

Support de cours Réseaux

Dr Yani-Athmane BENNAI

Université de Béjaia

Faculté des Sciences Exactes

Département d'Informatique

27 décembre 2023

Table des matières

| | | |
|----------|---|----------|
| 2 | La couche Réseau | 3 |
| 2.1 | Introduction | 3 |
| 2.2 | Responsabilités de la couche réseau | 3 |
| 2.3 | Fragmentation | 4 |
| 2.4 | Le protocole IP | 6 |
| 2.4.1 | Le datagramme IP | 7 |
| 2.5 | Le routage | 9 |
| 2.5.1 | Routage à vecteur de distance | 10 |
| | Algorithme de Bellman-Ford | 11 |
| | Protocole de routage RIP | 12 |
| 2.5.2 | Routage à états de liens | 14 |
| | Construction des tables de routage | 14 |
| | Algorithme de Dijkstra | 15 |
| | Protocole OSPF | 16 |

Table des figures

| | | |
|------|---|----|
| 2.1 | Internet est un réseau de réseaux | 3 |
| 2.2 | Principe de la MTU | 5 |
| 2.3 | Réplication de l'entete dans chaque fragment | 6 |
| 2.4 | Format de l'entête IP | 7 |
| 2.5 | Le champs FLAGS | 8 |
| 2.6 | Internet est divisé en AS (Systèmes Autonomes) | 10 |
| 2.7 | Types de protocoles de routage : Vecteur de distance et Etat de liens | 10 |
| 2.8 | Exemple de Bellman-Ford | 11 |
| 2.9 | Tableau de Bellman-Ford | 11 |
| 2.10 | Format de l'entête RIP | 12 |
| 2.11 | Les 3 minuteriers (Timers) de RIP | 13 |
| 2.12 | Algorithme de Dijkstra | 15 |
| 2.13 | Déroulement de l'algorithme de Dijkstra | 16 |
| 2.14 | Division de chaque AS en zones (Areas) | 17 |
| 2.15 | lien point-à-point | 17 |
| 2.16 | lien transitoire | 18 |
| 2.17 | lien stub (extrémité) | 18 |
| 2.18 | lien virtuel | 19 |

Chapitre 2

La couche Réseau

2.1 Introduction

Internet est un vaste réseau qui relie un grand nombre d'ordinateurs à travers le planète. Pour réaliser cette interconnexion complexe, de nombreux réseaux locaux (LAN) et réseaux étendus (WAN) sont utilisés. La couche réseau joue un rôle essentiel dans la gestion de la connectivité au sein d'Internet.

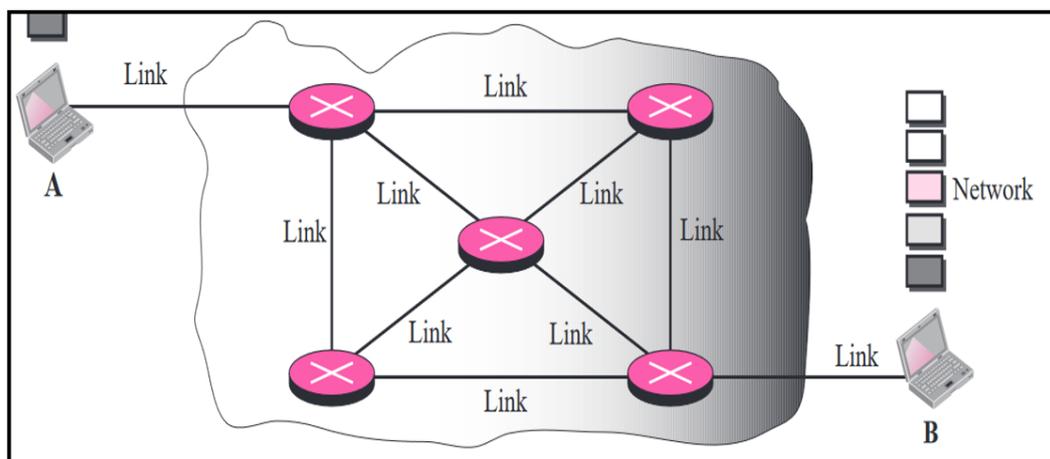


FIGURE 2.1 – Internet est un réseau de réseaux

2.2 Responsabilités de la couche réseau

La couche réseau est responsable de l'encapsulation des données dans un datagramme, qui contient des informations essentielles pour le routage des paquets. Lorsque les données doivent être transmises à travers plusieurs réseaux pour atteindre leur destination, la couche réseau joue un rôle crucial dans la détermination du prochain saut (le routeur suivant) vers lequel les données doivent être acheminées.

Voici les éléments clés du fonctionnement de la couche Réseau :

- Encapsulation : La couche réseau ajoute un en-tête aux données, notamment l'adresse source et destination du paquet, la longueur des données et éventuellement des options.
- Table de routage : L'ordinateur source consulte une table de routage pour déterminer l'adresse IP du prochain saut. Cette table contient des informations sur la manière de diriger les paquets vers différentes destinations. Elle est généralement mise à jour dynamiquement pour refléter l'état du réseau.
- Routage : Une fois que l'adresse du prochain saut est déterminée, la couche réseau transmet le datagramme au routeur approprié. Le routeur est chargé de faire progresser le datagramme vers la destination en fonction du protocole de routage utilisé.

Une fois que la couche réseau a déterminé l'adresse IP du prochain saut, la couche liaison de données est chargée de l'acheminement réel des données vers cette destination. Cependant, pour que la couche liaison de données puisse accomplir cette tâche, elle a besoin de l'adresse MAC (adresse physique) du prochain saut.

Pour obtenir l'adresse MAC correspondant à l'adresse IP du prochain saut, la couche réseau utilise le protocole de résolution d'adresses (ARP, Address Resolution Protocol), de cette manière :

1. Requête ARP : Lorsque la couche réseau a besoin de connaître l'adresse MAC correspondant à une adresse IP spécifique, elle envoie une requête ARP. Cette requête ARP contient l'adresse IP du prochain saut pour laquelle l'adresse MAC est nécessaire.
2. Réponse ARP : Tous les appareils du réseau local reçoivent cette requête ARP, mais seul l'appareil qui possède l'adresse recherchée y répond. Cet appareil envoie une réponse ARP contenant l'adresse MAC.
3. Mise en cache : Une fois que la réponse ARP est reçue, la couche réseau conserve cette information dans une table de cache ARP. Cela permet d'éviter de futures requêtes ARP pour la même adresse IP, car l'adresse MAC correspondante est désormais connue et les adresses MAC ne changent pas.
4. Acheminement : Avec l'adresse MAC du prochain saut maintenant disponible dans la table de cache ARP, la couche liaison de données peut acheminer les données vers sa destination en utilisant l'adresse MAC correcte.

2.3 Fragmentation

La fragmentation des datagrammes est une étape importante du processus de transmission de données, en particulier dans les réseaux où la taille maximale des trames est

limitée, comme c'est le cas pour de nombreux réseaux locaux, car rappelez vous, les datagrammes créés par la couche Réseau sont ensuite envoyés vers la couche liaison.

La fragmentation est basée sur certains concepts clés :

- Limite de taille (MTU) : Chaque réseau a une taille maximale de trame qu'il peut transporter, appelée MTU (Maximum Transmission Unit). Les datagrammes IP, générés par la couche réseau, peuvent parfois dépasser cette taille limite.
- Fragmentation : Lorsque la couche réseau prépare un datagramme pour l'envoyer sur un réseau dont la MTU est plus petite que la taille du datagramme, elle doit le fragmenter. Cela signifie qu'elle divise le datagramme en fragments plus petits, tout en préservant les informations de l'en-tête du datagramme d'origine.
- Répétition de l'en-tête : Chaque fragment doit contenir une copie de l'en-tête du datagramme d'origine, de manière à ce que le récepteur puisse comprendre comment réassembler les fragments.
- Ajout d'informations de position : Chaque fragment contient également des informations supplémentaires pour indiquer sa position dans l'ensemble du datagramme. Cela permet au récepteur de réassembler correctement les fragments dans le bon ordre.
- Assemblage : Lorsque les fragments arrivent à l'ordinateur de destination, il doit les réassembler pour reconstituer le datagramme d'origine. Cela nécessite de prendre en compte les informations de position ajoutées lors de la fragmentation.
- Minuteur de réassemblage : Un minuteur de réassemblage est généralement défini pour s'assurer que tous les fragments sont reçus et assemblés en temps voulu. Si le minuteur expire avant que tous les fragments ne soient reçus, ils sont détruits, et un message d'erreur peut être renvoyé pour indiquer que le datagramme fragmenté doit être renvoyé.

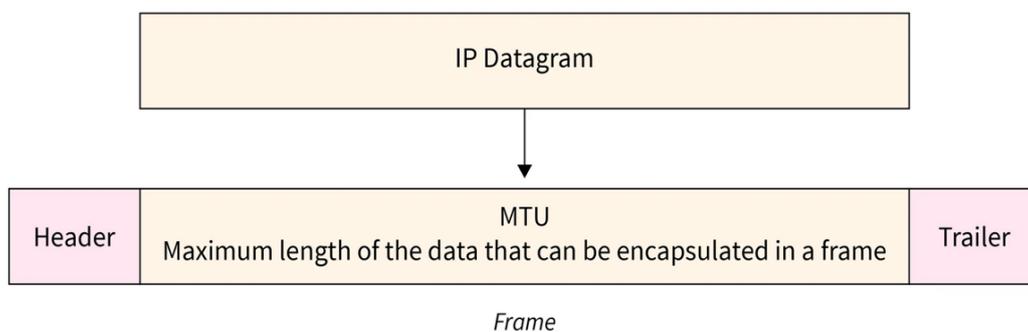


FIGURE 2.2 – Principe de la MTU

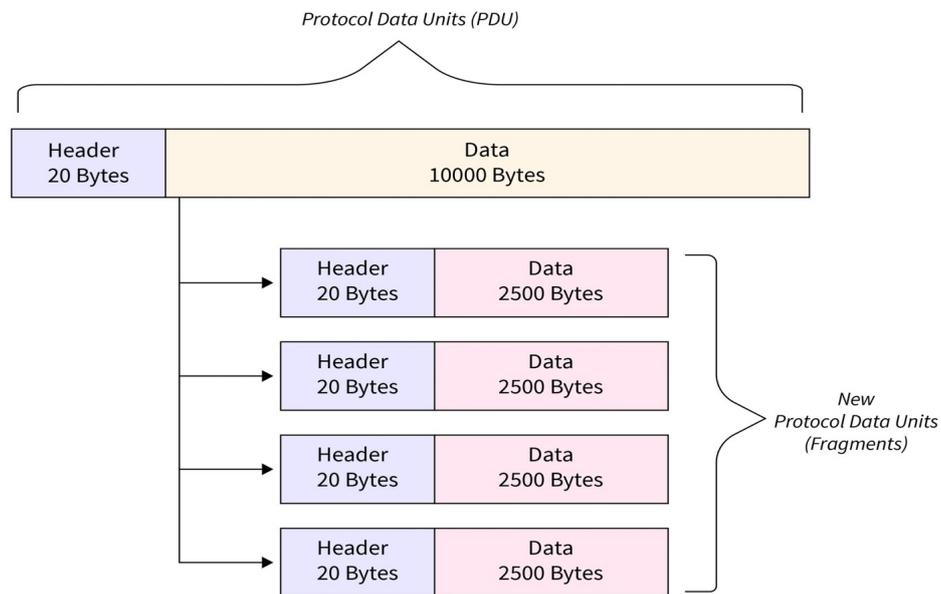


FIGURE 2.3 – Réplication de l’entête dans chaque fragment

2.4 Le protocole IP

Le protocole Internet (Internet Protocol, IP) est le protocole de base de la couche réseau du modèle TCP/IP. Il est responsable de l’acheminement des données entre deux hôtes sur un réseau.

IP est un protocole de datagramme, ce qui signifie qu’il découpe les données à transmettre en petits morceaux appelés paquets. Chaque paquet est ensuite acheminé indépendamment vers le destinataire.

IP est un protocole non fiable, ce qui signifie qu’il ne garantit pas la livraison des données. Les paquets peuvent être perdus, corrompus ou arriver dans le désordre.

IP est un protocole de best effort, ce qui signifie qu’il fait de son mieux pour acheminer les données, mais ne garantit pas le succès.

Pourquoi IP est-il non fiable ?

- Les réseaux peuvent subir des erreurs, telles que les collisions.
- Les paquets peuvent être perdus ou corrompus en route.

- Les routeurs peuvent choisir des routes inefficaces, ce qui peut entraîner des retards sur le réseau.

2.4.1 Le datagramme IP

Un datagramme est essentiellement un paquet d'informations de taille variable, composé de deux parties : l'en-tête et les données.

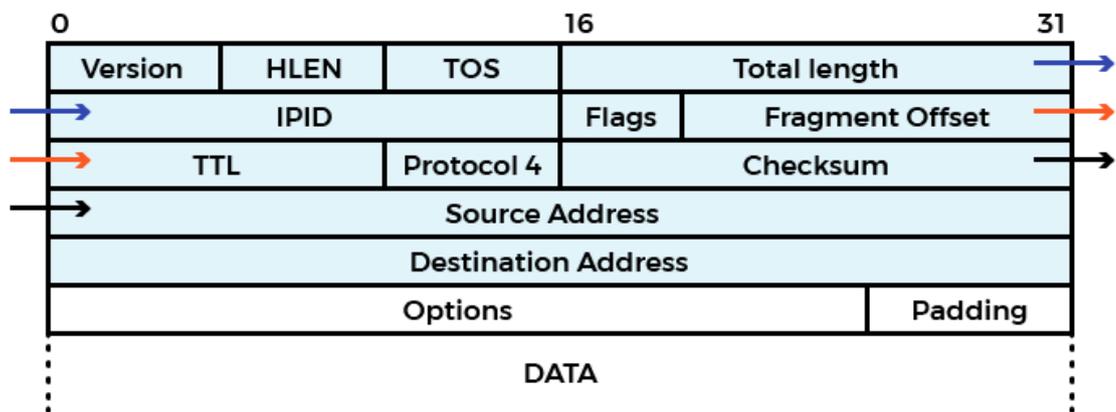


FIGURE 2.4 – Format de l'en-tête IP

L'en-tête, d'une longueur de 20 à 60 octets, joue un rôle crucial. C'est comme le dos d'une lettre de poste qui contient des détails importants tels que l'adresse de destination, etc. L'en-tête contient des instructions pour le routage, et d'autres informations nécessaires à la transmission correcte des données.

1. **Le champ "Version" (VER)** : est un code de 4 bits qui indique la version du protocole IP utilisée. Actuellement, la version la plus courante est la version 4 (IPv4). Il est important de noter que bien qu'IPv4 soit largement utilisé, il existe une version plus récente appelée IPv6 (version 6).
2. **Le champ "Longueur de l'en-tête" (HLEN)** : est crucial pour déterminer la taille totale de l'en-tête du datagramme, exprimée en mots de 4 octets (32 bits). Ce champ est nécessaire en raison de la variabilité de la longueur de l'en-tête, qui peut varier entre 20 et 60 octets.

Ce champ représente le nombre de lignes dans l'en-tête (car chaque ligne est sur 32 bits). Lorsqu'il n'y a pas d'options présentes, la longueur de l'en-tête est fixée à 20 octets, et la valeur du champ HLEN est alors de 5 (car 5 fois 4 équivaut à 20).

En revanche, lorsque le champ des options atteint sa taille maximale, la valeur du champ HLEN est de 15 (car 15 fois 4 équivaut à 60).

3. **Le champ "Type de service" TOS** : Il indique si le datagramme est prioritaire, urgent et s'il nécessite une transmission fiable. En résumé, il détermine les caractéristiques de traitement du datagramme pendant son passage à travers le réseau.
4. **Le champ "Longueur totale"** : définit la taille complète du datagramme en octets, comprenant à la fois l'en-tête et les données. Pour calculer la longueur des données, on soustrait simplement la longueur de l'en-tête de la longueur totale.

La longueur de l'en-tête se trouve en multipliant la valeur dans le champ HLEN par 4, car chaque mot représente 4 octets. Il est important de noter que, étant donné que la longueur du champ est de 16 bits, la longueur totale du datagramme ne peut pas dépasser 2^{16} octets.

5. **Le champ "Identification" IPID** : attribue une étiquette unique à chaque datagramme. Il s'agit d'un numéro qui s'incrémente à chaque nouvel envoi de datagramme.

En cas de fragmentation d'un datagramme, la valeur du champ d'identification est copiée dans tous les fragments. Ainsi, tous les fragments portent le même numéro d'identification. Cette caractéristique facilite le processus de réassemblage à la destination. La destination sait que tous les fragments partageant la même valeur d'identification doivent être rassemblés pour reconstituer le datagramme d'origine.

6. **Le champ "Flags"** : comporte trois bits. Le premier bit est réservé et inutilisé. Le deuxième bit, appelé "Do Not Fragment" (Ne pas fragmenter), a une valeur de 1 pour indiquer que la machine ne doit pas fragmenter le datagramme, et une valeur de 0 signifie que le datagramme peut être fragmenté si nécessaire.

Le troisième bit, appelé "More Fragments" (Plus de fragments), est important lorsqu'un datagramme est fragmenté. Si sa valeur est 1, cela indique que le datagramme n'est pas le dernier fragment, et d'autres fragments suivent. Si sa valeur est 0, cela signifie que le fragment actuel est le dernier ou le seul fragment.

D: Do not fragment
M: More fragments



FIGURE 2.5 – Le champs FLAGS

7. **Le champ "Fragment Offset"** : indique la position d'un fragment par rapport au datagramme complet. Ce décalage représente le déplacement des données dans le datagramme d'origine, mesuré en unités de 8 octets.

Par exemple, considérons un datagramme de 4000 octets fragmenté en trois parties :

- Le Fragment 1 (0 à 1399 octets) a un décalage de $0/8 = 0$.
- Le Fragment 2 (1400 à 2799 octets) a un décalage de $1400/8 = 175$.
- Le Fragment 3 (2800 à 3999 octets) a un décalage de $2800/8 = 350$.

Ces valeurs de décalage aident à reconstruire le datagramme original en indiquant où chaque fragment commence dans le datagramme complet

8. **Le champ TTL (Time to Live)** : indique le nombre maximum de routeurs que le datagramme peut traverser avant d'être détruit. Ce champ permet d'éviter que des datagrammes ne circulent indéfiniment dans le réseau, ce qui pourrait saturer le réseau.
9. **Le champ "Protocole"** : dans un datagramme IP indique le protocole de niveau supérieur utilisé. Ce champ permet de savoir quel protocole de communication est utilisé au-dessus de la couche réseau.

Un datagramme IP a la capacité d'encapsuler des données provenant de différents protocoles de niveau supérieur, tels que TCP et UDP.

10. **Le champ "Checksum"** : joue un rôle dans le contrôle d'erreurs. Il permet de vérifier l'intégrité des données du datagramme.
11. **le champ "Adresse source"** : est composé de 32 bits et indique l'adresse IP de la source, c'est-à-dire l'origine du datagramme.
12. **le champ "Adresse de destination"** également sur 32 bits, indique l'adresse IP du destinataire.
13. **Le champ "Options"** : dans un datagramme IP est facultatif et offre la possibilité d'inclure diverses informations personnalisées. Ces informations peuvent inclure des indications de routage spécifiques, des directives de sécurité, ou d'autres paramètres personnalisés.
14. **Dans le champ "Padding"** : , des bits de bourrage sont ajoutés pour garantir que la longueur totale de l'en-tête reste toujours un multiple de 32.

2.5 Le routage

Internet est si vaste qu'un seul protocole ne peut pas gérer toutes les mises à jour de routage. Internet est divisé en systèmes autonomes (AS), qui sont des groupes de ré-

seaux.

Le routage à l'intérieur d'un AS est appelé routage intra-domaine.

Le routage entre les AS est appelé routage inter-domaine.

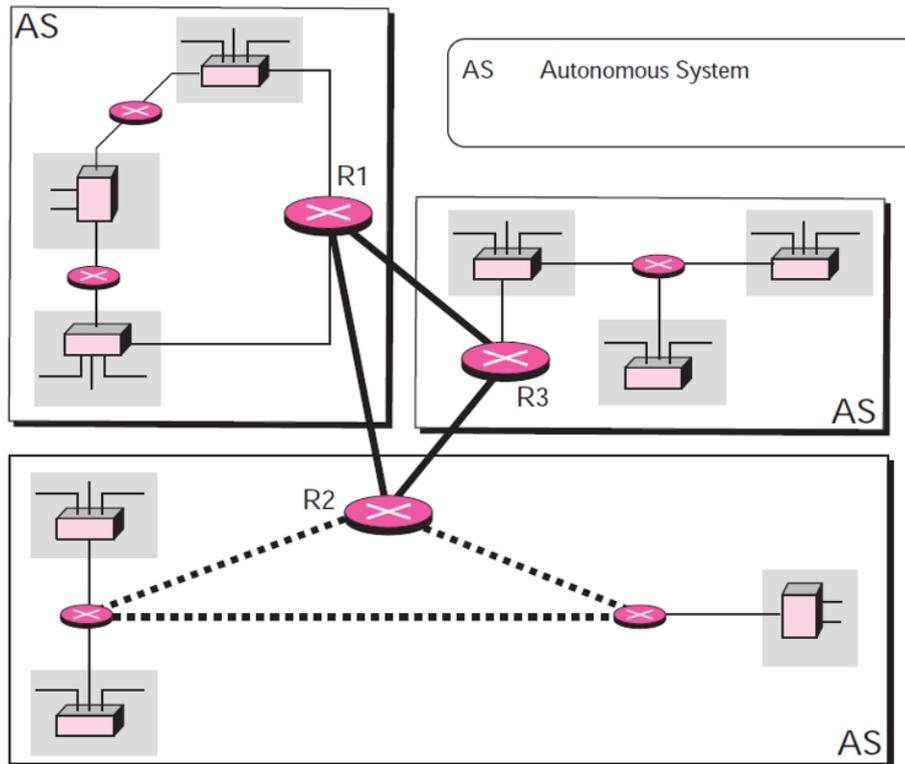


FIGURE 2.6 – Internet est divisé en AS (Systèmes Autonomes)

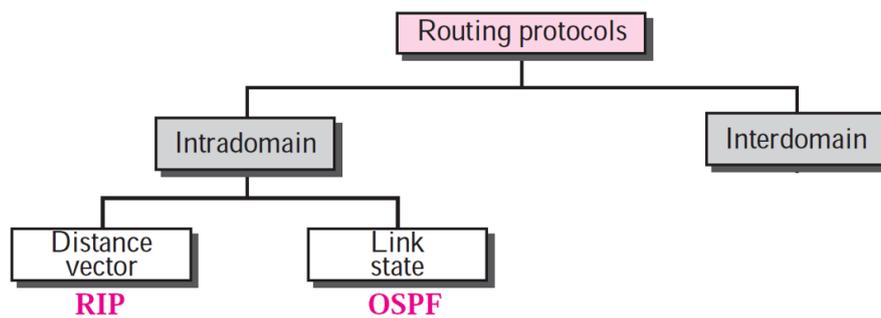


FIGURE 2.7 – Types de protocoles de routage : Vecteur de distance et Etat de liens

2.5.1 Routage à vecteur de distance

Le vecteur de distance est une méthode de routage qui envisage le réseau comme un graphe, composé de nœuds (points) et de liens (connexions). La théorie des graphes

utilise un algorithme appelé Bellman-Ford pour déterminer le chemin le plus court entre les nœuds de ce graphe.

Algorithme de Bellman-Ford

L'algorithme fonctionne par itérations, c'est à dire qu'on doit répéter le déroulement un certain nombre de fois, le nombre max d'itération est $n-1$, en sachant que n est le nombre de noeuds dans le graphe.

On arrête l'algorithme soit lorsqu'on a terminé toutes les itérations, soit lorsqu'on constate que deux itérations consécutives nous donnent les mêmes résultats (pas de changements dans les valeurs).

Un tableau est utilisé, tous les noeuds sont inscrits sur le tableau, en commençant toujours par le noeuds source à gauche du tableau, chaque itération se termine lorsqu'on a parcouru toutes le colonnes, c'est à dire tous les noeuds de gauche à droite.

Prenons un exemple :

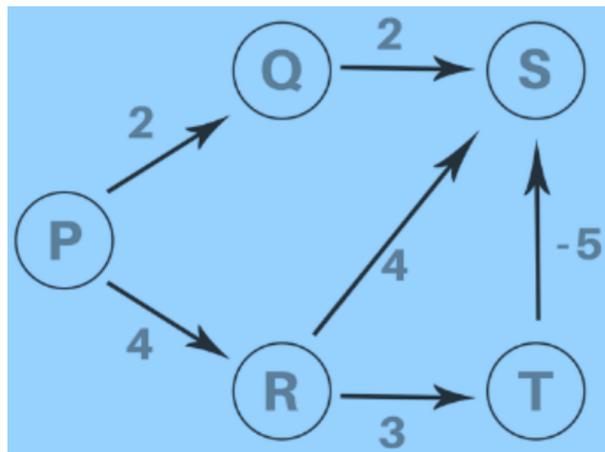


FIGURE 2.8 – Exemple de Bellman-Ford

| P | Q | R | S | T | |
|----------|----------|----------|--------------|----------|----------------|
| 0 | ∞ | ∞ | ∞ | ∞ | Initialisation |
| 0 | 2 | 4 | 2 | 7 | Itération 1 |
| 0 | 2 | 4 | 2 | 7 | Itération 2 |

FIGURE 2.9 – Tableau de Bellman-Ford

Vu que le graphe a 5 noeuds, nous savons que le nombre max d'itérations est de 4.

On commence notre première itération avec le noeud de départ P, on examine les arcs qui sortent de P, on peut voir qu'il y en a deux, on inscrit alors les valeurs 2 et 4 dans les colonnes correspondant à Q et R.

On en a terminé avec P pour cette itération, on se déplace à droite sur notre tableau, nous sommes maintenant au niveau de Q, il n'y a qu'un seul arc sortant de Q, on met alors à jour la colonne de S avec la valeur 4 (4 vient du fait d'ajouter la valeur actuelle de Q dans le tableau qui est 2, plus la valeur de l'arc qui sort de Q vers S qui est de 2 également).

On passe maintenant à R, il y'a un arc qui sort vers S, le cout serait de 8, c'est supérieur à la valeur actuelle dans S alors on ne fait rien. un autre arc sort vers T avec un cout total de 7, on met à jour la valeur de T dans le tableau avec 7.

On passe au noeud S, il n'y a aucun arc sortant, donc on ne fait rien.

On termine avec le noeud T, il ya un seul arc et il va vers S, le cout total est de $7+(-5) = 2$, ce qui est inférieur à la valeur actuelle dans la colonne de S, on met donc à jour avec la valeur 2.

La première itération est terminée, on se déplace alors de gauche à droite sur notre tableau et on refait la même chose pour P, Q, R, puis S et enfin T, cela va former notre deuxième itération, et ainsi de suite.

On constate dans notre cas que les valeurs n'ont pas changé entre la première et la deuxième itération, l'algorithme est donc terminé car ça ne sert à rien de continuer.

Protocole de routage RIP

RIP, un protocole de routage utilisé à l'intérieur d'un système autonome (AS), considère les routeurs comme des nœuds. La distance est mesurée en "hop count", représentant le nombre de sauts (routeurs) nécessaires pour atteindre la destination. L'infini est fixé à 16, signifiant qu'aucune route ne peut avoir plus de 15 sauts selon ce protocole.

| Command | Version | Reserved |
|-----------------|---------|----------|
| Family | | All 0s |
| Network address | | |
| All 0s | | |
| All 0s | | |
| Distance | | |

FIGURE 2.10 – Format de l'entête RIP

Commande (Command) : Un champ sur 8 bits qui spécifie le type de message : demande (1) ou réponse (2).

Version : Il existe deux versions de RIP.

Famille (Family) : Ce champ définit la famille du protocole utilisé. Pour TCP/IP, la valeur est toujours de 2.

Adresse du Réseau : Ce champ définit l'adresse du réseau de destination.

Distance : Il représente le nombre de sauts (coût) nécessaires pour atteindre le routeur de destination.

Dans le cadre de RIP, il existe deux types de messages : demande (request) et réponse (response).

Demande (Request) : Un routeur envoie un message de demande lorsqu'il démarre ou s'il a des entrées en attente de mise à jour sur sa table de routage.

Réponse (Response) : Une réponse peut être sollicitée ou non sollicitée. Une réponse sollicitée est envoyée uniquement en réponse à une demande. Une réponse non sollicitée est envoyée périodiquement toutes les 25-35 secondes ou en cas de changements dans le réseau.

RIP fonctionne avec 3 minuteriers :

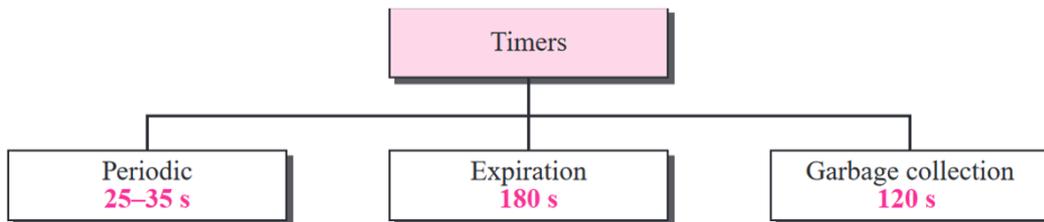


FIGURE 2.11 – Les 3 minuteriers (Timers) de RIP

1. la minuterie périodique : utilisée pour contrôler l'envoi régulier des messages de mise à jour. Pour réduire les envois simultanés, elle utilise un nombre aléatoire compris entre 25 et 35 secondes.

Le fonctionnement de la minuterie est basé sur un compte à rebours. Lorsqu'elle atteint zéro, chaque routeur envoie son message de mise à jour, et ensuite la minuterie est réinitialisée avec une nouvelle valeur aléatoire entre 25 et 35.

2. La minuterie d'expiration :

Réglage Initial : Lorsqu'un routeur reçoit des informations sur une route, la minuterie est initialement réglée à 180 secondes pour cette route.

Réinitialisation : Chaque fois qu'une nouvelle mise à jour est reçue, la minuterie est réinitialisée à 180.

Expiration de la Route : En l'absence de mises à jour pendant 180 secondes, la route est considérée comme expirée. Le nombre de sauts de la route est alors fixé à 16, indiquant qu'il faut l'éviter.

3. la minuterie de collecte des déchets :

Après l'Expiration de la Route : Une fois qu'une route a expiré (une fois que son nombre de sauts a été fixé à 16), la minuterie de collecte des déchets est réglée à 120 secondes pour cette route spécifique.

Suppression de la Route : Lorsque le compte à rebours atteint zéro, la route est définitivement supprimée. Cette minuterie permet aux routeurs de se préparer car la route est sur le point d'être retirée, il ont ainsi le temps de changer de route.

2.5.2 Routage à états de liens

Dans le routage par état de lien, chaque routeur possède une connaissance complète de la topologie. Cela englobe la liste de tous les routeurs et des liens, avec leur coût respectif et leur état (actif ou inactif).

Grâce à ces informations, chaque nœud peut utiliser l'algorithme du plus court chemin (Dijkstra) pour construire une table de routage.

Construction des tables de routage

Dans le routage par état de lien, la construction des tables de routage suit ces étapes :

1. Création des États des Liens (LSP) : Chaque routeur crée des états pour ses liens, regroupés sous forme de paquets d'état de lien (LSP).
 2. Diffusion des LSP à Chaque Routeur (Inondation) : Les LSP sont diffusés à chaque routeur, processus appelé inondation, assurant que chaque routeur a une vue complète des états de tous les liens.
 3. Calcul d'une Table de Routage basée sur l'Arbre de Chemin le Plus Court : En se basant sur l'arbre de chemin le plus court, chaque nœud calcule une table de routage optimale, facilitant ainsi la prise de décision lors de l'acheminement des paquets dans le réseau.
-

Contenu du LSP : Il comprend l'identité du routeur, un numéro de séquence, et un âge. Le numéro de séquence est utilisé pour différencier les nouveaux LSP, tandis que l'âge empêche que des informations trop vieilles restent dans le réseau.

Le processus d'inondation est très simple :

Envoi d'une Copie par Chaque Interface : Le routeur envoie une copie du LSP par chaque interface qu'il possède.

Comparaison des LSP : Un routeur recevant un LSP le compare avec sa copie existante. S'il est plus ancien, il est rejeté. S'il est plus récent, le nœud suit ces étapes :

- a. Il rejette l'ancien LSP.
- b. Il envoie une copie par chaque interface, sauf celle par laquelle le paquet est arrivé.

Cela assure que l'inondation des LSP s'arrête quelque part.

Algorithme de Dijkstra

Pour expliquer le fonctionnement de l'algorithme, considérons le réseau suivant :

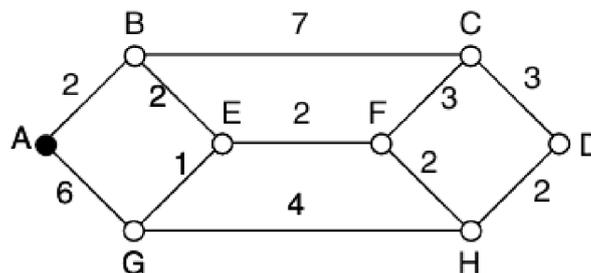


FIGURE 2.12 – Algorithme de Dijkstra

Nous voulons trouver le plus court chemin de A à D.

- Chaque voisin de A est analysé puis étiqueté provisoirement avec sa distance par rapport à lui
- Lorsque l'étiquette d'un nœud est modifiée on inscrit également le nœud à partir duquel on vient d'arriver afin de pouvoir reconstruire le parcours final.
- Une fois que tous les nœuds adjacents à A sont traités, toutes les étiquettes provisoires sont examinées, l'étiquette portant la valeur la plus basse devient active (on se déplace vers le nœud actif).

- Nous partons maintenant du nouveau nœud actif soit B, Une fois les nœuds adjacents à B sont examinés, le nœud ayant la plus petite étiquette devient permanent. Et ainsi de suite.

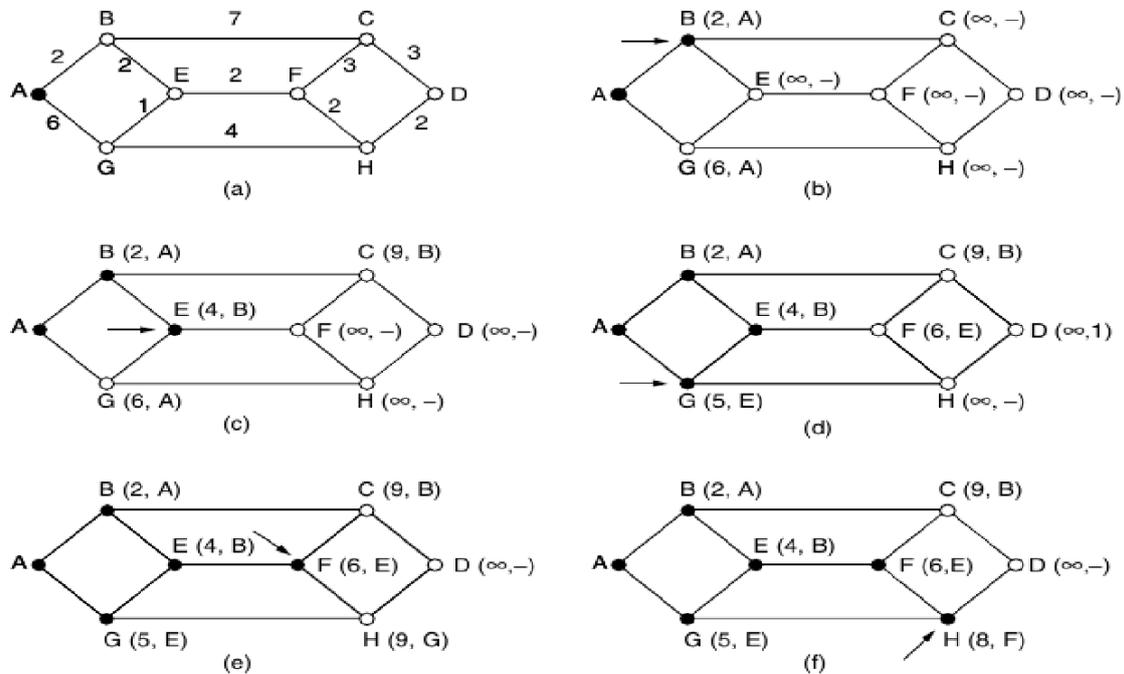


FIGURE 2.13 – Déroulement de l'algorithme de Dijkstra

Protocole OSPF

Le protocole OSPF (Open Shortest Path First) est un protocole de routage intradomaine basé sur le routage par état de lien.

Zones : Pour une gestion efficace du routage, OSPF divise chaque système autonome en zones. Une zone est une collection d'ordinateurs et de routeurs. Tous les réseaux à l'intérieur d'une zone doivent être connectés. Cette organisation par zones facilite la gestion et la hiérarchisation des informations dans un grand réseau.

Dans OSPF, la diffusion de l'information se fait de la manière suivante :

Dans une Zone : Les routeurs à l'intérieur d'une zone inondent d'informations, assurant que chaque routeur a une vue complète de la topologie de la zone.

Frontières de Zone : Aux frontières d'une zone, des routeurs spéciaux appelés routeurs de bord de zone récapitulent les informations et les envoient à d'autres zones.

Zone Principale (Backbone) : Toutes les zones doivent être connectées au backbone (zone principale). Le backbone OSPF assure la connectivité entre toutes les zones, créant

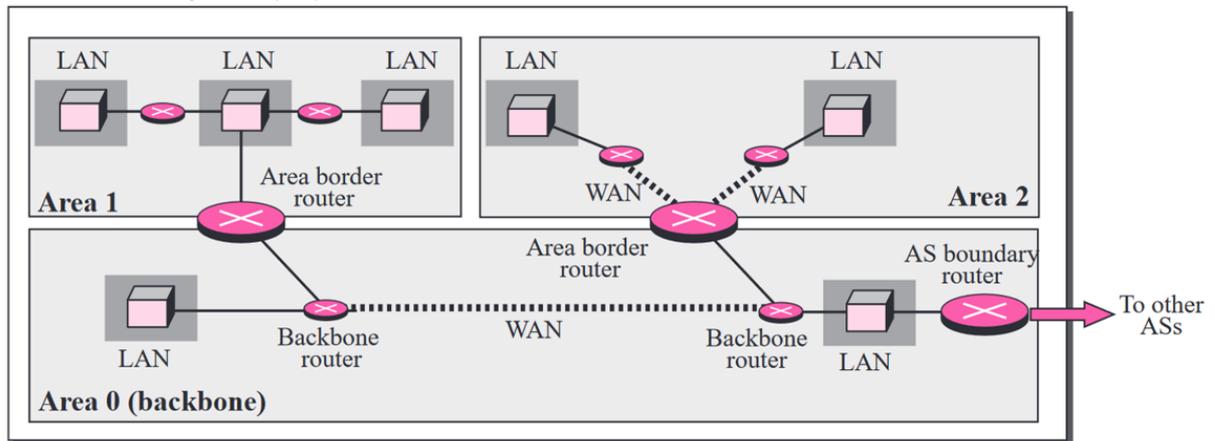
Autonomous System (AS)

FIGURE 2.14 – Division de chaque AS en zones (Areas)

ainsi une structure hiérarchique qui facilite la gestion des informations et la mise à jour de la topologie du réseau.

Il existe 4 types de liens dans OSPF :

1. lien point-à-point :

Connecte Deux Routeurs : Il établit une connexion directe entre deux routeurs sans aucun autre élément entre eux.



FIGURE 2.15 – lien point-à-point

2. lien transitoire :

Il s'agit d'un réseau auquel plusieurs routeurs sont connectés, permettant une communication entre eux.

Les données peuvent entrer par n'importe lequel des routeurs connectés au réseau et sortir par n'importe quel autre routeur. Cette interchangeabilité offre une flexibilité dans la transmission des données.

Exemples : Toutes les LAN (réseaux locaux) et WAN (réseaux étendus) avec deux routeurs ou plus sont de ce type.

3. lien stub (extrémité) :

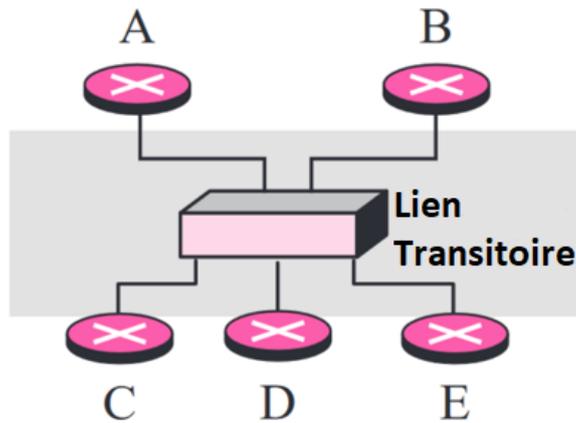


FIGURE 2.16 – lien transitoire

Il s'agit d'un réseau qui est connecté à un seul routeur.

Les paquets entrent dans le réseau par ce routeur et en sortent également par le même routeur. Ce type de lien crée une voie de communication simple et unidirectionnelle.

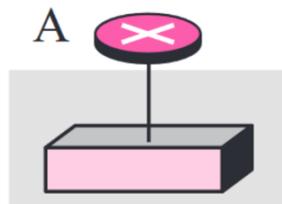


FIGURE 2.17 – lien stub (extrémité)

4. lien virtuel :

Lorsqu'un lien direct entre deux routeurs est rompu, un lien virtuel est créé. Ce lien virtuel utilise un chemin alternatif plus long qui passe par plusieurs routeurs pour maintenir la connectivité.

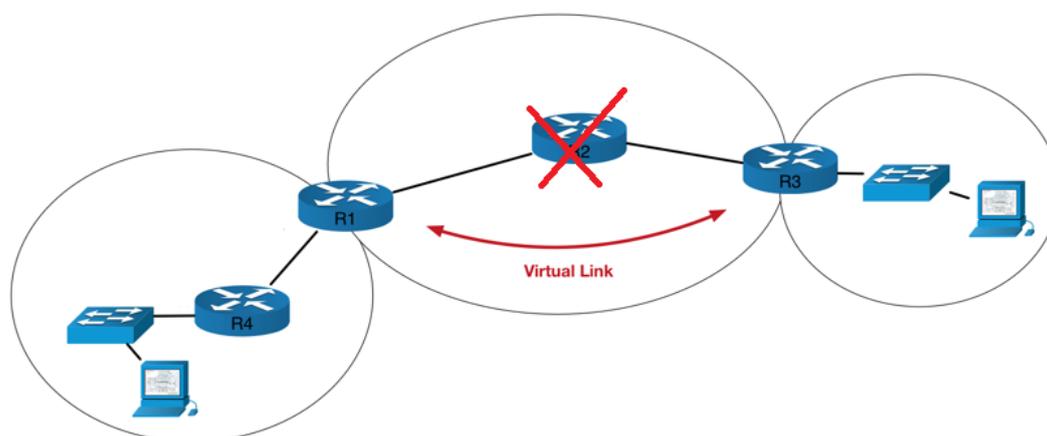


FIGURE 2.18 – lien virtuel