

L'interconnexion de réseaux - Plan

- Les réseaux
- Le protocole IP
- Les protocoles de l'Internet
- **L'adaptation des chemins**
 - fonctionnement des routeurs
 - protocoles de routage



Couche réseau

La couche réseau

- **routage** : terme général, utilisé pour 2 fonctions
 - **acheminement** (*forwarding*)
 - trouver un chemin au paquet vers sa destination par consultation de tables
 - fonction simple, se déroulant localement à un noeud
 - **adaptation des chemins** (*routing*)
 - construire et mettre à jour les tables servant à l'acheminement
 - fonction complexe, mettant en œuvre des algorithmes distribués qui s'appuient sur des protocoles de routage



Acheminement par datagramme

- Principe

- étiquette = adresse du destinataire
- les paquets sont acheminés individuellement

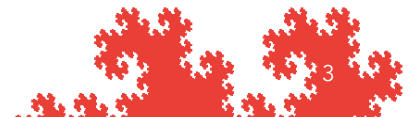
- Avantages et inconvénients

l'expéditeur peut émettre
sans accord préalable
explicite du réseau

fiabilité non garantie

- déséquence
- pertes de paquets

overhead (étiquette + longue)



Acheminement des datagrammes IP

- Les routeurs de l'Internet

- Chaque routeur possède sa table de routage propre avec :
 - des adresses IP d'autres réseaux (@IP + masque)
 - des adresses IP d'hôtes spécifiques

Pas utile de stocker toutes les adresses des hôtes de chaque sous-réseau : un sous-réseau est vu par le monde extérieur comme un réseau entier.



Acheminement des datagrammes IP

- Routage en fonction du réseau destination
réduction de la taille des tables de routage

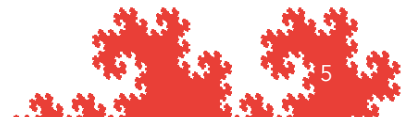
Utilisée pour le routage

Référence unique du
réseau dans le monde

Référence unique de la
machine dans le réseau

Adresse IP

- Table de routage:
 - ensemble de paires (R, P)
 - R: adresse IP du réseau destination
 - P: adresse IP de la passerelle suivante sur le chemin qui mène à R



Acheminement des datagrammes IP

- le module IP examine l'adresse de destination du datagramme et détermine si la destination est sur un même réseau physique
 - si oui : **routage direct**
 - la destination étant sur un même réseau physique, le datagramme est transmis directement
 - ↳ l'adresse physique suivante est celle de la destination
 - si non : **routage indirect**
 - la destination étant sur un réseau physique différent, le datagramme est transmis au routeur suivant
 - ↳ l'adresse physique suivante est celle du routeur suivant

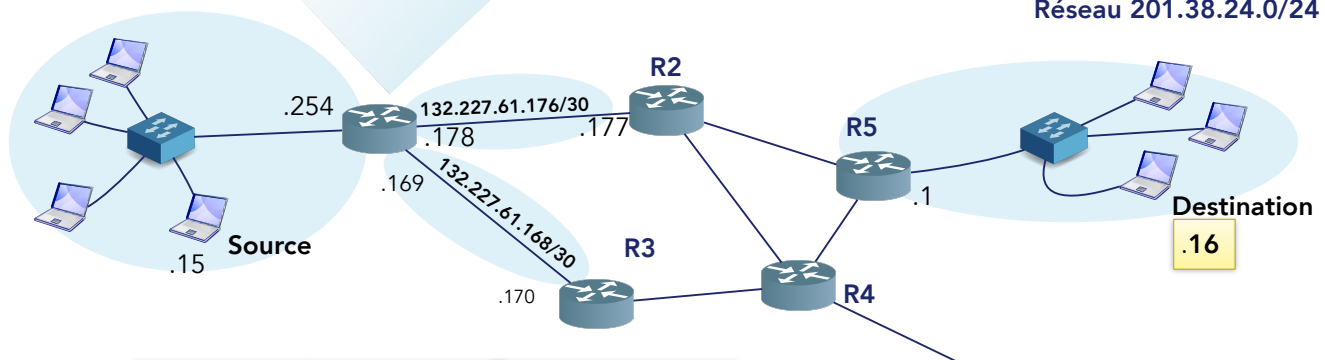


Acheminement des datagrammes IP

destination		prochain	métrique
@ réseau	132.227.60.0	*	0
@ réseau	201.38.24.0	R2	2
@ machine	201.38.24.25	R3	2
@ réseau	132.227.192.128	R3	1
@ réseau	132.227.61.176	*	0
@ réseau	132.227.61.168	*	0
@ réseau par défaut	défaut	R3	1

132.227.60.0/24

Réseau 201.38.24.0/24



201.38.24.16 132.227.60.15

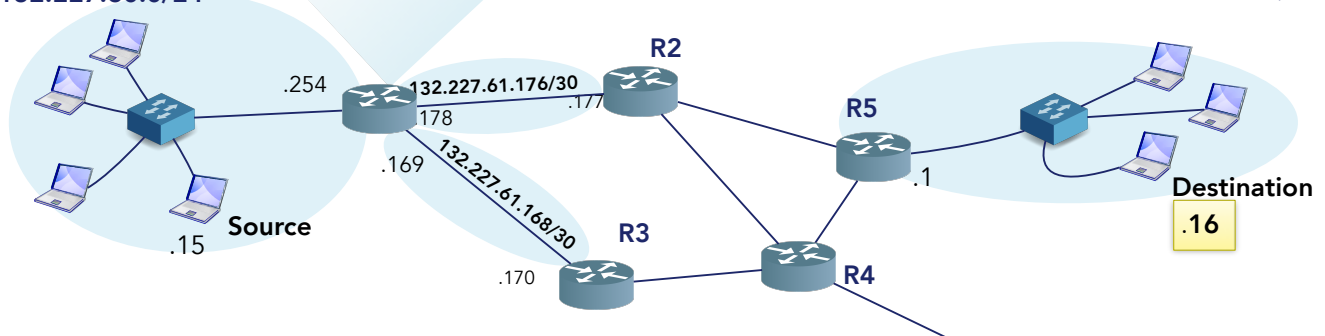
Acheminement des datagrammes IP

@ réseau
@ réseau
@ machine
@ réseau
@ réseau
@ réseau
@ par défaut

destination	masque	prochain routeur	métrique
132.227.60.0	255.255.255.0	*	0
201.38.24.0	255.255.255.0	132.227.61.177	2
201.38.24.25	255.255.255.255	132.227.61.170	2
132.227.192.128	255.255.255.128	132.227.61.170	1
132.227.61.176	255.255.255.252	*	0
132.227.61.168	255.255.255.252	*	0
0.0.0.0	0.0.0.0	132.227.61.170	1

132.227.60.0/24

Réseau 201.38.24.0/24



201.38.24.16 132.227.60.15

Acheminement des datagrammes IP

- Table de routage sur une machine UNIX

```
hera:[67] ifconfig -au
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
inet 132.227.61.135 netmask fffffff00 broadcast 132.227.61.255
le1: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING>
inet 132.227.60.32 netmask fffffff00 broadcast 132.227.60.255
```

hera:[68] netstat -rn

Routing tables

Destination	Gateway	•••//•••	Interface
132.227.64.0	132.227.60.23		le1
default	132.227.60.3		le1
•••			
132.227.84.0	132.227.60.18		le1
132.227.60.0	132.227.60.32		le1
134.157.0.0	132.227.60.3		le1
132.227.61.0	132.227.61.135		le0
132.227.62.0	132.227.60.18		le1

hera:[69] netstat -r

Routing tables

Destination	Gateway	•••//•••	Interface
ocean-net	masi		le1
default	cisco-upmc		le1
•••			
litp4-net	kawarismi		le1
ibp-net	hera-gw		le1
upmc-net	cisco-upmc		le1
olympie-net	hera		le0
rxfr-net	kawarismi		le1

Acheminement des datagrammes IP

Pour chaque datagramme IP qui **traverse le routeur**, IP :

- Détermine si ce sont des **données utilisateur** ou **de contrôle** destinées au routeur (ICMP)
- vérifie le **checksum** (si faux → destruction du datagramme)
- décrémente le **TTL** (si nul → destruction du datagramme)
- Détermine le **prochain saut** (parcours la table de routage)
- **fragmente** le datagramme *si nécessaire*
- **Met à jour l'en-tête IP** (TTL, Checksum,...)
- **transmet** le(s) datagramme(s) au protocole d'accès de l'interface réseau de sortie (carte *eth0*, *eth1* ? - @MAC du destinataire)

Acheminement des datagrammes IP

A réception dans l'hôte destinataire IP :

- vérifie le checksum
- s'il y a eu **fragmentation**, stocke les fragments puis **réassemble**
- délivre au niveau supérieur les données

La fonction d'adaptation des chemins

Lorsqu'un datagramme arrive sur un routeur, le routeur consulte sa table de routage pour décider du prochain nœud pour ce paquet....

Comment la table de routage est-elle construite ?

Les protocoles de routage construisent et maintiennent des tables de routage

L'interconnexion de réseaux - Plan

- Les réseaux
- Le protocole IP
- Les protocoles de l'Internet
- L'adaptation des chemins
 - fonctionnement des routeurs
 - protocoles de routage
 - le routage
 - protocoles à états des liens
 - protocole à vecteurs de distance

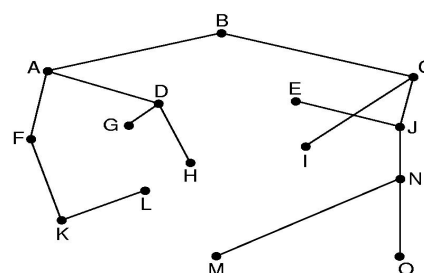
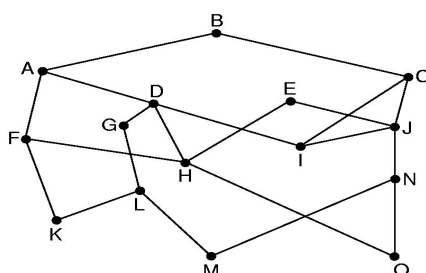
Couche réseau

Le routage

- Le routage est un problème de la **théorie des graphes**

trouver le chemin de coût minimum entre deux nœuds, sachant que le coût d'un chemin est la somme des coûts des liens qui le composent

- L'ensemble des routes optimales de toutes les sources vers une destination donnée forment un **arbre** dont la racine est la destination. (Plusieurs arbres peuvent exister pour une même destination).
- But des **algorithmes de routage** : construire l'arbre.



Le routage

- Le routage doit être :
 - correct
 - robuste (les réseaux sont en place pour longtemps)
 - stable (pas de boucle)
 - optimal et juste (contradiction ? Optimiser les performances du délai de transit des paquets et l'utilisation du réseau peut être fait en réduisant le nombre de sauts faits par les paquets)
 - Il peuvent être statique ou dynamique



Le routage

routage statique

- manuel
- peut inclure des critères autres que le plus court chemin
- ne passe pas à l'échelle

routage dynamique

- réactualisé dynamiquement en fonction des modifications de topologie
- choix des routes sur un critère de plus court chemin
- lourd et complexe (protocoles souvent distribués diffusant périodiquement de l'information)



Le routage

Plusieurs classes de routage dynamique

- **Routage isolé** : sans aucun échange d'information entre les nœuds (pas de table de routage). Ex. **routage par inondation**
- **Routage centralisé** : Un centre de contrôle reçoit périodiquement des informations sur la topologie, calcule les tables de routage et les leur expédie.
- **Routage distribué** : Echange périodiquement d'informations avec ses voisins et recalcule par chaque routeur de sa table de routage. Deux types de protocoles de routage distribué sont largement utilisés :
 - Le **routage à vecteurs de distance** (utilisé par exemple avec le protocole RIP) ;
 - Le **routage à état de liens** (utilisé par exemple avec le protocole OSPF).

Le routage dans l'Internet

Le **protocole de routage** communique des informations sur la **topologie globale du réseau** à chaque routeur afin que chacun puisse prendre une décision *locale* de routage...

...mais l'information globale est :

- difficile à collecter
- sujette à des modifications fréquentes
- volumineuse

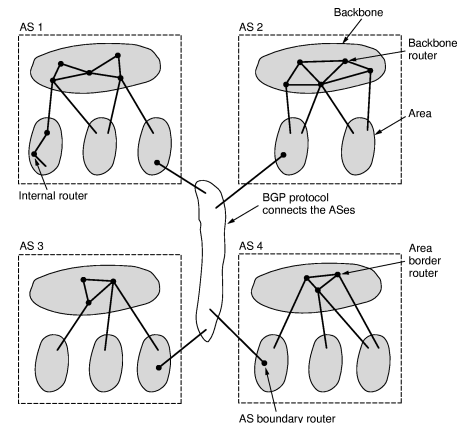
Le protocole de routage doit :

- minimiser le nombre de messages de contrôle échangés
- minimiser l'espace des tables de routage
- créer des chemins robustes :
éviter les trous noirs, les boucles et les oscillations
- définir le chemin « optimal »

Le routage dans l'Internet

• A l'intérieur d'un système autonome

- protocole à **vecteurs de distance** - algorithme **Bellman Ford** (Protocole Routing Information Protocol - **RIP**)
- protocoles à **états des liens** - algorithme **Dijkstra** (Protocole Open shortest path first - **OSPF**)



• Entre systèmes autonomes

pour déplacer les paquets le plus vite possible (Protocole **Border Gateway Protocol** - **BGP**)

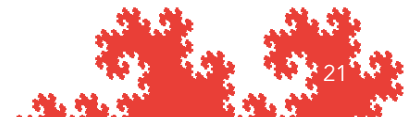
L'interconnexion de réseaux - Plan

- Les réseaux
- Le protocole IP
- Les protocoles de l'Internet
- L'adaptation des chemins
 - fonctionnement des routeurs
 - protocoles de routage
 - le routage
 - **protocoles à états des liens**
 - protocole à vecteurs de distance

Le routage à état de liens

– Principe :

- Chaque routeur connaît le coût pour atteindre ses voisins directs
- Chacun **diffuse périodiquement** cette information (état de liens) pour que chaque routeur ait connaissance de la topologie complète
- Chaque routeur exécute alors l'**algorithme de Dijkstra** pour déterminer sa table de routage



21

Le routage à état de liens

```

DIJKSTRA (G, s)
pour chaque sommet  $v \in S$  faire
     $d[v] = \infty$ 
     $\text{pred}[v] = \text{NUL}$ 
fin pour

 $d[s] = 0$ 
 $E = \emptyset$ 
 $R = S$ 
tant que  $R \neq \emptyset$  faire
     $u =$  sommet avec l'estimation de pondération
    minimum de  $R$ 
     $E = E \cup \{u\}$ 
    pour chaque sommet  $v \in \text{voisin}[u]$  faire
        si  $d[v] > d[u] + \text{coût}(u,v)$  alors
             $d[v] = d[u] + \text{coût}(u,v)$ 
             $\text{pred}[v] = u$ 
     $R = R - \{u\}$ 
fin tant que
  
```

Plus court chemin entre s et les autres sommets d'un graphe $G = (S, A)$ valué par une fonction de pondération c

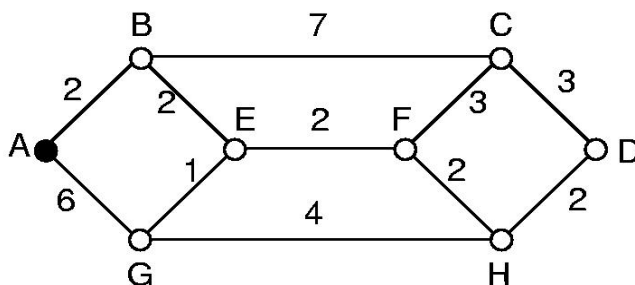
- E : le sous-ensemble des sommets déjà évalués
- R : le sous-ensemble des sommets restants
- $d[v]$: l'estimation du coût minimum,
- $\text{pred}[v]$: le prédécesseur de v sur le chemin estimé.



22

Le routage à état de liens

- La **représentation** du sous-réseau se fait à l'aide d'un **graphe** où les **routeurs** sont représentés par des **nœuds** et les **liens** par des **arcs**.
- La **valeur des arcs** donne :
 - le plus court chemin : distance métrique
 - le chemin le plus rapide : délai d'attente dans les routeurs.
 - qualité du lien

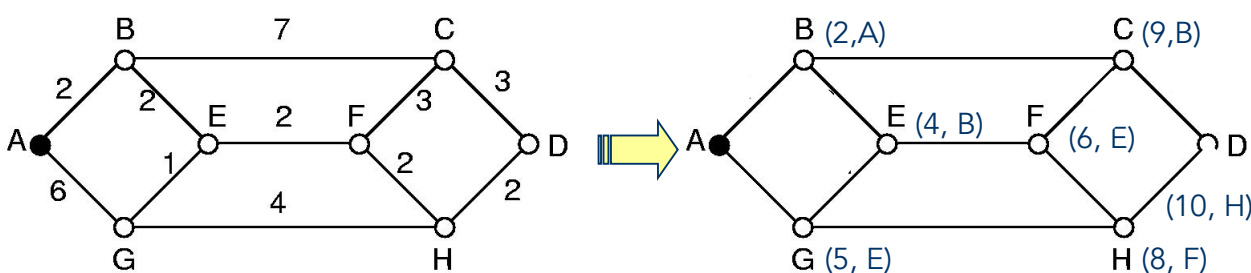


Le routage à état de liens

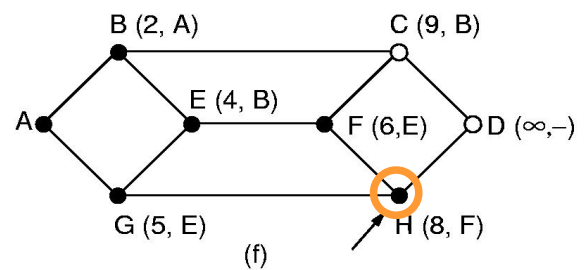
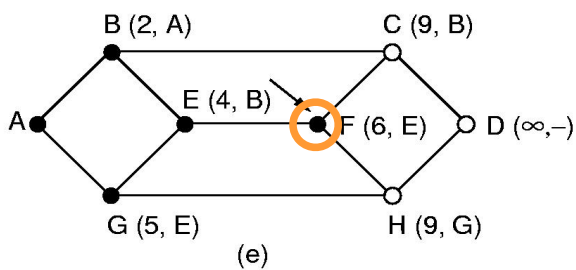
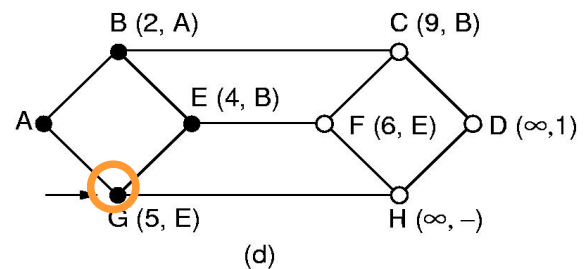
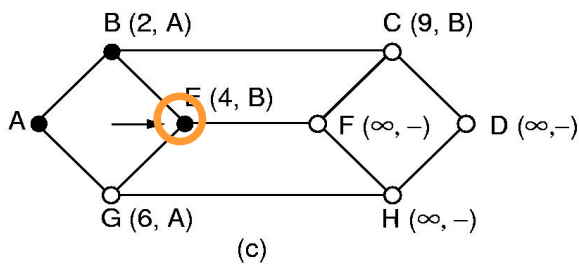
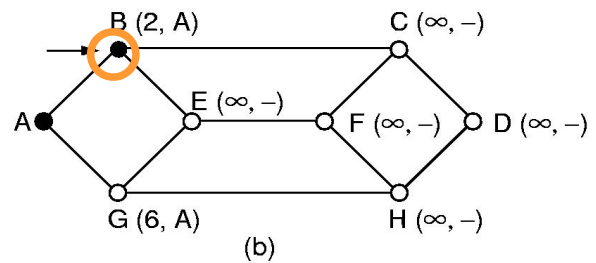
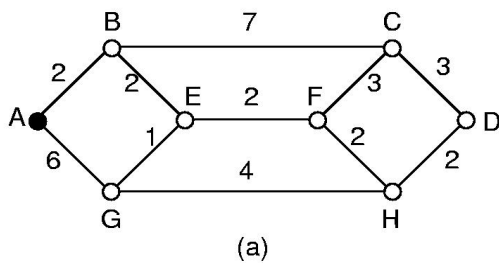
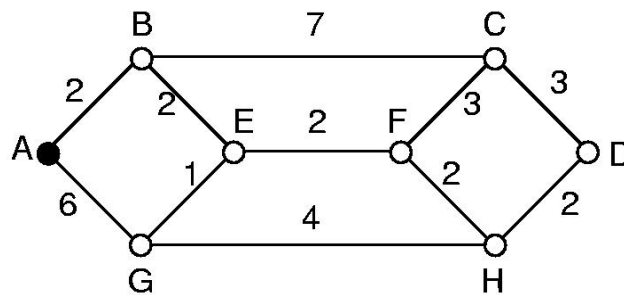
Routage du chemin le plus court

- Une implémentation a été proposée par **Dijkstra**:
 - A chaque **nœud** est associée une **étiquette** contenant sa distance la plus courte jusqu'au nœud source (choix du meilleur chemin).
 - Initialement le meilleur chemin n'est pas connu, l'algorithme le crée en mettant à jour progressivement la valeur des étiquettes des nœuds.

(La source est instaurée comme nœud permanent mais on peut également appliquer cet algorithme en fixant la destination).



A	B	C	D	E	F	G	H
	(2,A)	(∞ , -)	(∞ , -)	(∞ , -)	(∞ , -)	(6,A)	(∞ , -)



Le routage à état de liens

- **Le protocole OSPF (Open Shortest Path First)**
 - **Les routeurs OSPF doivent remplir les tâches suivantes :**
 - établir la liste des routeurs voisins
 - élire le routeur désigné (DR) dans sa zone et le routeur de secours (BDR). *DR = porte parole*
 - découvrir les routes
 - élire les routes à utiliser
 - maintenir la base topologique (LSBD)

Le routage à état de liens - OSPF

- Le protocole définit :
 - 5 messages différents :
 - ➔ **HELLO** : découverte du voisinage
 - ➔ **DBD - DataBase Description** : décrit la BD état des liens d'un routeur
 - ➔ **LSR - Link-State Request** : demande d'éléments des BD d'état des liens (sur réception d'un DBD indiquant de nouvelles entrées).
 - ➔ **LSU : Link-State Update** : transporte les LSA (link-state Advertisements) aux voisins.
 - ➔ **LSAck** : Link-State Ack : Ack des LSA voisins
 - utilise les transmissions unicast ou multicast pour limiter les diffusions (vers tous les routeurs OSPF, vers les routeurs désignés)

Le routage à état de liens - OSPF

- OSPF définit 5 messages encapsulés dans un datagramme IP (protocole : 89)

version	type	longueur
Routeur ID		
Area ID		
Checksum		AuType
Authentification		
Authentification		

Type 1 Hello	Établit et maintient de la base d'adjacence
Type 2 Database Description packet (DBD)	Décrit le contenu des bases de données d'état de liens (<i>link-state database</i>) des routeurs OSPF.
Type 3 Link-state request (LSR)	Demande des éléments spécifiques des bases de données d'état de liens (<i>link-state database</i>) des routeurs OSPF.
Type 4 Link-state update (LSU)	Transporte les <i>link-state advertisements</i> , les LSA, aux routeurs voisins.
Type 5 Link-state ackt (LSAck)	Accusés de réception des LSA des voisins.

Le routage à état de liens - OSPF

OSPF, à l'état initial :

Deux routeurs sont dits **voisins**, s'ils ont au moins un lien commun

Découverte du voisinage

- les routeurs diffusent toutes les 10 sec. des messages « **hello** » à l'@ de destination multicast 224.0.0.5
-> tous les routeurs
- Sur réception d'un « **hello** » d'un de ses voisins :
 - s'il est déjà dans la liste de ses voisins (ok) : Reply

Le *dead-interval* est la durée après laquelle un lien est considéré "down" par OSPF.

Ce délai est fixé par défaut à 4 fois le délai des « hellos »

Le routage à état de liens - OSPF

• Format des messages HELLO d'OSPF

- ▶ Frame 22: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
- ▶ Ethernet II, Src: 00:95:c6:1e:bc:02 (00:95:c6:1e:bc:02), Dst: IPv4mcast_05 (01:00:5e:00:00:05)
- ▶ Internet Protocol Version 4, Src: 192.168.95.1, Dst: 224.0.0.5
- ▼ Open Shortest Path First
 - ▼ OSPF Header
 - Version: 2
 - Message Type: Hello Packet (1)
 - Packet Length: 48
 - Source OSPF Router: 1.1.1.1
 - Area ID: 0.0.0.0 (Backbone)
 - Checksum: 0xa643 [correct]
 - Auth Type: Null (0)
 - Auth Data (none): 0000000000000000
 - ▼ OSPF Hello Packet
 - Network Mask: 255.255.255.252
 - Hello Interval [sec]: 10
 - Options: 0x12 ((L) LLS Data block, (E) External Routing)
 - Router Priority: 1
 - Router Dead Interval [sec]: 40
 - Designated Router: 192.168.95.2
 - Backup Designated Router: 192.168.95.1
 - Active Neighbor: 2.2.2.2
 - ▶ OSPF LLS Data Block

Il contient notamment

- L'ID du routeur
- L'ID de l'area
- Les timers Hello et Dead-interval
- La liste de voisin

Le routage à état de liens - OSPF

OSPF échange des BDB

- Sur réception d'une BDB un routeur peut s'apercevoir qu'il lui manque des informations
- Demande de détails sur la topologie
 - Envoi d'un **LSR – Link State Request**
 - Le destinataire répond par un **LSU – Link State Update**.
 - Le LSU est acquitté par un **LSAck**

Après ces échanges les bases sont synchronisées.

Le routage à état de liens - OSPF

Election des routes avec OSPF

- Avec la BD topologique, le routeur peut créer la table de routage.
- L'algorithme du SPF (dijkstra) est appliqué pour identifier les « meilleurs » chemins.

Note : sur une BD topologique importante, le calcul consomme pas mal de ressources CPU (algorithme complexe !)



Le routage à état de liens - OSPF

OSPF maintient de la base topologique

- Un changement de l'état d'un lien peut-être détecté par un routeur grâce aux paquets **HELLO** périodiques
- un paquet **LSU** est alors diffusé

Toute modification de la topologie déclenche :

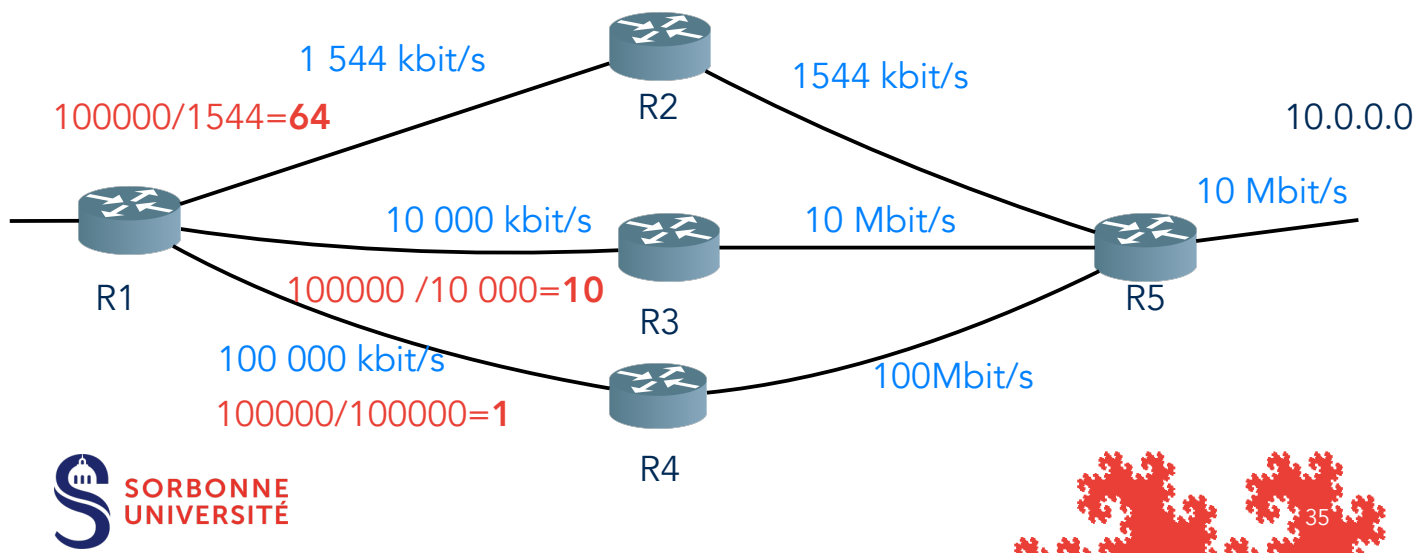
- l'exécution de l'algorithme du SPF
- la constitution d'une nouvelle table de routage



Le routage à état de liens - OSPF

• Métrique de coût d'OSPF

- bande passante de référence / bande passante du lien
- (ref. BP 100 000 kbit/s)



Le routage à état de liens - OSPF

Avantages OSPF :

- convergence rapide
- algo performant pour neutraliser les boucles
- les échanges des LS des routeurs voisins permettent de déterminer le plus court chemin

Inconvénients :

- le protocole est complexe (élection des DR, ...)
- le protocole - l'algorithme est gourmand

L'interconnexion de réseaux - Plan

- Les réseaux
- Le protocole IP
- Les protocoles de l'Internet
- L'adaptation des chemins
 - fonctionnement des routeurs
 - protocoles de routage
 - le routage
 - protocoles à états des liens
 - **protocole à vecteurs de distance**

Routage à vecteurs de distance

- Il s'agit de l'algorithme original de l'ARPANET également utilisé dans l'Internet **Routing Information Protocol** (RIP), dans les premières versions de DECnet et IPX, et amélioré pour les routeurs CISCO.
- Il définit le chemin que doit emprunter les paquets grâce à :
 - une **distance** - unité métrique (**nombre de sauts**)
 - une **direction** : (*next-hop* : @IP du prochain routeur, *interface de sortie* : interface vers laquelle envoyer les paquets)

Routage à vecteurs de distance

Principe des protocoles à vecteurs de distance

- Chaque routeur maintient une table de routage donnant la meilleure distance pour chaque destination.
- Les tables sont mises à jour quand les routeurs échangent **périodiquement** des informations avec les routeurs voisins (**vecteurs de distance** : liste de toutes les destinations connues et le coût associé)
- Chaque routeur reçoit les vecteurs de distance de ses voisins et exécute l'**algorithme Bellman-Ford** pour déduire le plus court chemin pour aller à chacun d'entre eux.

Routage à vecteurs de distance

Algorithme Bellman-Ford

$c(x,z)$: le coût pour aller de x à un voisin z

$d_v(y)$: le coût pour aller de z à y

$$\text{alors } d_x(y) = \min_z (c(x,z) + d_z(y))$$

Initialement chaque routeur ne connaît que $c(x,z)$ le coût vers ses voisins directs.

Ils s'échangent cette information et mettent ainsi à jour leur table de routage ajoutant ainsi des entrées pour des voisins y .

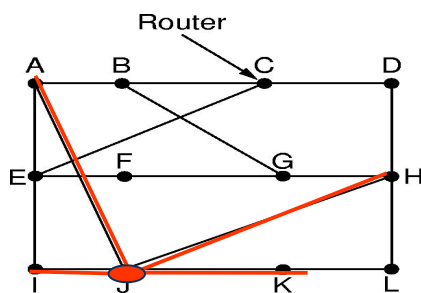
Routage à vecteurs de distance

Exemple

J reçoit les vecteurs de délais de ses voisins, et calcule un temps estimé pour chacun d'eux.

Supposons que les délais mesurés pour aller aux voisins soient

$JA=8$, $JI=10$, $JH=2$, $JK=6$



To	A	I	H	K	Line
A	0	24	20	21	8 A
B	12	36	31	28	
C	25	18	19	36	
D	40	27	8	24	
E	14	7	30	22	
F	23	20	19	40	
G	18	31	6	31	
H	17	20	0	19	
I	21	0	14	22	
J	9	11	7	10	
K	24	22	22	0	
L	29	33	9	9	



Routage à vecteurs de distance

R4 est mis en marche

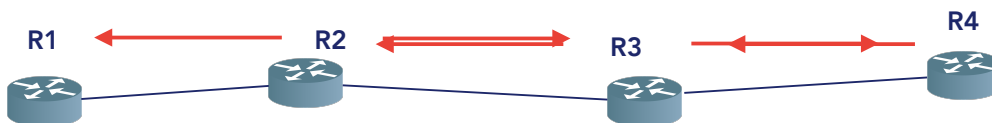


Table de R1		
@ dest.	next hop	coût
R2	R2	1
R3	R2	2
R4	R3	3

Table de R2		
@ dest.	Next hop	coût
R1	R1	1
R3	R3	1
R4	R3	2

Table de R3		
@ dest.	Next hop	coût
R1	R2	2
R2	R2	1
R4	R4	1

Table de R4		
@ dest.	Next hop	coût
R1	R3	3
R2	R3	2
R3	R3	1

les bonnes nouvelles voyagent vite...



Routage à vecteurs de distance

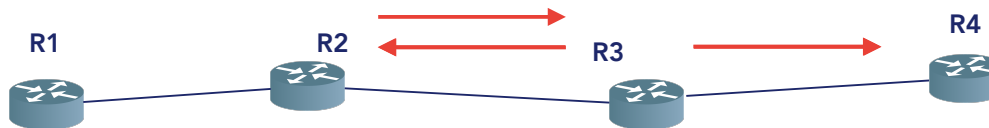


	Table de R1	
@ dest.	next hop	coût
R2	R2	1
R3	R2	2
R4	R3	3

	Table de R2	
@ dest.	Next hop	coût
R1	R3	infini
R3	R3	1
R4	R3	2

	Table de R3	
@ dest.	Next hop	coût
R1	R2	infini
R2	R2	1
R4	R4	1

	Table de R2	
@ dest.	Next hop	coût
R1	R3	infini
R2	R3	2
R3	R3	1

.... et les mauvaises nouvelles vont lentement
Problème: comptage vers l'infini

Routage à vecteurs de distance

Le protocole RIP (Routing Information Protocol)

- Envoi des vecteurs de distance
 - **Update** : délai des mises à jour, 30 s. par défaut
 - si une modification est détectée
- Route considérée invalide après un certain délai sans annonce. (180 sec. par défaut)

- Métrique : nombre de sauts
 Nombre maximum de sauts (15) et métrique infinie (16)

Routage à vecteurs de distance

Le protocole RIP (Routing Information Protocol)

- Limité à des réseaux de petite taille
- Amélioration de la convergence du protocole
 - **Route poisoning** : lorsqu'une route vers un réseau tombe, la route prend la valeur infinie.
 - **horizon partagé** : empêche à un routeur d'envoyer des informations (de métrique plus élevée) à travers l'interface de laquelle elle a appris l'information.

Routage

RIP

Avantages :

- convient aux petits réseaux.
- facile à comprendre et à configurer.
- supporté par presque tous les routeurs.

Inconvénients :

- peut créer un goulot d'étranglement (actualisations toutes les 30 secondes)
- sauts de RIP est limité à 15.
- convergence est lente
- métrique = nombre de sauts
- ne supporte pas les chemins multiples sur la même route

OSPF

- connaissance complète de la topologie du réseau
- pas limité dans le nombre de sauts
- complexité