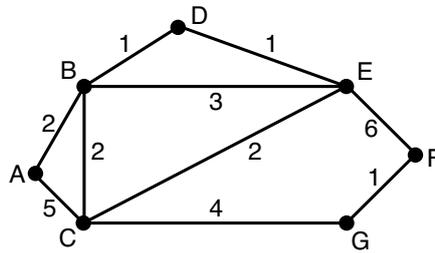
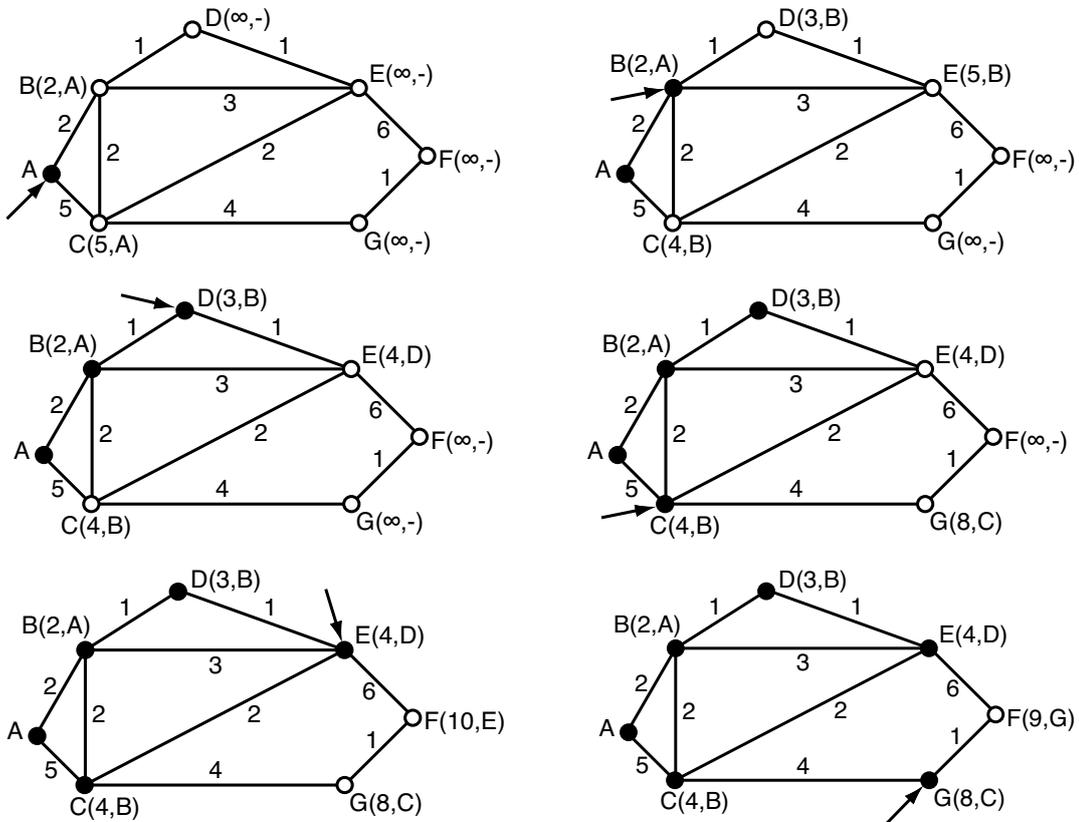


1 On considère le sous-réseau suivant :



1. Les étapes de l'algorithme du plus court chemin pour la destination A sont :



Les étiquettes des nœuds à la fin de l'algorithme correspondent aux entrées des tables relatives à la destination A :

table de B :

dest.	prochain saut	coût
A	A	2
...

table de C :

dest.	prochain saut	coût
A	B	4
...

table de D :

dest.	prochain saut	coût
A	B	3
...

table de E :

dest.	prochain saut	coût
A	D	4
...

table de F :

dest.	prochain saut	coût
A	G	9
...

table de G :

dest.	prochain saut	coût
A	C	8
...

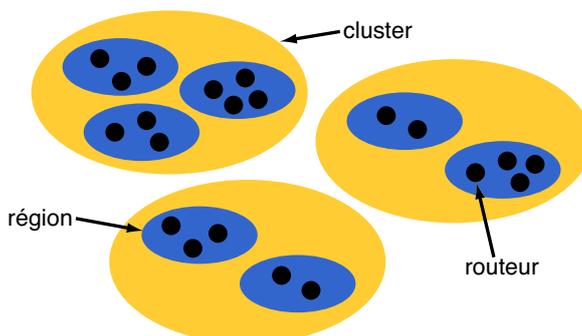
2. Les datagrammes de source F et destinés à A suivront la route FGCBA.

3. On utilise l'algorithme d'information par état de lien. Le tampon d'arrivée de F est

source	séqu.	âge	indicateurs d'envoi		indicateurs d'acquittement		données
			E	G	E	G	
A	21	60	1	0	0	1	
B	21	60	0	0	1	1	
C	21	59	1	0	0	1	
D	20	60	0	1	1	0	
E	21	58	0	1	1	0	
G	22	59	1	0	0	1	

- (a) Les champs *indicateurs* renseignent le routeur F des actions à envisager vis-à-vis de ses voisins. Par exemple, un indicateur d'envoi 1 pour le voisin E—symétriquement, un indicateur d'acquittement 1 pour le voisin G—pour l'entrée A signifie que F doit retransmettre à E le paquet d'état du lien vers A qu'il a reçu de G, auprès duquel il doit alors acquitter ce paquet.
- (b) Un paquet d'informations de G arrive à F via E.
- Si son numéro de séquence est 19 ou 20, le paquet est périmé donc ignoré.
 - Si son numéro de séquence est 22, il s'agit d'un doublon donc ignoré.
 - Si son numéro de séquence est 23, le paquet contient des nouvelles fraîches qui permettent de mettre à jour les données concernant la destination G. Les indicateurs pour l'entrée G deviennent 0, 1, 1 et 0.

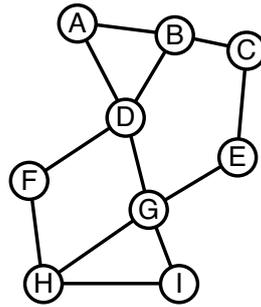
4 On se sert d'un algorithme de routage hiérarchique à trois niveaux dans un réseau de 8000 nœuds.



Soient r le nombre maximum de routeurs dans une région, h le nombre maximum de régions dans un cluster et c le nombre de clusters. On demande de minimiser la taille du tampon réservé à la table de routage de chaque nœud : cette taille est proportionnelle à $t = r - 1 + h - 1 + c - 1$. En effet, il y a au plus $r - 1$ façons d'atteindre un routeur dans la même région, au plus $h - 1$ façons d'atteindre une région dans le même cluster et $c - 1$ façons d'atteindre un cluster. On a la condition $rhc \geq 8000$. On obtient $t = r + h + c - 3$, or au minimum c vaut $\frac{8000}{rh}$ d'où $t = r + h + \frac{8000}{rh} - 3$.

On a $\frac{\partial t}{\partial r} = 1 - \frac{8000}{r^2h}$ et $\frac{\partial t}{\partial h} = 1 - \frac{8000}{rh^2}$. Les conditions $\frac{\partial t}{\partial r} = 0$ et $\frac{\partial t}{\partial h} = 0$ impliquent alors $r^2h = 8000$ et $rh^2 = 8000$ respectivement. Finalement, on arrive à la conclusion $r = h = \sqrt[3]{8000} = 20$ et $c = 20$. Le minimum de t est ainsi atteint pour $r = h = c = 20$. On obtient $t = 57$ (ou $t = 58$ si l'on inclut dans la table l'adresse propre du nœud).

6 On considère le réseau ad hoc suivant.



1. Avec l'algorithme AODV (Ad hoc On-demand Distance Vector), les routes sont construites à la demande avec des cycles RREQ/RREP. Le format d'un paquet ROUTE REQUEST est :

Adresse source	Identifiant de requête	Adresse de destination	Numéro de séq. de source	Numéro de séq. de destination	Nombre de sauts
----------------	------------------------	------------------------	--------------------------	-------------------------------	-----------------

Le format d'un paquet ROUTE REPLY est :

Adresse source	Adresse de destination	Numéro de séq. de destination	Nombre de sauts
----------------	------------------------	-------------------------------	-----------------

Chaque nœud gère deux compteurs (un compteur de requêtes et un compteur de séquences) et maintient trois tables (une table de routage, une table d'historique et une table de routes inverses) Une source cherchant une route vers une destination diffuse un RREQ.

Quand un nœud reçoit un paquet RREQ :

- il met à jour sa table d'historique (un RREQ dupliqué est ignoré)
- si le *numéro de séquence de destination* de sa table de routage est inférieur à celui du RREQ, il rediffuse le RREQ (après avoir incrémenté *nombre de sauts*) et met à jour sa table de routes inverses
- sinon il envoie en unicast un paquet RREP vers la source

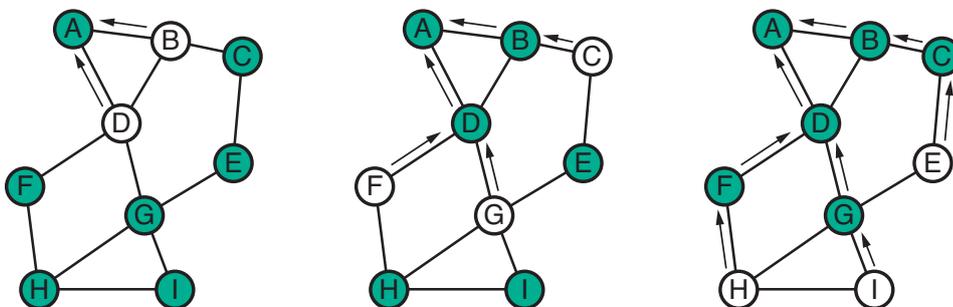
Quand un nœud reçoit un paquet RREP :

- si le *numéro de séquence de destination* de sa table de routage est inférieur à celui du RREQ ou égal à celui du RREQ mais la route plus courte, il met à jour sa table de routage

Quand un nœud devient inaccessible (ce qui est détecté par la non réponse à un paquet HELLO) :

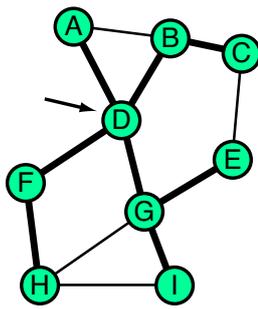
- ses nœuds voisins répercutent l'information aux voisins concernés

2. Voici le comportement de cet algorithme dans le cas où un processus du nœud A cherche à envoyer un paquet au nœud I.



Les flèches représentent les routes inverses enregistrées tout au long du cheminement des RREQ.

3. Voici une possible table de routage pour le nœud D :

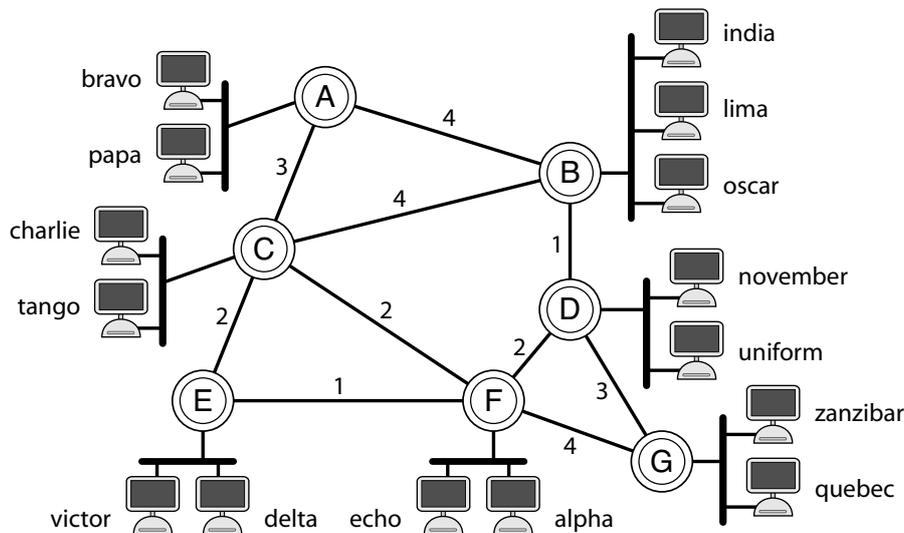


destination	prochain saut	distance	voisins actifs	numéro séq. destination
A	A	1	F, G	23
B	B	1	F, G	43
C	B	2	F	111
E	G	2		4
F	F	1	A, B	62
G	G	1	A, B	61
H	F	2	A, B	97
I	G	2	A, B	199

Si D apprenait que G ne fait plus partie du réseau, il supprimerait toutes les entrées ayant G comme prochain saut (en l'occurrence les destinations E, G et I) et informerait les voisins actifs vis-à-vis de ces entrées (en l'occurrence les voisins A et B). Les entrées supprimées resteraient vides jusqu'à ce qu'un processus de D en ait besoin et que D reçoive des RREP à sa RREQ ou jusqu'à ce que D obtienne des informations grâce à des RREQ ou des RREP en transit.

4. Supposons que le nœud B vienne juste de redémarrer et que sa table de routage soit vide. Il a subitement besoin d'une route vers H. Il envoie des paquets broadcast avec des valeurs de durée de vie de 1, 2, 3, etc. Le nœud H est à 3 sauts du nœud B, ce dernier met donc 3 cycles de diffusion pour découvrir la route.

7 On considère le réseau ci-dessous. Les hôtes dont le nom termine par *a* forme un groupe multicast.



1. Le plus court chemin unicast de *papa* vers *delta* est ACE (coût 5) ; celui vers *alpha* est ACF (coût 5) ; celui vers *india* et *lima* est AB (coût 4).
2. Un arbre multicast enraciné en *papa* peut être obtenu en superposant les chemins unicast précédents : on obtient l'arbre A(C(E)(F))(B) de coût 11. Cet arbre multicast cependant n'est pas de coût minimal : l'arbre linéaire A(B(D(F(E)))) se trouve être de coût 8, en l'occurrence minimal.
3. L'arrivée de *tango* dans le groupe fait que l'arbre A(B(D(F(E)))) n'est plus valable. Il faut utiliser l'arbre A(B(D(F(E)(C)))) ou l'arbre A(B(D(F(E(C)))) (chacun de coût 10). L'arrivée de *zanzibar* modifie l'arbre précédent par le greffage de la branche DG (coût 13).
4. Le départ d'*india* ne provoque aucun changement, mais le départ de *lima* du groupe entraîne la modification de l'arbre en A(C(E(F(G)))) (coût 10).