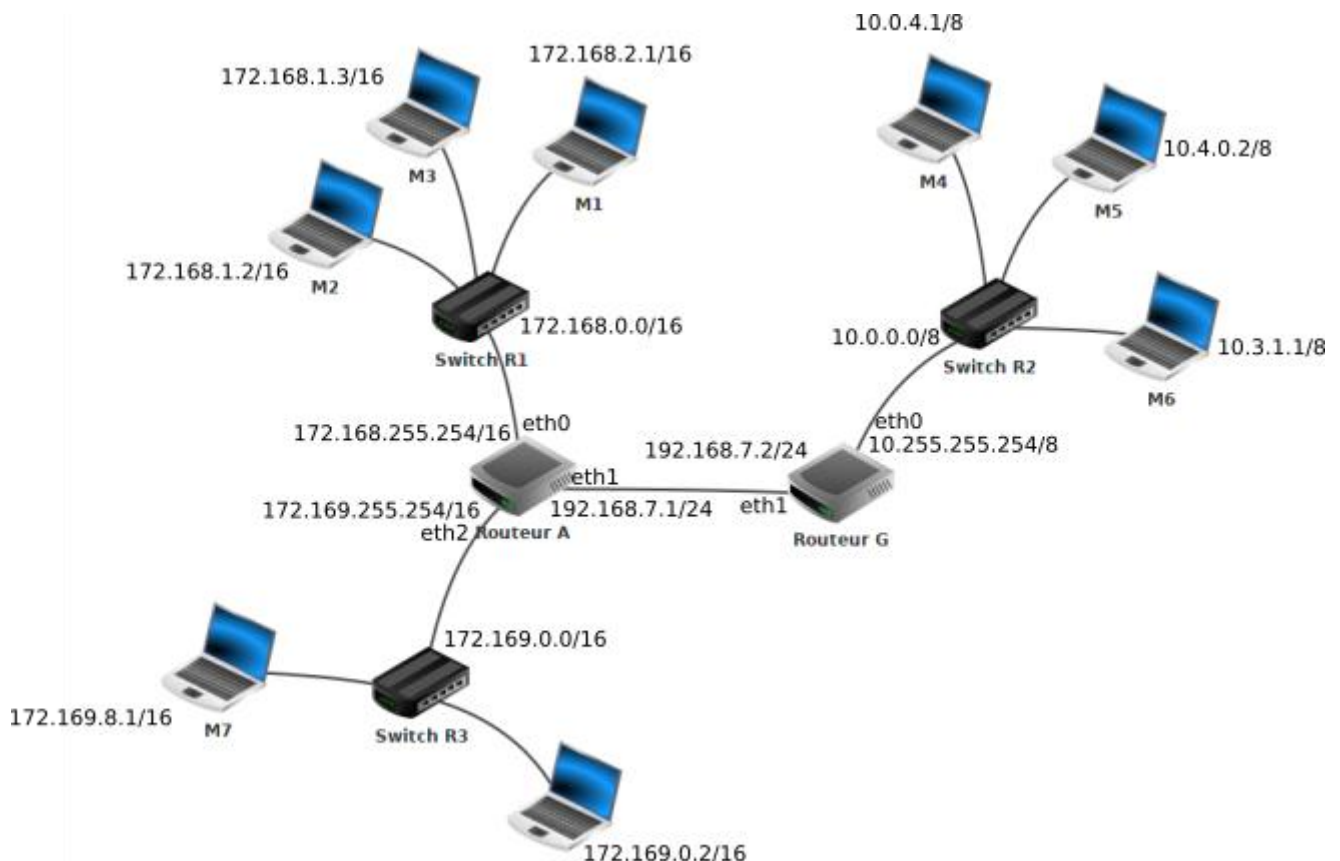
### Cours & Activité Pratique



En classe de Première, nous avons appris (cf. « Cours\_Decouverte\_Réseaux\_V1.pdf ») que deux machines appartenant au même réseau local doivent avoir la même adresse réseau. Or sur le schéma ci-dessus, nous constatons que M1 et M4 n'appartiennent pas au même réseau local. Par conséquent, 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 en lisant la valeur de son adresse IP. R1 va donc envoyer le paquet de données vers le routeur A. Cela sera donc au routeur A de gérer le "problème" : comment atteindre M4 ?

Chaque routeur possède une table de routage. Une table de routage peut être vue comme un tableau qui va contenir des informations permettant au routeur d'envoyer le paquet de données dans une direction qui permettra d'atteindre le bon destinataire.

Considérons maintenant le schéma de configuration suivant :



*NB : les adresses réseaux sont notées à côté des différents switches*



## A Faire Vous-même VI

- a. Étudiez attentivement le schéma ci-dessus. Sachant que les adresses IP sont fictives, vérifiez la cohérence des adresses machines avec les adresses réseaux.

### Analyse du schéma de configuration réseau

Nous avons deux routeurs A et G :

- Le routeur A 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 G 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)

Voici les informations présentes dans la table de routage de A :

- Le routeur A est directement relié au réseau 172.168.0.0/16 par l'intermédiaire de son interface eth0
- Le routeur A est directement relié au réseau 172.169.0.0/16 par l'intermédiaire de son interface eth2
- Le routeur A est directement relié au réseau 192.168.7.0/24 par l'intermédiaire de son interface eth1. Notons que le réseau 192.168.7.0/24 est un peu particulier car il est uniquement composé des routeurs A et G.
- 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 envoyés à la machine d'adresse IP 192.168.7.2/24 (c'est à dire le routeur G qui lui est directement relié au réseau 10.0.0.0/8)

On peut résumer tout cela avec le tableau suivant (table de routage simplifiée de A) :

Réseau	Moyen de l'atteindre	Métrieque
172.168.0.0/16	eth0	0
192.168.7.0/24	eth1	0
172.169.0.0/16	eth2	0
10.0.0.0/8	192.168.7.2/24	1

b. Sans tenir compte de la colonne « Métrieque », déterminez la table de routage du routeur G

Réseau	Moyen de l'atteindre	Métrieque
10.0.0.0/8....	eth..0	....
192.168.7..0	eth..1	....
172.168.0..0/16	192.168..7.1/24	....
172.169.0..0/16	192.168..7.1/24	....

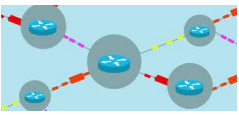


Dans des réseaux très complexes, chaque routeur aura une table de routage qui comportera de très nombreuses lignes (des dizaines, voire des centaines...). En effet chaque routeur devra savoir vers quelle interface réseau il faudra envoyer un paquet afin qu'il puisse atteindre sa destination. On peut trouver dans une table de routage plusieurs lignes pour une même destination, il peut en effet, à partir d'un routeur donné, 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". Pour choisir ce chemin le plus court, le routeur va utiliser la métrieque : plus la valeur de la métrieque est petite, plus le chemin pour atteindre le réseau est "court". Un réseau directement lié à un routeur aura une métrieque de 0.

*Comment un routeur arrive à remplir sa table de routage ?*

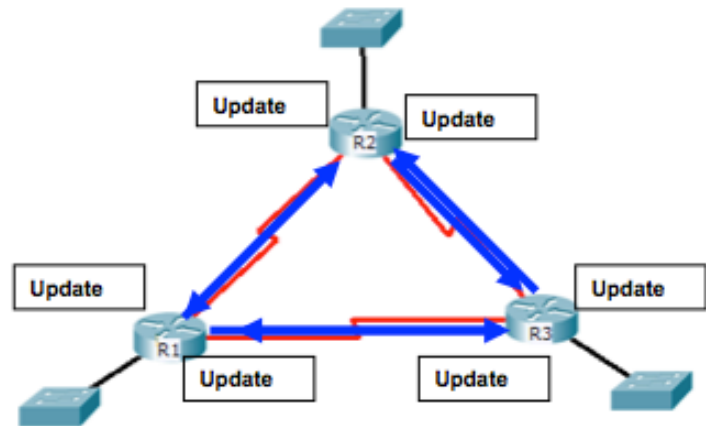
La réponse est simple pour les réseaux qui sont directement reliés au routeur (métrieque = 0), mais les autres réseaux (métrieque supérieure à zéro), il existe deux méthodes :

- 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 : tout se fait "automatiquement", on utilise des protocoles qui vont permettre de "découvrir" les différentes routes automatiquement afin de pouvoir remplir la table de routage tout aussi automatiquement.



## 3.4 Les protocoles de routage

Un protocole de routage spécifie comment les routeurs communiquent entre eux pour distribuer des informations qui leur permettent de sélectionner des routes entre deux sommets sur un réseau informatique. Les routeurs exécutent les fonctions de "direction du trafic" sur « Internet ». Les paquets de données sont transmis via les réseaux Internet d'un routeur à l'autre jusqu'à ce qu'ils atteignent leur ordinateur de destination.



Ces protocoles permettent de choisir, pour chaque destination possible, le « meilleur chemin ». La liste des meilleurs chemins, pour chaque destination, est appelée la « Table de Routage (Routing Table) ».

Un réseau de réseaux comportant des routeurs peut être modélisé par un graphe (cf. le support « SDD\_V1.pdf »). Chaque routeur est un sommet et chaque liaison entre les routeurs ou entre un routeur et un switch est une arête. Les algorithmes utilisés par les protocoles de routages sont donc des algorithmes issus de la théorie de graphes.

Plusieurs protocoles de routage existent. Nous allons en étudier deux : RIP (Routing Information Protocol) et OSPF (Open Shortest Path First).

### 3.4.1 Le protocole « RIP »

Au départ, les tables de routage contiennent uniquement les réseaux qui sont directement reliés au routeur (dans notre exemple ci-dessus, à l'origine, la table de routage de A contient uniquement les réseaux 172.168.0.0/16, 192.168.7.0/24 et 172.169.0.0/16). Chaque routeur envoie périodiquement (toutes les 30 secondes) à tous ses voisins (routeurs adjacents) un message. Ce message contient la liste de tous les réseaux qu'il connaît. Dans l'exemple ci-dessus, le routeur A envoie un message au routeur G avec les informations suivantes : "je connais les réseaux 172.168.0.0/16, 192.168.7.0/24 et 172.169.0.0/16". De la même manière G envoie un message à A avec les informations suivantes : "je connais les réseaux 192.168.7.0/24 et 10.0.0.0/8".

À la fin de cet échange, les routeurs mettent à jour leur table de routage avec les informations reçues. Dans l'exemple ci-dessus, le routeur A va pouvoir ajouter le réseau 10.0.0.0/8 à sa table de routage. Après cet ajout, il "sait" qu'un paquet à destination du réseau 10.0.0.0/8 devra transiter par le routeur G). Pour renseigner la colonne "métrique", le protocole utilise le nombre de sauts, autrement dit, le nombre de routeurs qui doivent être traversés pour atteindre le réseau cible (dans la table de routage de A, on aura donc une métrique de 1 pour le réseau 10.0.0.0/8 car depuis A il est nécessaire de traverser le routeur G pour atteindre le réseau 10.0.0.0/8).

Le protocole RIP s'appuie sur l'algorithme de Bellman-Ford (algorithme qui permet de calculer les plus courts chemins dans un graphe).



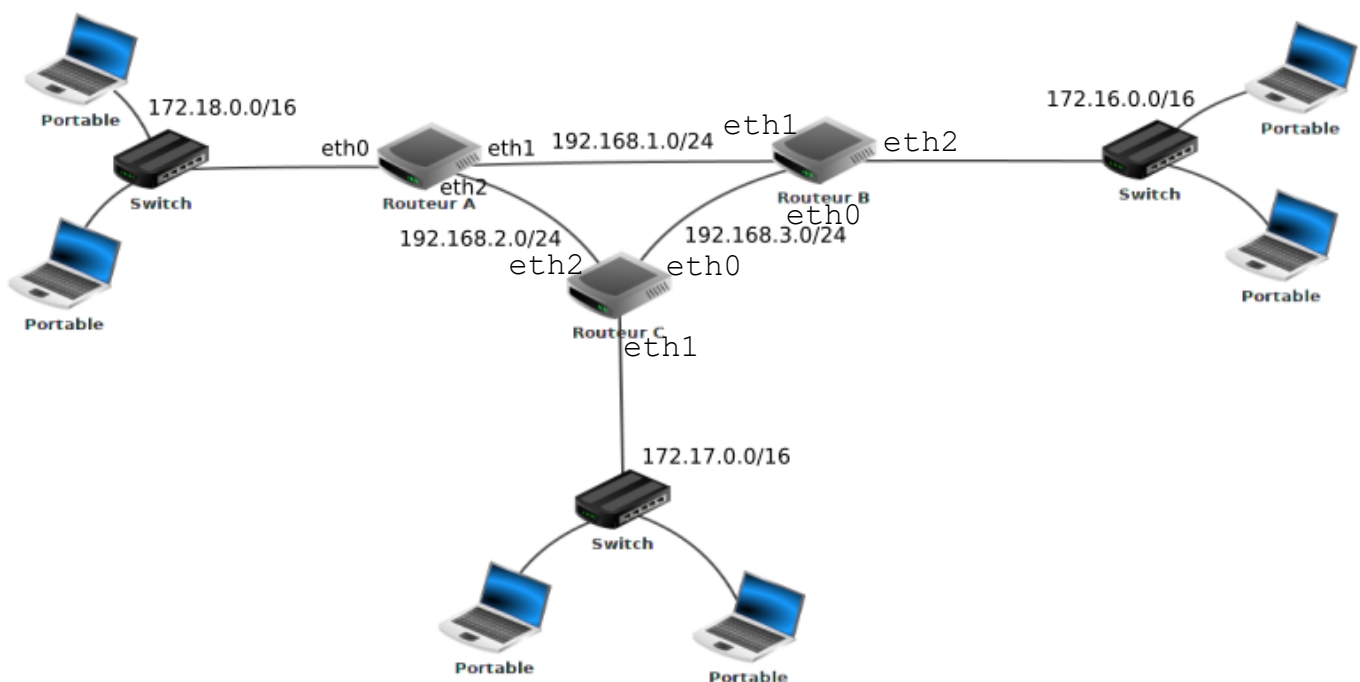
## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique

Le protocole RIP est aujourd'hui très rarement utilisé dans les grandes infrastructures. En effet, du fait de l'envoi périodique de message, il génère un trafic réseau important (surtout si les tables de routages contiennent beaucoup d'entrées). De plus, le protocole RIP est limité à 15 sauts (on traverse au maximum 15 routeurs pour atteindre sa destination). On lui préfère donc souvent le protocole OSPF.

### A Faire Vous-même VII

Considérons le schéma de configuration suivant :



- a. En vous basant sur le protocole RIP (métrique = nbre de sauts), déterminez la table de routage du routeur A

Réseau	Moyen de l'atteindre	Métrique
172.18.0.0/16	eth0	0
192.168.1.0/24	eth1	0
192.168.2.0/24	eth2	0
172.16.0.0/16	192.168.1.0/24	1
172.17.0.0/16	192.168.2.0/24	1
....	....	....

- b. Quel est, d'après la table de routage construite ci-dessus, le chemin qui sera emprunté par un paquet pour aller d'une machine ayant pour adresse IP 172.18.1.1/16 à une machine ayant pour adresse IP 172.16.5.3/16 ?

```
-> switch (172.18.0.0/16) -> A -> B -> switch(172.16.0.0/16) ->
```

### 3.4.2 Le protocole OSPF

Comme dans le cas du protocole RIP, nous allons retrouver des échanges d'informations entre les routeurs. Cependant, ces échanges sont plus "intelligents" dans le cas d'OSPF l'occupation du réseau est moindre. Nous n'allons pas rentrer dans les détails de ces échanges et nous allons principalement insister sur la métrique produite par OSPF.

Le protocole OSPF, au contraire de RIP, n'utilise pas le "nombre de sauts nécessaire" pour établir la métrique, mais la notion de "coût des routes". Dans les messages échangés par les routeurs, on trouve le coût de chaque liaison (plus le coût est grand et moins la liaison est intéressante). Quand on parle de "liaison" on parle simplement du câble qui relie un routeur à un autre routeur. Le protocole OSPF permet de connaître le coût de chaque liaison entre routeurs, et donc, de connaître le coût d'une route (en ajoutant le coût de chaque liaison traversée). On notera que pour effectuer ces calculs, le protocole OSPF s'appuie sur l'algorithme de Dijkstra.

La notion de coût est directement liée au débit des liaisons entre les routeurs. Le débit correspond au nombre de bits de données qu'il est possible de faire passer dans un réseau par seconde. Le débit est donc donné en bits par seconde (bps), mais on trouve souvent des kilo bits par seconde (kbps) ou encore des méga bits par seconde (Mbps) => 1 kbps = 1000 bps et 1 Mbps = 1000 kbps. Connaissant le débit d'une liaison, il est possible de calculer le coût d'une liaison à l'aide de la formule suivante :

$$\text{Coût} = 10^8 / \text{débit (bps)}$$

Pour obtenir la métrique d'une route, il suffit d'additionner les coûts de chaque liaison. Par exemple, si pour aller d'un réseau 1 à un réseau 2, on doit traverser une liaison de coût 1, puis une liaison de coût 10 et enfin une liaison de coût 1 ; la métrique de cette route sera de  $1 + 10 + 1 = 12$ .

Évidemment, comme dans le cas de RIP, les routes ayant les métriques les plus faibles sont privilégiées.



## A Faire Vous-même VIII

Considérons le schéma de configuration du AFVM VII

- a. En vous basant sur le protocole OSPF (métrique = somme des coûts), déterminez la table de routage du routeur A

On donne les débits suivants :

	172.18.0.0/16	eth 0	0
	192.168.1.0/24	eth 1	0
	192.168.2.0/24	eth 2	0

- Liaison routeur A - routeur B : 1 Mbps
- Liaison routeur A - routeur C : 10 Mbps
- Liaison routeur C - routeur B : 10 Mbps

Réseau	Moyen de l'atteindre	Métrique	
172.16.0.0/16	Routeur B	100	
172.17.0.0/16	Routeur C	10	
172.16.0.0/16	Routeur A-C-B	20	(10+10)
172.17.0.0/16	Routeur A-B-C	110	(100+10)

- b. Quel est, d'après la table de routage construite ci-dessus, le chemin qui sera emprunté par un paquet pour aller d'une machine ayant pour adresse IP 172.18.1.1/16 à une machine ayant pour adresse IP 172.16.5.3/16 ?

switch -> Routeur A -> Routeur C -> Routeur B -> switch

Lycée d'enseignement général et technologique international Victor Hugo COLOMIERS		
	Architectures matérielles et Systèmes d'Exploitation Cours & Activité Pratique	

## 4 La sécurisation des communications

Soit deux individus A et B qui cherchent à s'envoyer des messages par l'intermédiaire d'un réseau informatique. A et B désirent qu'une tierce personne (par exemple P) ne soit pas capable de lire les messages si par hasard ces derniers devaient être interceptés par P. Pour ce faire, A va chiffrer le message. Toute personne qui ne possèdera pas le moyen de déchiffrer ce message chiffré se verra dans l'impossibilité de comprendre le contenu du message.

En effet, si P intercepte le message chiffré et qu'il ne possède pas le moyen de déchiffrer ce message, l'interception aura été totalement inutile puisque P sera dans l'incapacité de comprendre le contenu du message.

Il existe deux types de chiffrement : le chiffrement symétrique et le chiffrement asymétrique.

### 4.1 Le chiffrement symétrique

Pour chiffrer un message, A va utiliser une suite de caractère que l'on appelle "clé de chiffrement". Dans le cas du chiffrement symétrique, cette clé de chiffrement sera aussi utilisée par B pour déchiffrer le message envoyé par A. Dans ce cas, la clé de chiffrement est identique à la clé de déchiffrement.

L'idée de chiffrer des messages (de les rendre illisibles pour des personnes non autorisées) ne date pas du début de l'ère de l'informatique. En effet, dès l'antiquité, on cherchait déjà à sécuriser les communications en chiffrant les messages sensibles (pour en savoir plus sur l'histoire du chiffrement, n'hésitez pas à consulter la page Wikipédia consacrée à ce sujet).

Nous nous intéresserons ici uniquement aux communications ayant lieu par l'intermédiaire d'un réseau informatique. Comme nous avons déjà eu l'occasion de le voir en première, toute "donnée informatique" peut être vue comme une suite de « zéro » et de « un ». Nous chercherons donc à chiffrer une suite de « zéro » et de « un ».

#### A Faire Vous-même IX

Soit le « messageC » codé en binaire ASCII « Vive les Sciences du Numérique ! » :

« 010101100110100101110110011001010010000011101100011001010111001100100000010100110110001101101001010110111001100011011001010111001100100000011001000111010100100000010011100111010101101101110000111010100101110010011010010111000101110101011001010010000000100001 »

Ici, nous avons simplement utilisé le code ASCII de chaque caractère.

- Identifiez et extraire le code ASCII de la chaîne « les » contenue dans l'expression « Vive les Sciences du Numérique ! ». Pour effectuer la "conversion" texte vers code binaire ASCII ou vice versa, vous pouvez utiliser le site « [ce site](#) »

```
les=01101100 01100101 01110011
```

Choisissons maintenant un mot (ou une phrase) qui nous servira de clé de chiffrement, prenons pour exemple le mot "NSI".

b. Coder la clé « NSI » en binaire ASCII

```
01001110 01010011 01001001
```

Pour chiffrer le message nous allons effectuer un « OU exclusif » (XOR) bit à bit. Pour rappel, vous trouverez la table de vérité du XOR ci-dessous :

X	Y	S = X ou Y
0	0	0
0	1	1
1	0	1
1	1	0

Dans notre exemple, la clé étant plus courte que le message, il faut "reproduire" la clé vers la droite autant de fois que nécessaire. Si la taille du message n'est pas un multiple de la taille de la clé, on peut reproduire seulement quelques bits de la clé pour la fin du message.

c. A l'aide d'un éditeur de texte, superposez les codes binaires du message et de la clé. Compte tenu de la taille du message, cette superposition tient sur plusieurs lignes. Prenez soin de bien respecter le colonage.

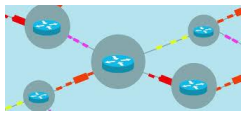
```
0101011001101001011101100110010100100000011011000110010101110011001000000101001101100011011010
```

```
0101100101011011100110001101100101011100110010000001100100011101010010000001001110011101010110
```

```
1101110000111010100101110010011010010111000101110101011001010010000000100001
```

d. A l'aide de la [calculatrice](#), faites l'opération « messageC » XOR « clé ». Notez ci-dessous le chiffrement obtenu.

```
00011000 00111010 00111111 00101011 01110011 00100101 00101011 00100000
01101001 00011101 00110000 00100000 00101011 00111101 00101010 00101011
00100000 01101001 00101010 00100110 01101001 00000000 00100110 00100100
10001101 11111010 00111011 00100111 00100010 00111100 00101011 01110011
01101000
```



## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



- e. Tenter d'afficher le résultat du chiffrement d. avec un éditeur de texte, par exemple [celui-ci](#). Quel résultat obtenez-vous ? Comprenez-vous ce message ? Selon vous, ce message peut-il être exploité par P ?

```
:?+s%+ i 0 +=*+ i*&i&$;'"<+sh
```

Ce message n'est pas exploitable

- f. B a maintenant reçu le message chiffré, il possède la clé et va donc pouvoir déchiffrer le message en appliquant un XOR entre le message chiffré et la clé. A l'aide de la [calculatrice](#), vérifiez que B a bien reçu le « messageC ». Pourquoi n'est-ce pas tout à fait le cas ? Expliquez et proposez une solution afin de retrouver exactement le même message.

```
000110000011101000111110010101101110011001001010010101100100000011010010001110100110000001000
000010101100111101001010100010101100100000011010010010101000100110011010010000000001001100010
0100100011011111101000111011001001110010001000111100001010110111001101101000
```

⊕

<b>Lycée d'enseignement général et technologique international Victor Hugo</b> COLOMIERS		<b>NSI</b> NUMÉRIQUE ET SCIENCES INFORMATIQUES
	<b>Architectures matérielles et Systèmes d'Exploitation</b>  Cours & Activité Pratique	

## A Faire Vous-même X

Vous allez jouer le rôle de A. Choisissez une ou un camarade dans la classe qui jouera le rôle de B. Mettez-vous d'accord avec B sur une clé de chiffrement/déchiffrement en choisissant un mot qui jouera le rôle de clé. Ce mot ne doit être connu que de vous deux, A et B. Choisissez un message à faire parvenir à B puis procédez au chiffrement de ce message. Faites attention à l'encodage utilisé si votre message utilise des caractères accentués.

Choisissez P, une ou un deuxième camarade dans la classe. P ne possèdera pas la clé de chiffrement envoyée à B.

En utilisant la messagerie électronique de l'ENT, faites parvenir à B et à P, le résultat du chiffrement (en binaire). P devra décoder (binaire à texte) le message avant de le transmettre à B. B devra déchiffrer le message à l'aide de la clé. B répondra (par mail) à A et à P en leur signifiant s'il a pu interpréter leur message.



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.

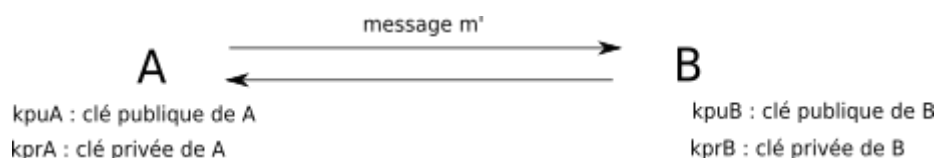
Le gros problème avec le chiffrement symétrique, c'est qu'il est nécessaire pour A et B de se mettre d'accord à l'avance sur la clé qui sera utilisée lors des échanges. Le chiffrement asymétrique permet d'éviter ce problème.

## 4.2 Le chiffrement asymétrique

Dans le cas du chiffrement asymétrique A et B n'ont pas besoin de partager une "clé secrète".

A possède une "clé privée" que l'on notera  $k_{prA}$  et une "clé publique" que l'on notera  $k_{puA}$ . En aucun cas A ne devra divulguer sa clé privée à quiconque, elle devra rester strictement secrète. En revanche sa clé publique pourra être connue de tout le monde sans aucun problème.

B possède une "clé privée" que l'on notera  $k_{prB}$  et une "clé publique" que l'on notera  $k_{puB}$ . En aucun cas B ne devra divulguer sa clé privée à quiconque, elle devra rester strictement secrète. En revanche sa clé publique pourra être connue de tout le monde sans aucun problème.





Lycée d'enseignement général et technologique international Victor Hugo COLOMIERS		NSI NUMÉRIQUE ET SCIENCES INFORMATIQUES
	<b>Architectures matérielles et Systèmes d'Exploitation</b>  Cours & Activité Pratique	

Si A désire envoyer un message « m » à B, il va utiliser la clé publique de B afin de réaliser le chiffrement (m est chiffré en m'). Le message chiffré « m' » va ensuite pouvoir transiter entre A et B. Une fois le message « m' » en sa possession, B 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 suivant :

- 1)  $k_{puB}(m) \rightarrow m'$
- 2)  $A \xrightarrow[\text{réseau}]{m'} B$
- 3)  $k_{prB}(m') \rightarrow m$

Si P intercepte le message m', il sera incapable de déterminer m à partir de « m' » sans la clé privée de B.

Le chiffrement asymétrique repose sur des problèmes très difficiles à résoudre dans un sens et faciles à résoudre dans l'autre sens. Prenons un exemple : l'algorithme de chiffrement asymétrique RSA (du nom de ses 3 inventeurs : Rivest Shamir et Adleman), est très couramment utilisé, notamment dans tout ce qui touche au 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 est facile de déterminer C le produit de A par B (ici on a  $A \times B = C$  avec  $C = 4338219660050123$ ). En revanche si je vous donne C (ici 4338219660050123) il est très difficile de retrouver A et B. A ce jour, aucun algorithme n'est capable de retrouver A et B connaissant C dans un temps "raisonnable". 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 relativement complexes 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 quasiment impossible de trouver la clé privée de quelqu'un à partir de sa clé publique.

## 4.3 Le protocole HTTPS

Nous allons maintenant voir une utilisation concrète de ces chiffrements symétriques et asymétriques : le protocole HTTPS.

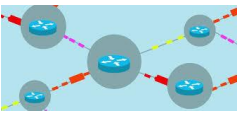
Avant de parler du protocole HTTPS, rappelons brièvement ce qu'est le protocole HTTP.

### 4.3.1 HTTP

**HTTP** (pour HyperText Transfer Protocol) est le protocole de communication communément utilisé pour transférer les ressources du Web. HTTPS est la variante avec authentification et chiffrement.

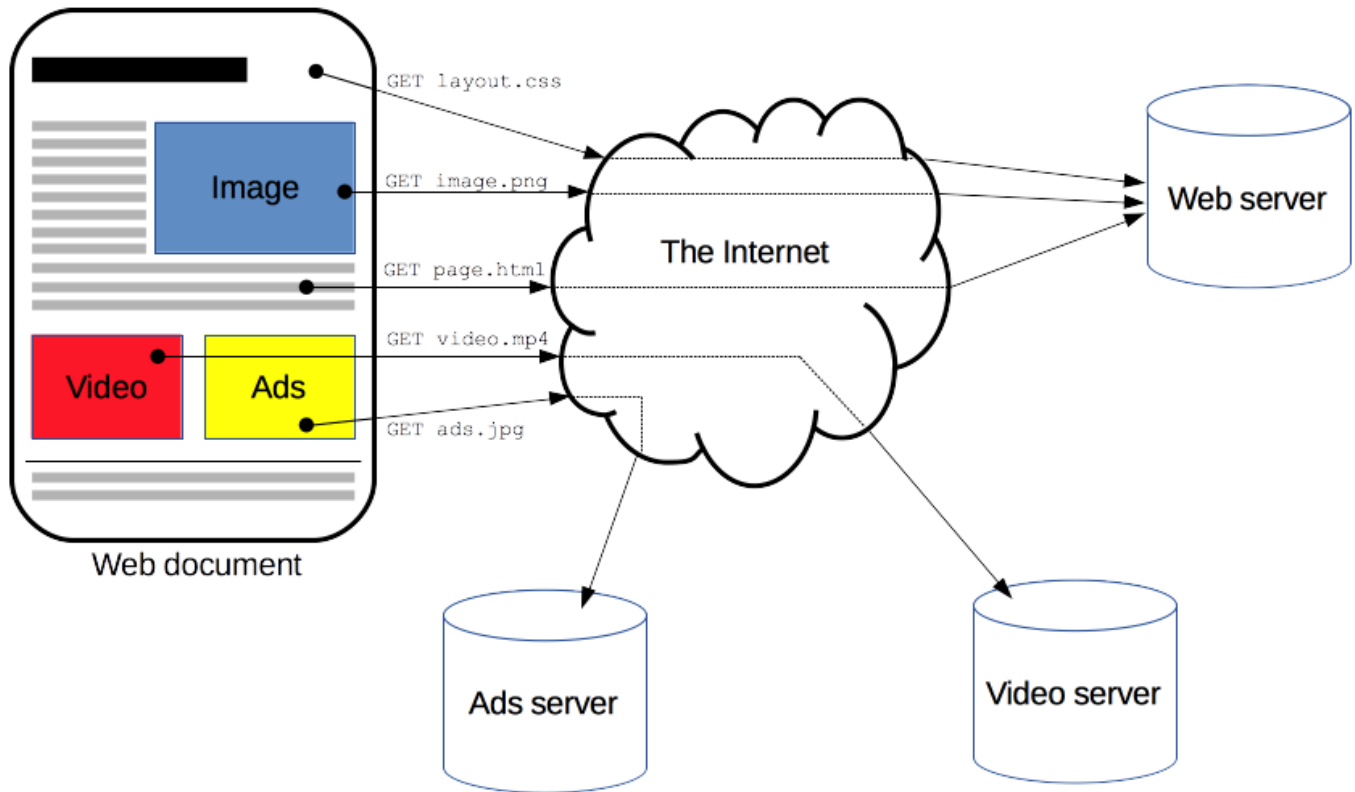
Ce protocole qui permet de récupérer des ressources telles que des documents HTML. Il est à la base de tout échange de données sur le Web. C'est un protocole de type client-serveur (cf. page suivante), ce qui signifie que les requêtes sont initiées par le destinataire (qui est généralement un navigateur web).

Un document complet est construit à partir de différents sous-documents qui sont récupérés, par exemple du texte, des descriptions de mise en page, des images, des vidéos, des scripts et bien plus.



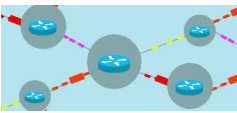
## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



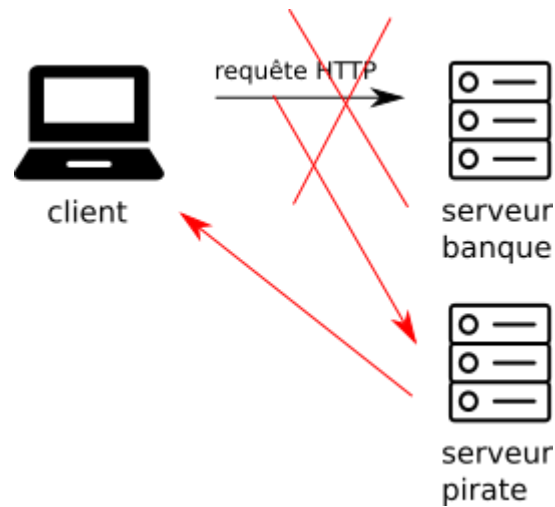
Le protocole HTTP pose 2 problèmes en termes de sécurité informatique :

- 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 qui ne sera pas détaillée ici (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 la situation suivante : 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.



## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



### 4.3.2 HTTPS

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

Nous allons nous pencher sur problème du chiffrement des données circulant entre le client et le serveur.

Les communications vont être chiffrées grâce à une clé symétrique. Problème : comment échanger cette clé entre le client et le serveur ? Simplement en utilisant une paire clé publique / clé privée !

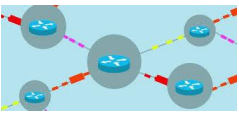
Voici le déroulement des opérations :

- 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 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 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).

### A Faire Vous-même XI

Reprendre la suite des opérations précédentes en complétant le schéma « vierge » suivant :





## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



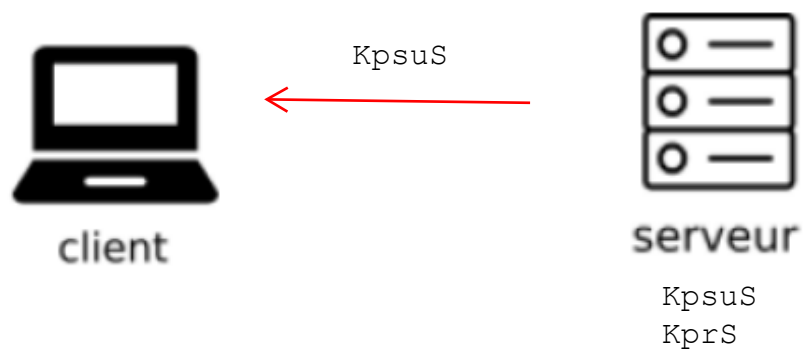
Sur chaque schéma, vous préciserez :

- La nature des clés utilisées ainsi que les opérations réalisées de chaque côté,
- La nature et sens des éventuels échanges entre client et serveur

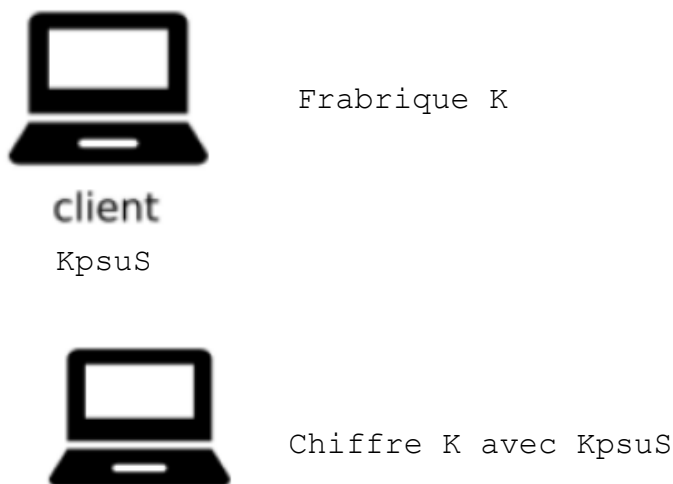
Vous réaliserez la suite des opérations en cinq schémas après le premier schéma suivant :



1)

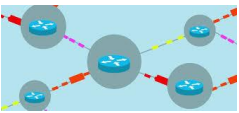


2)



3)

client  
KpsuS  
K



## Architectures matérielles et Systèmes d'Exploitation

### Cours & Activité Pratique



4)



client  
KpsuS  
K

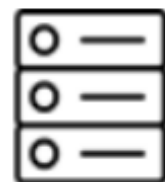
K (chiffre par KpsuS)



serveur  
KpsuS  
KprS

5)

Dechiffre K avec KprS



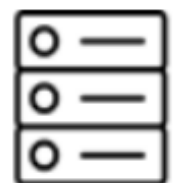
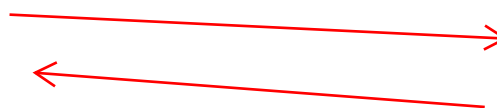
serveur  
KpsuS  
KprS  
K

6)



client  
KpsuS  
K

Echange de donnés chiffre avec K



serveur  
KpsuS  
KprS  
K

<div>Lycée d'enseignement général et technologique international Victor Hugo COLOMIERS</div>		<div></div>	<div></div>
<div></div>	<div>Architectures matérielles et Systèmes d'Exploitation</div>		<div></div>
<div>Cours &amp; Activité Pratique</div>			

Ce processus se répète à chaque fois qu'un nouveau client effectue une requête HTTPS vers le serveur.

Comment éviter les conséquences fâcheuses d'une attaque 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 ce faire, chaque site désirant proposer des transactions HTTPS doit, périodiquement, demander (acheter dans la plupart des cas) un certificat d'authentification (sorte de carte d'identité pour un site internet) auprès d'une autorité habilitée à fournir ce genre de certificats (chaque navigateur web possède une liste des autorités dont il accepte les certificats). Comme dit plus haut, ce certificat permet au site de prouver son "identité" auprès des clients. Nous n'allons pas entrer dans les détails du fonctionnement de ces certificats, mais vous devez juste savoir que le serveur envoie ce certificat au client en même temps que sa clé publique (étape 2 du schéma ci-dessus). 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), dans ce cas, le navigateur web côté client affichera une page de mise en garde avec un message du style "ATTENTION le certificat d'authentification du site XXX a expiré, il serait prudent de ne pas poursuivre vos échanges avec le site XXXX".

\*\*\*\* *Fin du Document Support* \*\*\*\*