Internet Nouvelle Génération

Module de M1 de la mention Informatique
Spécialité Réseaux, 2e semestre 2007 - 2008
Université Pierre et Marie Curie

Support de TD

Révisions
Sécurité
Multicast
Qualité de service
Routage BGP
IPv6

Bénédicte LE GRAND
Prométhée SPATHIS

Spécialité RES

U.E. MI020 ING

TD de révision

1. Couches protocolaires (pré-requis pour la sécurité et le multicast)

Question 1.1

Quels sont les avantages et les inconvénients de la structuration en couches ?

Question 1.2

Lister 5 tâches qu'une couche peut effectuer. Est-il possible que l'une (ou plusieurs) de ces tâches soit effectuée(s) par plusieurs couches ?

Question 1.3

Quelles sont les 5 couches de la pile protocolaire de l'Internet ? Quels sont les rôles principaux de chaque couche ?

Question 1.4

Quelles sont les couches de la pile TPC/IP qui sont traitées par les routeurs ?

Question 1.5

Si tous les liens de l'Internet assuraient une remise de paquets fiable, le service de remise de paquets fiable serait-il complètement redondant ? Pourquoi ou pourquoi pas ?

Question 1.6

Quels types de services la couche liaison peut-elle offrir à la couche réseau ? Parmi ces services, lesquels ont des « équivalents » au niveau IP ? Au niveau TCP ?

2. Temps de transmission, de propagation, délai de bout en bout (prérequis pour le contrôle de trafic)

Question 2.1

Soit un paquet envoyé d'un émetteur vers un récepteur sur une route fixe. Lister les éléments constituants le délai de bout-en-bout pour ce paquet. Lesquels sont constants, lesquels sont variables ?

Question 2.2

On considère une autoroute avec des péages tous les 100 km. On peut voir les tronçons d'autoroute entre les péages comme des liens et les péages comme des routeurs. On suppose que les voitures roulent (c'est-à-dire se propagent) à la vitesse de 100 km/h (accélération supposée instantanée).On suppose qu'il y a une file de 10 voitures qui se suivent dans un ordre donné. On peut voir chaque voiture comme un bit et la file comme un paquet. On suppose aussi que chaque péage traite (i.e. transmet) une voiture à la vitesse d'une voiture toutes les 12s, et que les voitures de la file sont les seules sur l'autoroute. Finalement, on suppose que lorsque la première voiture de la file arrive au péage, elle attend que les 9 autres voitures soient arrivées (ainsi la file tout entière doit être « stockée » au niveau du péage avant qu'elle puisse être « forwardée »).

a) On suppose que la file de voiture parcourt 200 km, en commençant devant un péage, en passant à travers un deuxième et en s'arrêtant jsute avant un 3^{e}. Quel est le délai de bout en bout ?

b) Même question que dans a), en supposant cette fois qu'il y a 7 voitures dans la file au lieu de 10.

Question 2.3

Le délai tend-il vers l'infini quand la charge excède la capacité du réseau ?

3. Rappels couche transport (Pré-requis fonctionnalités avancées de TCP)

Question 3.1

Quels sont les différents types de services offerts par Internet ?

Question 3.2

Décrivez brièvement le mécanisme permettant de fournir un transport fiable.

Question 3.3

Expliquez pourquoi un concepteur d'application pourrait préférer faire tourner son application sur UDP plutôt que sur TCP.

Question 3.4

Est-il possible qu'une application bénéficie d'un transfert de données fiable même si elle tourne au-dessus d'UDP ?

Question 3.5

Supposons qu'un client A initie une session FTP avec u serveur S. A peu près au même moment, un client B initie également une session FTP avec le serveur S. Donnez des numéros de port source et destination pour :
(a) les segments envoyés par A à S
(b) les segments envoyés par B à S
(c) les segments envoyés par S à A
(d) les segments envoyés par S à B.
(e) Si A et B sont des hôtes différents, est-il possible que les numéros de port dans les segments de A à S soient les mêmes que ceux des segments de B vers S ?

Question 3.6

Considérons le transfert d'un énorme fichier de L octets d'un hôte A vers un hôte B. Supposons que la MSS soit de 1460 octets.

a) Quelles est la longueur maximale L telle que les numéros de séquence TCP ne soient pas épuisés ? (Rappel : le champ de numérotation TCP contient 4 octets)

b) Pour la valeur de L trouvée dans a), quel est le temps de transmission du fichier ? On suppose qu'un total de 66 octets de transport, réseau et liaison sont ajoutés à chaque segment avant d'envoyer le paquet résultant sur un lien à 10 Mbps. Ignorez le contrôle de flux et de congestion (A peut donc envoyer les segments les uns après les autres de manière continue).

4. Contrôle de congestion / contrôle de flux – Notion de fenêtre (Pré-requis pour les fonctionnalités avancées de TCP et le contrôle de trafic)

Question 4.1

Le contrôle de flux et le contrôle de congestion ont-ils le même objectif ?

Question 4.2

Décrivez le système de crédit utilise par TCP pour le contrôle de flux

Question 4.3

Vrai ou faux ?
a) Un hôte A envoie à B un gros file sur une connexion TCP. Supposons que B n'a pas de données à envoyer à A. L'hôte B n'enverra pas d'acquittements à A car B ne peut pas transporter d'acquittement avec des données.

b) La taille de la fenêtre TCP RcvWindow ne change jamais pendant la durée de la connexion

c) Supposons que l'hôte A envoie à B un gros fichier sur une connexion TCP. Le nombre d'octets non acquittés envoyés par A ne peut pas dépasser la taille du buffer de réception.

d) Supposons que l'hôte A envoie à B un gros fichier sur une connexion TCP. Si le numéro de séquence de l'un des segments de cette connexion est m, le numéro de séquence du segment suivant sera nécessairement m+1.

e) Le segment TCP possède dans son en-tête un champ pour RcvWindow?

f) Supposons que le dernier SampleRTT d'une connexion TCP soit égal à 1 sec. Alors le Timeout pour la connexion sera nécessairement fixé à une valeur >= 1 sec.

g) Supposons que l'hôte A envoie à B un segment avec le numéro de séquence 38 et contenant 4 octets de données. Alors dans ce même segment la valeur de l'acquittement est nécessairement 42.

Question 4.4

Quelle est la différence entre le système de crédit TCP et le système de contrôle de flux par fenêtre glissante utilisé par d'autres protocoles (par ex HDLC) ?

Question 4.5

Supposons que 2 connexions TCP coexistent sur un même lien (goulot d'étranglement) de débit R bps. Les deux connexions ont un gros fichier à envoyer (dans la même direction sur le goulot d'étranglement). La transmission des deux fichiers commence au même moment. Quel débit d'émission TCP accordera-t-il à chacune des connexions?

Considérons le contrôle de congestion dans TCP. Lorsqu'un timer expire du côté de l'émetteur, est-il vrai que la valeur du seuil passe à la moitié de sa valeur précédente ?

Question 4.6

TCP attend d'avoir reçu trois ACK dupliqués pour effectuer un fast retransmit. Pourquoi les concepteurs de TCP n'ont-ils pas choisi d'effectuer un fast retransmit après la réception du premier ACK dupliqué ?

5. Adressage (pré-requis pour le multicast et pour IPv6)

Question 5.1

Quel est la taille de l'espace d'adressage LAN ? De l'espace d'adressage IPv4 ?

Question 5.2

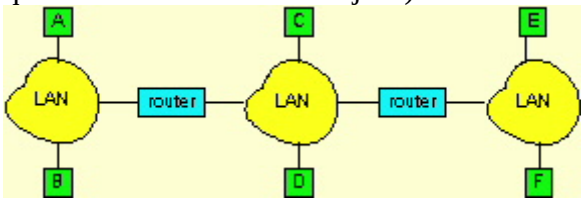
Quel est l'équivalent en binaire de l'adresse IP 223.1.3.27 ?

Question 5.3

Supposons que les noeuds A, B et C sont tous attachés au même LAN à diffusion. Si A envoie un milliers de trames à B avec l'adresse MAC destination égale à l'adresse MAC de B, est-ce que la couche MAC de C traitera ces trames ? Si oui, ces paquets seront-ils transmis à la couche 3 de C ? Mêmes questions si A envoie ses trames avec l'adresse broadcast de destination.

Question 5.4

Considérons 3 LANs interconnectés par 2 routeurs, comme le montre le schéma ci-dessous.
a) Redessiner le schéma en ajoutant les adaptateurs (niveau 2).
b) Affectez des adresses IP à chacune des interfaces. Pour le LAN 1, utilisez des adresses de la forme 111.111.111.xxx ; pour le LAN 2 utilisez des adresses de la forme 122.222.222.xxx ; et pour le LAN 3 des adresses de la forme 133.333.333.xxx .
c) Affectez des adresses LAN à chaque adaptateur de niveau 2.
d) Considérons l'envoi de datagrammes IP de l'hôte A vers l'hôte F. Supposons que les tables ARP sont à jour. Enumérez toutes les étapes.
e) Même question que d), en supposant maintenant que la table ARP de l'émetteur est vide (et que les autres tables sont à jour).



Question 5.5

Considérons un LAN auquel sont attachées 10 interfaces d'hôtes et 3 interfaces de routeurs. Supposons que les 3 LANs utilisent des adresses de classe C. Sur combien des 32 bits les adresses IP des 13 équipements seront-elles identiques ?

Question 5.6

Considérons un routeur avec 3 interfaces. Supposons que les 3 interfaces utilisent des adresses de classe C. Les adresses IP des 3 interfaces auront-elles nécessairement les mêmes 8 premiers bits ?

6. Routage intra-domaine (Pré-requis Multicast et IPv6)

Question 6.1

Qu'est-ce qu'un algorithme de routage à coût minimum ?

Question 6.2

Quels sont les principes et les différences entre les algorithmes à vecteurs de distance et à état des liens ? A quelle catégorie appartiennent les algorithmes de Djikstra et celui de Bellman-Ford ?

Question 6.3

Décrire et comparer les annonces utilisées par RIP et par OSPF.

Question 6.4

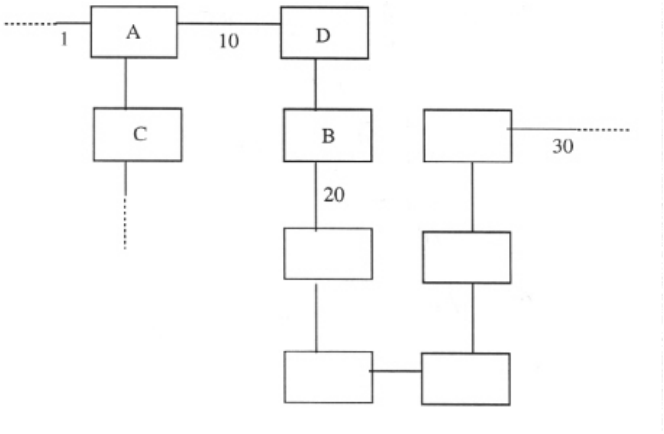
Dans quels cas peut-on préférer le routage par la source plutôt que le routage décidé par les routeurs ?

Question 6.5

Supposons qu'il y a 3 routeurs entre une source et sa destination. Par combien d'interfaces un segment IP envoyé de la source à la destination va-t-il passer ? Combien de tables de routage

Question 6.7

Considérons la portion de réseau ci-dessous :



La table de routage ci-dessous est la table de routage de D.

destination network	next router	number of hops to destination
1	A	2
20	B	2
30	B	7
10	--	1
....	....	....

On suppose que D reçoit de A l'annonce suivante :

destination network	next router	number of hops to destination
30	C	10
1	--	1
10	--	1
....	....	....

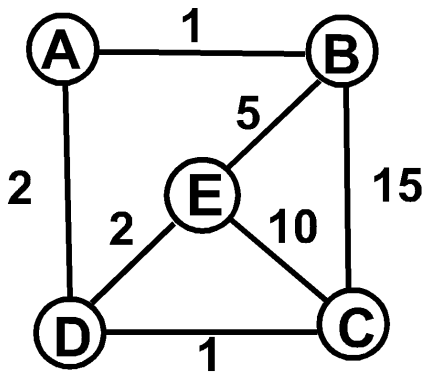
Est-ce que la table de D va changer ? Si oui, comment ?

Question 6.8

Considérons le réseau ci-dessous (le coût des liens est indiqué). Utilisez l'algorithme de Dijkstra pour calculer le plus court chemin de F vers tous les autres nœuds du réseau. Détaillez les étapes de l'algorithme.

Question 6.9

On considère le réseau ci-dessous et on suppose qu'à l'initialisation, chaque noeud connaît le coût vers chacun de ses voisins. Utilisez un algorithme à vecteur de distance et calculez la table de routage du noeud E.



LA SECURITE DES RESEAUX

1. INTRODUCTION

La cryptologie (étymologiquement la science du secret) ne peut être vraiment considérée comme une science que depuis peu de temps. Cette science englobe la cryptographie – l'écriture secrète – et la cryptanalyse – l'analyse de cette dernière.

La cryptologie peut être vue à la fois comme un art ancien et une science nouvelle : un art ancien car Jules César l'utilisait déjà ; une science nouvelle parce que ce n'est un thème de recherche scientifique académique que depuis les années 1970. Cette discipline est liée à beaucoup d'autres, par exemple la théorie des nombres, l'algèbre, la complexité, la théorie de l'information, ou encore les codes correcteurs d'erreurs.

2. VOCABULAIRE

Vocabulaire relatif aux différents termes relatifs à la cryptographie :

- chiffrement (en anglais *encryption*): transformation à l'aide d'une clé de chiffrement d'un message en clair (*plaintext*) en un message incompréhensible si on ne dispose pas d'une clé de déchiffrement ;
- chiffre : anciennement code secret, par extension l'algorithme utilisé pour le chiffrement ;
- cryptogramme (*cyphertext*) : message chiffré ;
- décrypter : retrouver le message clair correspondant à un message chiffré sans posséder la clé de déchiffrement (terme que ne possèdent pas les anglophones, qui eux « cassent » des codes secrets) ;
- cryptographie : étymologiquement « écriture secrète », devenue par extension l'étude de cet art (donc aujourd'hui la science visant à créer des cryptogrammes, c'est-à-dire à chiffrer) ;
- cryptanalyse : science analysant les cryptogrammes en vue de les décrypter ;
- cryptologie : science regroupant la cryptographie et la cryptanalyse.

3. CRYPTOGRAPHIE

La **cryptographie** est une des disciplines de la cryptologie s'attachant à protéger des messages (assurant confidentialité et/ou authenticité) en s'aidant souvent de *secrets* ou *clés*. Elle est utilisée depuis l'Antiquité, mais certaines de ses méthodes les plus importantes, comme la cryptographie asymétrique, n'ont que quelques dizaines d'années d'existence.

La cryptographie se scinde en deux parties nettement différenciées :

- la cryptographie à clef secrète, encore appelée symétrique ou bien classique ;
- la cryptographie à clef publique, dite également asymétrique ou moderne.

La première est la plus ancienne : on peut la faire remonter à l'Égypte de l'an 2000 av. J.-C. en passant par Jules César ; la seconde remonte à l'article de W. Diffie et M. Hellman, *New directions in cryptography* daté de 1976.

Toutes deux visent à assurer la confidentialité de l'information, mais la cryptographie à clef secrète nécessite au préalable la mise en commun entre les destinataires d'une certaine information : la clef (symétrique), nécessaire au chiffrement ainsi qu'au déchiffrement des messages. Dans le cadre de la cryptographie à clef publique, ce n'est plus nécessaire. En effet, les clefs sont alors différentes, ne peuvent se déduire l'une de l'autre, et servent à faire des opérations opposées, d'où l'asymétrie entre les opérations de chiffrement et de déchiffrement.

Bien que beaucoup plus récente et malgré d'énormes avantages — signature numérique, échange de clefs, … — la cryptographie à clef publique ne remplace pas totalement celle à clef secrète, qui pour des raisons de vitesse de chiffrement et parfois de simplicité reste présente. À ce titre, signalons la date du dernier standard américain en la matière, l'AES : décembre 2001, ce qui prouve la vitalité encore actuelle de la cryptographie symétrique.

Dans le bestiaire des algorithmes de chiffrement, on peut citer :

- pour les systèmes symétriques, le DES, l'AES, Blowfish, IDEA, etc.
- pour les systèmes asymétriques, le RSA, DSA-DH, ElGamal, les courbes elliptiques, etc.

3.1.1. Quelles sont les différences entre la confidentialité, l'authentification et l'intégrité d'un message ? Sont-elles nécessairement liées ? Donnez des exemples.

3.1.2. Quelle est la différence fondamentale entre un système à clé symétrique et un système à clé publique ?

3.1.3. Supposons que N personnes souhaitent communiquer avec chacune des N-1 autres en utilisant un chiffrement à clé symétrique. Toute communication entre deux personnes i et j est visible de toutes les autres et personne d'autre ne doit pouvoir décoder leur communication. De combien de clés a-t-on besoin en tout ? Supposons maintenant que l'on utilise un chiffrement à clé publique. Combien de clés sont nécessaires dans ce cas ?

4. CRYPTOGRAPHIE A CLE SYMETRIQUE

La cryptographie symétrique, également dite à clé secrète (par opposition à la cryptographie asymétrique ou à clé publique), est la plus ancienne forme de chiffrement. On a des traces de son utilisation par les Égyptiens vers 2000 av. J.-C. Plus proche de nous, on peut citer le célèbre chiffre de Jules César.

L'un des concepts fondamentaux de la cryptographie symétrique est la clé. Il s'agit d'une information qui doit permettre le chiffrement et le déchiffrement d'un message et sur laquelle peut reposer toute la sécurité de la communication. L'algorithme doit quant à lui pouvoir être divulgué. C'est ce que l'on appelle désormais le principe de Kerckhoffs. Il faut ajouter que cette clé doit pouvoir prendre suffisamment de valeurs pour qu'une attaque exhaustive — essai systématique de toutes les clés — ne puisse être menée à bien car trop longue. On parle de sécurité calculatoire.

4.1. ROT13

Le ROT13 (une variante de la méthode César) est un algorithme très simple de chiffrement de texte. Comme son nom l'indique, il s'agit d'un décalage de 13 caractères de chaque lettre du texte à chiffrer. Le défaut de ce chiffrement est que s'il s'occupe des lettres, il ne s'occupe pas des chiffres, des symboles et de la ponctuation. C'est pourquoi on supprime du texte à chiffrer toute accentuation, et si on veut conserver un texte correctement chiffré, il est nécessaire d'écrire les nombres en toutes lettres. Enfin, un caractère étant invariablement remplacé par un autre, cet algorithme est aussi qualifié de substitution mono-alphabétique.

À l'aide de la définition de cet algorithme, on peut alors définir la correspondance entre les caractères en clair et chiffrés :

Caractère non-chiffré	A	B	C	D	E	F	G	H	I	J	K	L	M
Caractère chiffré	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

4.1.1. Donnez le chiffrement en ROT13 du message suivant : « Ce problème est simple ». Déchiffrez à présent le message suivant : « Cnf fv qebye dhr pn ».

4.1.2. L'algorithme ROT13 utilise-t-il une clef secrète ?

4.1.3. En vous plaçant dans le cadre d'une substitution mono-alphabétique, montrez qu'il suffit de savoir que les noms des correspondants Bob et Alice apparaissent dans le message chiffré pour réduire le nombre de substitutions possibles d'un facteur de l'ordre de10^9.Cette attaque est connu sous le nom de l'attaque par texte connu (*known plaintext*).

4.2. Chiffre de Vigenère

Le Chiffre de Vigenère est un système de chiffrement, élaboré par Blaise de Vigenère (1523-1596), diplomate français du XVIe siècle.

C'est un système de substitution polyalphabétique. Cela signifie qu'il permet de remplacer une lettre par une autre qui n'est pas toujours la même, contrairement au Chiffre de César ou à ROT13. Ces deux derniers procédés se contentent d'utiliser la même lettre de substitution. C'est donc un système relativement plus robuste que ces deux systèmes.

La clé se présente généralement sous la forme d'un mot ou d'une phrase. Pour pouvoir chiffrer un texte, à chaque caractère est utilisée une lettre de la clé pour effectuer la substitution. Évidemment, plus la clé est longue et variée et mieux le texte est chiffré. L'outil indispensable du chiffrement de Vigenère est : « La table de Vigenère ».

		Lettre en clair																									
		A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
Clé Utilisée	**A**	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
	G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
	J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
	K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
	L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
	M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
	N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M

Lettre Codée

	O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
	P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
	Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
	R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
	S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
	T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
	U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
	V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
	W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
	X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
	Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
	Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	

Tableau 1 *Table de Vigenère*

Pour chaque lettre en clair, on sélectionne la colonne correspondante et pour une lettre de la clé on sélectionne la ligne adéquate, puis au croisement de la ligne et de la colonne on trouve la lettre codée. La lettre de la clé est à prendre dans l'ordre dans laquelle elle se présente et on répète la clé en boucle autant de fois que nécessaire.

4.2.1. En utilisant la clef ‘MUSIQUE’, donnez le résultat du chiffrement du texte en clair ‘j’adore la ecouter la radio toute la journee’ par le chiffre de Vigenère.

Clef : MUSIQUE

Texte : j'adore ecouter la radio toute la journee

```
Texte en clair :   j'adore ecouter la radio toute la journee
Clé répétée    :   M USIQU EMUSIQU EM USIQU EMUSI QU EMUSIQU
                   ^ ^^^
                   | ||Colonne O, ligne I: on obtient la lettre W.
                   | |Colonne D, ligne S: on obtient la lettre V.
                   | Colonne A, ligne U: on obtient la lettre U.
                   Colonne J, ligne M: on obtient la lettre V.
```

4.2.2. Le chiffre de Vigenère est-il sensible à l’attaque par texte connu ? Pourquoi ?

5. CRYPTOGRAPHIE A CLEF PUBLIQUE

Pour résoudre en partie le problème de la gestion des clés, la cryptographie asymétrique, ou cryptographie à clé publique a été mise au point dans les années 1970. La cryptographie asymétrique est fondée sur l'existence de fonctions à sens unique — c'est-à-dire qu'il est simple d'appliquer cette fonction à un message, mais extrêmement difficile de retrouver ce message une fois chiffré.

Elle se base sur le principe de deux clés :

- une publique, permettant le chiffrement ;
- une privée, permettant le déchiffrement.

Comme son nom l'indique, la clé publique est mise à la disposition de quiconque désire chiffrer un message. Ce dernier ne pourra être déchiffré qu'avec la clé privé, qui doit être confidentielle.

Quelques algorithmes de cryptographie asymétrique très utilisés :

- RSA ;
- DSA ;
- Protocole d'échange de clés Diffie-Hellman ;
- et d'autres.

5.1. Rivest Shamir Adleman (RSA)

RSA est un algorithme asymétrique de cryptographie à clé publique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ron Rivest, Adi Shamir et Len Adleman, d'où le sigle RSA. RSA a été breveté par le MIT en 1983 aux États-Unis d'Amérique, mais le brevet a expiré le 21 septembre 2000.

Cet algorithme est fondé sur l'utilisation d'une paire de clés composée d'une clé publique et d'une clé privée pour chiffrer des données confidentielles. La clé publique correspond à une clé qui est accessible par n'importe quelle personne souhaitant chiffrer des informations, la clé privée est quant à elle réservée à la personne ayant créé la paire de clés. Lorsque deux personnes, ou plus, souhaitent échanger des données confidentielles, une personne, nommée par convention Alice prend en charge la création de la paire de clés, envoie sa clé publique aux autres personnes Bob, Carole… qui peuvent alors chiffrer les données confidentielles à l'aide de celle-ci puis envoyer les données chiffrées à la personne ayant créé la paire de clés, Alice. Cette dernière peut alors déchiffrer les données confidentielles à l'aide de sa clé privée.

5.1.1. En utilisant RSA, codez le mot "hello" avec p = 3 et q = 11. Appliquez l'algorithme de déchiffrement pour retrouver le message en clair d'origine.

6. AUTHENTIFICATION

L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une entité (personne, ordinateur...), afin d'autoriser l'accès de cette entité à des ressources (systèmes, réseaux, applications...). L'authentification permet donc de valider l'authenticité de l'entité en question.

L'authentification peut inclure une phase d'identification, au cours de laquelle l'entité indique son identité. Cependant, cela n'est pas obligatoire ; il est en effet possible d'avoir des entités munies de droits d'accès mais restant anonymes.

6.1. Signature numérique

La signature numérique est un mécanisme permettant d'authentifier l'auteur d'un document électronique et de garantir son intégrité, par analogie avec la signature manuscrite d'un document papier. Un mécanisme de signature numérique doit présenter les propriétés suivantes :

- Il doit permettre au lecteur d'un document d'identifier la personne ou l'organisme qui a apposé sa signature.
- Il doit garantir que le document n'a pas été altéré entre l'instant où l'auteur l'a signé et le moment où le lecteur le consulte.

Pour cela, les conditions suivantes doivent être réunies :

- L'identité du signataire doit pouvoir être retrouvée de manière certaine.

- La signature ne peut pas être falsifiée.
- La signature n'est pas réutilisable. Elle fait partie du document signé et ne peut être déplacée sur un autre document.
- Un document signé est inaltérable. Une fois qu'il est signé, on ne peut plus le modifier.

La signature électronique n'est devenue possible qu'avec la cryptographie asymétrique.

Elle se différencie de la signature écrite par le fait qu'elle n'est pas visuelle, mais correspond à une suite de nombres.

6.2. Exercices

6.2.1. Que signifie, pour un document signé, le fait d'être vérifiable, non falsifiable et non répudiable ?

6.2.2. A quoi sert un *nonce* dans un protocole d'authentification ?

6.2.3. Dans quel sens un nonce est-il une valeur unique dans une vie (*once in a lifetime*) ? Dans quelle vie ?

7. INTÉGRITÉ

De manière générale, l'intégrité désigne l'état de quelque chose qui est sain, qui n'a pas subi d'altération. En fait, l'expression intégrité des données correspond à deux notions légèrement différentes, selon que le contexte est celui des télécommunications et de la cryptographie.

Si le principe général est le même – les données ne doivent pas avoir été modifiées depuis leur création, à comprendre au sens large (écriture sur un support de stockage, transmission ...) – la cryptographie veut pouvoir affirmer que les données ont ou n'ont pas été modifiées, ce qui se fait souvent via une fonction de hachage ou, mieux, un MAC (Message Authentication Code) qui ajoute l'usage d'une clé secrète, tandis qu'en télécommunication, on souhaite simplement pouvoir détecter et souvent corriger ces modifications.

7.1. Fonction de hachage

Une fonction de hash (anglicisme) ou fonction de hachage est une fonction qui associe à un grand ensemble de données un ensemble beaucoup plus petit (de l'ordre de quelques centaines de bits) qui est caractéristique de l'ensemble de départ. Cette propriété fait qu'elles sont très utilisées en informatique, en particulier pour accéder rapidement à des données grâce aux tables de hachage (*hash tables*). En effet, une fonction de hachage permet d'associer à une chaîne de caractères un entier particulier. Ainsi, connaissant l'empreinte des chaînes de caractères stockées, il est rapide de vérifier si une chaîne se trouve ou non dans cette table (en O(1) si la fonction de hachage est suffisamment bonne). Les fonctions de hachage sont aussi extrêmement utiles en cryptographie.

Une fonction de hachage cryptographique est utilisée entre autres pour la signature électronique, et rend également possible des mécanismes d'authentification par mot de passe sans stockage de ce dernier. Elle doit être résistante aux collisions, c'est-à-dire que deux messages distincts doivent avoir très peu de chances de produire la même signature. De par sa nature, tout algorithme de hachage possède des collisions mais on considère le hachage comme cryptographique si les conditions suivantes sont remplies :

- il est très difficile de trouver le contenu du message à partir de la signature (attaque sur la première préimage) ;

- à partir d'un message donné et de sa signature, il est très difficile de générer un autre message qui donne la même signature (attaque sur la seconde préimage) ;
- il est très difficile de trouver deux messages aléatoires qui donnent la même signature (résistance aux collisions).

Par très difficile, on entend « techniquement impossible » que ce soit au niveau algorithmique ou matériel. Le MD5 par exemple n'est plus considéré comme sûr car on a trouvé deux messages qui génèrent la même empreinte. Toutefois, la mise en œuvre de ces techniques n'est pas aisée et dans le cas du MD5, les chercheurs ont trouvé une collision sur deux messages au contenu aléatoire. On peut cependant construire à partir d'une collision des attaques réelles.

7.1.1. Calculez un 3^{e} message dont la somme de contrôle est identique aux deux autres représentés ci-dessous :

message 1	codes ASCII	message 2	codes ASCII
I O U 1	49 4F 55 31	I O U 9	49 4F 55 39
0 0 . 9	30 30 2E 39	0 0 . 1	30 30 2E 31
9 B 0 B	39 42 D2 42	9 B 0 B	39 42 D2 42
	B2 C1 D2 AC		B2 C1 D2 AC

7.1.2. Pour quelle raison les fonctions de hachage constituent-elles un meilleur moyen de vérifier l'intégrité qu'une somme de contrôle tel que le *checksum Internet* ?

7.1.3. Est-il nécessaire de chiffrer les messages numériquement signés ?

7.1.4. Pour quelle raison privilégie-t-on les fonctions de hachage pour la génération de signatures numériques aux systèmes à clef publique ?

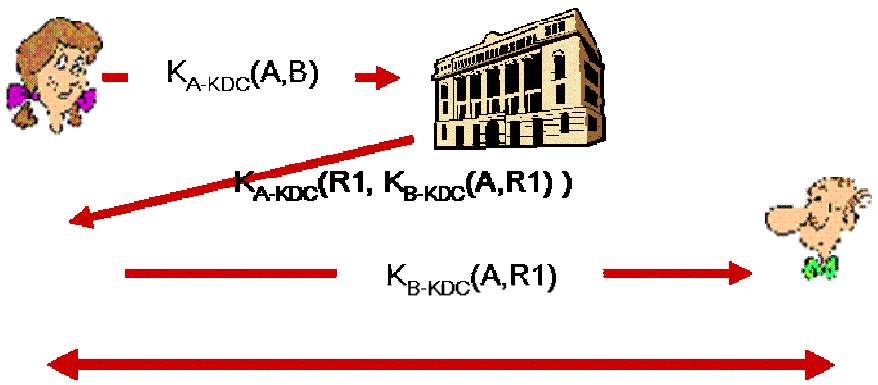
7.1.5. Le protocole de routage Internet BGP utilise un message digest MD5 pour signer les messages BGP. Pourquoi pensez-vous que le MD5 a été préféré au chiffrement de l'intégralité du message par clé publique ?

8. DISTRIBUTION DES CLEFS

8.1. Quelles différences y a-t-il entre un centre de distribution de clés et une autorité de certification ? Rappelez le fonctionnement de chacun de ces intermédiaires de confiance.

8.2. Supposons que le centre de distribution de clefs tombe en panne. Quel est l'impact de cette panne sur la capacité des différentes parties à communiquer de manière sécurisée ? Supposons à présent que l'autorité de certification tombe en panne. Quel en est l'impact ?

8.3. Dans l'échange tel que représenté dans la figure ci-dessous, Comment Alice peut-elle faire pour s'assurer que la réponse provient bien du KDC et non d'un intrus qui procède à une attaque par rejeu ? Pourquoi Alice n'a-t-elle pas besoin d'authentifier Bob explicitement ?



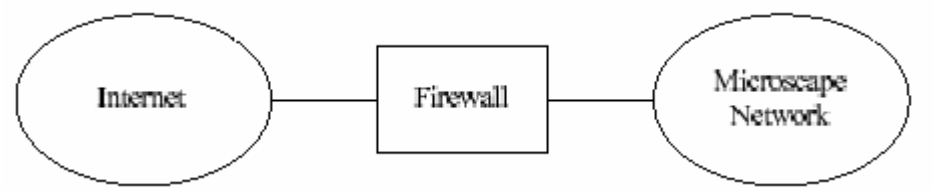
8.4. Pourquoi n'y a-t-il pas d'authentification explicite entre le KDC et Alice dans le protocole de la figure ci-dessus ? L'authentification est-elle nécessaire ? Pourquoi ?

9. IPSEC

9.1. Résumez les différences entre les services fournis par le protocole Authentication Header protocol et le protocole Encapsulation Security Payload (ESP) dans IPsec.

10. FIREWALLS

M. Dupont est administrateur réseau chez Microscape. Son premier travail est d'installer un firewall pour l'entreprise. Il décide d'utiliser un simple firewall à filtrage de paquets. Malheureusement, M. Dupont ne connaît pas très bien les firewalls et il a besoin d'aide pour installer son système. La topologie du système est illustrée sur la figure ci-dessous. Le réseau Microscape utilise des adresses 10.1/16.



Les règles pour ce firewall sont décrites par les règles simples du tableau ci-dessous. Les entrées de type matching de préfixe (128.32/16) ou * sont possibles. Les paquets qui ne correspondent à aucune règle sont éliminés.

Src Addr	Dst Addr	Src Port	Dst Port	Protocol	Action
128.32/16	*	*	telnet	TCP	discard
10.1/16	*	*	sendmail	TCP	allow

La première règle empêche les hôtes du réseau 128.32/16 d'effectuer des telnets dans le réseau de Microscape ; la deuxième règle permet aux hôtes du réseau de Microscape d'envoyer des mails à des hôtes de l'Internet. Ces règles peuvent également autoriser ou interdire d'autres trafics.

10.1. Ecrivez des règles simples autorisant les employés de Microscape de naviguer sur le Web. Rendez ces règles aussi restrictives que possible (i.e. on ne doit pas laisser d'autre trafic entrer ou sortir de Microscape si c'est possible)..

10.2. Supposons que le réseau de Microscape contienne deux hôtes (A et B). On fait l'hypothèse que le firewal ne contient que les règles ajoutées dans la partie a. Un attaquant dans l'Internet peut-il effectuer une attaque par déni de service de bande passante qui interfère avec le trafic entre A et B ? Pourquoi ?

10.3. M. Dupont installe un proxy cache HTTP dans le réseau de Microscape. Il veut s'assurer que tous les clients de Microscape utilisent ce proxy pour naviguer sur le Web. L'adresse du proxy est 10.1.2.3. Comment doit-il modifier les règles établies en a pour atteindre ce but ?

10.4. En prenant en compte l'installation obtenue dans c et en supposant que le proxy Web n'est pas sur l'un des liens entre les hôtes A et B, les transferts entre A et B peuvent-ils être affectés par une attaque par déni de service provenant d'un attaquant dans l'Internet ?

11. ATTAQUES

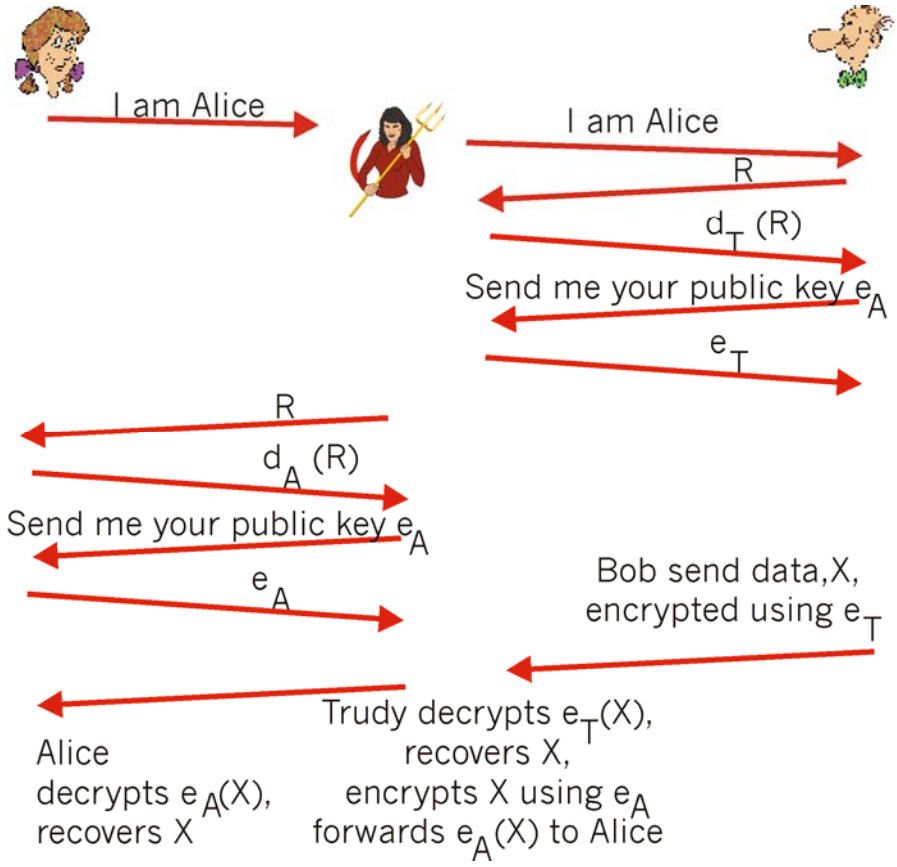
11.1. Quelle est la différence entre un intrus actif et passif ?

11.2. Supposons qu'un intrus possède un message chiffré ainsi que la version déchiffrée de ce message. L'intrus peut-il faire une attaque par texte chiffré (*cipher text*), texte connu (known-*plaintext*) ou texte choisi (*chosen-plaintext*) ?

11.3. Qu'est-ce que l'attaque de « l'homme du milieu » (*man-in-the-middle*) ? Cette attaque peut-elle se produire lorsque des clés symétriques sont utilisées ?

11.4. Considérez le protocole d'authentification 4.0 (chiffrement d'un nonce avec clé symétrique), dans lequel Alice s'authentifie auprès de Bob. Supposons maintenant qu'en même temps qu'Alice s'authentifie auprès de Bob, Bob doit s'authentifier auprès d'Alice. Donnez un scénario dans lequel Trudy, prétendant être Alice, peut maintenant s'authentifier en tant qu'Alice. (Indice : considérez la séquence des opérations du protocole ap4.0, une avec Trudy et une avec Bob qui commencent. Attention au fait que Bob et Alice utiliseront un nonce, et que le même *nonce* peut être utilisé malicieusement si l'on ne prend pas de précautions).

11.5. Dans l'attaque *man-in-the-middle* de la figure ci-dessous, Alice n'a pas authentifié Bob. Si Alice exigeait l'authentification de Bob en utilisant *ap5.0*, l'attaque man-in-the-middle pourrait-elle être évitée ? (Rappel : protocole ap5.0 : utilisation d'un nonce, chiffré en réponse avec la clé privée de la personne qui cherche à s'authentifier)



11.6. Max la Menace a mis un cheval de Troie dans votre navigateur Web préféré. C'est ennuyeux, car vous utilisez fréquemment votre browser Web pour accéder à votre compte bancaire en ligne… Max a déjà installé la backdoor pour obtenir discrètement votre numéro de compte et votre mot de passe ; maintenant, il doit trouver un moyen de sortir ces infos de votre ordinateur et les ramener vers lui par le réseau.

Proposez plusieurs canaux sécurisés (couverts) que Max pourrait utiliser pour transmettre vos secrets bancaires à un complice quelque part dans le réseau. Pouvez-vous en trouver un qui ne puisse pas être facilement détecté par un défenseur qui peut sniffer tout le trafic du réseau ?

11.7. Chacune des questions suivantes s'intéresse à une caractéristique particulière d'un protocole ; l'objectif est de savoir si cette caractéristique peut poser un problème de sécurité.

11.7.1. Considérons la structure de l'en-tête TCP, en particulier le champ *numéro de séquence*. Du point de vue de la sécurité, pourquoi serait-ce une mauvaise idée que le numéro de séquence soit prévisible ? Supposons que vous soyez un étudiant de Paris 6 et que vous sachiez que votre ami de Paris 12 est en train de faire un telnet dans le cluster. Quels dommages pourriez-vous faire ?

11.7.2. Souvenez-vous que tout réseau possède une adresse de diffusion. Les messages envoyés à cette adresse de diffusion sont envoyés à tous les ordinateurs du réseau. Par exemple, si je veux connaître les autres ordinateurs de mon réseau, je peux pinguer l'adresse de diffusion et tous les ordinateurs connectés répondront à mon ping. Pourquoi serait-ce une mauvaise idée que les hôtes répondent aux pings de diffusion (c'est-à-dire les echo requests d'ICMP, envoyées à l'adresse de diffusion) ? Quel type de problème cela pourrait-il causer ?

11.7.3. Une autre attaque classique de Déni de Service est l'attaque SYN. Dans une attaque SYN, un attaquant envoie à une victime un flot de paquets SYN avec des adresses sources usurpées. La victime initialise des états de connexion et essaie de répondre aux adresses

spoofées. Si suffisamment de paquets SYN sont envoyés, la table de connexions d'un serveur peut être remplie, et les nouvelles requêtes seront refusées. Proposez des solutions pour résoudre ce problème. Analysez les forces et faiblesses de vos solutions.

LA MULTIDIFFUSION AVEC IP

1. INTRODUCTION

La **multidiffusion** (ou ***multicast***) est le service de communication point à multipoint qui offre un moyen efficace de diffuser des unités de données à un groupe de récepteurs, en ce sens qu'une seule copie de chaque unité est injectée dans le réseau. Les unités de données sont ensuite dupliquées par les routeurs situés aux embranchements d'un arbre recouvrant. Cet arbre est construit par le protocole de routage multidestinataire sous-jacent de manière à ce que les duplications aient lieu au plus près des récepteurs. Les copies de chaque unité de données sont ainsi réduites au nombre strictement nécessaire pour que l'ensemble des récepteurs du groupe en reçoive une seule. Un service de diffusion permet ainsi de préserver les ressources du réseau telle que sa bande passante. *IP multicast*, le protocole qui rend un service de multidiffusion dans l'Internet est un standard Internet (RFC 1112) paru en 1989 suite aux travaux de Steve Deering au *Xerox Palo Alto Research Center*.

1.1 Si plutôt que d'utiliser un service de multidiffusion directement fourni par la couche réseau (tel que *IP Multicast*), on devait se contenter d'un service point-à-point (*unicast*) tel que celui rendu par la pile TCP/IP, quelle serait la charge supplémentaire qui pèserait sur la source ?

1.2 L'objectif de cet exercice est de comparer la bande passante consommée par une transmission multidestinataire (point-à-multipoint) réalisée en utilisant un service de multidiffusion rendu par la couche réseau à celle consommée en émulant une telle transmission par plusieurs transmissions point-à-point (*unicast*).

Considérons une session multidestinataire constituée d'une source émettrice unique et de 32 récepteurs. Source émettrice et récepteurs sont connectés par un arbre binaire.

a. Comparer le coût d'une transmission multidestinataire réalisée en utilisant un service de multidiffusion tel que celui rendu par IP Multicast à celle émulée par plusieurs transmissions point-à-point. Le coût sera calculé en comptabilisant le nombre de fois que l'unité de données à diffuser ou qu'une copie de celle-ci est relayée sur un lien.

b. Que pouvez-vous dire du nombre de duplications nécessaires selon que l'on utilise un service de multidiffusion tel que celui rendu par IP Multicast ou une émulation basée sur plusieurs transmissions point-à-point ?

c. Identifiez la topologie d'interconnexion de l'émetteur avec les récepteurs qui rend l'utilisation du service de multidiffusion au niveau réseau la moins coûteuse par rapport à l'émulation basée sur plusieurs transmissions ? Vous êtes libre de choisir autant de routeurs que vous le souhaitez.

2. FORMAT DES ADRESSES IP DE MULTIDIFFUSION

Parmi l'ensemble des adresses IP, toutes classes confondues, on distingue trois grands types d'adresses IPv4 :

- les adresses de *unicast* (transmission unidestinataire ou point-à-point),
- les adresses de *broadcast* (diffusion générale) et

- les adresses de *multicast* (diffusion restreinte, transmission multidestinataire ou multidistribution).

2.1 Quelle est la classe des adresses IPv4 de multidiffusion ? Comment se différencient les adresses IP de cette classe de celles des autres classes ?

2.2 Comment la fourniture d'un service de diffusion est-elle perçue à la périphérie du réseau ? IP assure-t-il une garantie de remise supplémentaire (autre que *best-effort*) pour ces paquets ?

3. CORRESPONDANCE ENTRE ADRESSES DE CLASSE D ET ADRESSES ETHERNET

La portion des adresses MAC IEEE-802 de multidiffusion[1] dont les 18 bits de poids fort valent en hexa 01:00:5E: a été réservée à l'initiative de l'IANA (*Internet Assigned Numbers Authority*) pour rendre possible leur mise en correspondance avec les adresses IPv4 de multidiffusion. La procédure de mise en correspondance est simple : elle consiste à reporter les 23 bits de poids faible d'une adresse de classe D dans ceux d'une adresse MAC (IEEE 802) de multidiffusion appartenant à la plage réservée par l'IANA.

3.1. En appliquant cette procédure, quelle est l'adresse MAC (IEEE 802) qui résulte de la mise en correspondance de l'adresse 224.10.8.5 ? Et celle obtenue avec l'adresse 224.138.8.5 ?

3.2. Quel est l'intérêt de rendre possible cette mise en correspondance ? Pour quel type de support physique est-il intéressant de mettre en œuvre cette procédure ?

3.3. Cette procédure est-elle bijective ? Cela pose-t-il un quelconque problème ?

3.4. Au vue des problèmes posées par la mise en correspondance entre adresse de classe D et adresse Ethernet de multidiffusion, est-il possible qu'une machine hôte reçoive un paquet IP diffusé à l'intention d'un groupe de diffusion auquel la machine hôte n'est pas abonné ? Calculez la probabilité qu'une telle erreur survienne.

3.5. Dans Internet, une session multidestinataire est constituée d'au moins une source émettrice et d'un groupe de récepteurs identifié par une adresse de multidiffusion. En supposant que l'attribution des adresses de classe D se fasse sans concertation, quelle est la probabilité que la même adresse IP Multicast soit choisie pour identifier les groupes de deux sessions multidestinataires concurrentes ? Indication : il vous sera nécessaire de déterminer la taille de l'espace d'adressage des adresses *IP Multicast*.

Supposons à présent qu'il y ait non plus 2 mais 1000 sessions multidestinataires en parallèle dans le réseau. En supposant que l'attribution des adresses *IP Multicast* se fait sans concertation, quelle est alors la probabilité que ces sessions interfèrent les unes avec les autres ?

3.6. Une machine hôte doit préalablement s'être abonnée au groupe de multidiffusion avant de pouvoir recevoir les données diffusées à l'intention des membres de ce groupe. Pour ce faire, la machine hôte doit-elle remplacer son adresse IP par l'adresse de multidiffusion qui permet de joindre l'ensemble des récepteurs du groupe de multidiffusion ?

3.7. Comment se fait la remise d'un paquet dont l'adresse de destination est de classe D sachant que la source et les récepteurs du groupe de multidiffusion identifié par cette adresse appartiennent au même réseau Ethernet ? Est-il nécessaire que la source sache si tous les récepteurs appartiennent au même réseau physique avant de procéder à la transmission multidestinataire de ses paquets ?

---

[1] Les adresses MAC (IEEE 802) dont le bit de poids le plus faible de leur premier octet est positionné à 1 sont des adresses de multidiffusion.

3.8. Identifiez les problèmes à traiter si à présent source et récepteurs du groupe de multidiffusion n'appartiennent plus au même réseau physique. Proposer une esquisse des mécanismes à mettre en œuvre pour venir à bout de ces problèmes et rendre possible les transmissions multidestinataires ? Ces mécanismes affectent-ils le comportement de la source tel que décrit dans la question précédente ?

4. INTERNET GROUP MANAGEMENT PROTOCOL (IGMP)

IGMP (RFC 1112) est un protocole exécuté entre les ordinateurs hôtes d'un même réseau physique et le (ou les) routeur(s) qui assure(nt) l'interconnexion de ce réseau (au reste de l'Internet). Les mécanismes de ce protocole permettent à un ordinateur hôte d'informer le routeur local de son réseau d'appartenance qu'il désire s'abonner à un groupe de multidiffusion donné. Pour ce faire, l'ordinateur hôte envoie au routeur un message appelé *Host Membership Report* dans lequel il spécifie l'adresse de diffusion qui identifie le groupe auprès duquel il souhaite s'abonner. Le routeur lui fait dès lors parvenir les paquets IP dont le champ d'en-tête « destination » contient l'adresse de classe D identifiant ce groupe de multidiffusion. Les routeurs envoient également des messages appelés *Host Membership Query*. Ces messages permettent aux routeurs d'interroger à intervalles réguliers les ordinateurs hôtes qui s'étaient abonnés à un groupe de multidiffusion. Sur réception d'un message *Host Membership Query*, un ordinateur hôte toujours abonné à ce groupe retourne un message appelé *Host Membership Report*.

4.1. Pourquoi un routeur qui exécute IGMP interroge-t-il à intervalles réguliers les ordinateurs hôtes *abonnés* des réseaux physiques qu'il interconnecte ? Quand cesse-il de leur faire ?

4.2. Plusieurs routeurs peuvent assurer l'interconnexion d'un même réseau physique. Est-il concevable que tous s'occupent d'envoyer des *Host Membership Queries* ?

4.3. Sur réception d'un message *Host Membership Query*, un ordinateur hôte toujours abonné à au moins un groupe retourne un message appelé *Host Membership Report* dont il temporise l'envoi. Pourquoi ? Quelle est la durée d'un temporisateur qui précède l'envoi des *Host Membership Reports* ? Que fait un récepteur lorsqu'il voit passer un *Report* pour un groupe de diffusion auquel il est toujours abonné ? Les *Reports* sont-ils adressés en point-à-point au routeur à l'origine du *Query* ?

4.4. Pourquoi lorsqu'un ordinateur hôte décide de s'abonner à un groupe de diffusion en informe-t-il immédiatement son routeur et n'attend-t-il pas de répondre à un *Query* ?

4.5. Quels sont les états qu'un routeur crée et maintient à la réception d'un *Host Membership Report* ? S'agit-il de la liste détaillée de tous les ordinateurs hôtes abonnés à des groupes de diffusion ? Maintient-il une telle liste par groupe de multidiffusion ? Quand décide-t-il de supprimer ces états ?

4.6. Dans IGMP Version 2, deux nouveaux formats de messages ont été définis : les messages *Group-Specific Query* et *Leave Group*. Les messages *Leave Group* permettent aux récepteurs de manifester explicitement leur volonté de se désabonner d'un groupe de diffusion donné. Quelles sont les raisons qui ont motivé l'introduction de ces nouveaux messages ?

4.7. Dans IGMP Version 3, les messages *Host Membership Report* ont été étendus pour contenir, une adresse supplémentaire, celle d'une des sources émettrices de la session multicast. Expliquez pourquoi ?

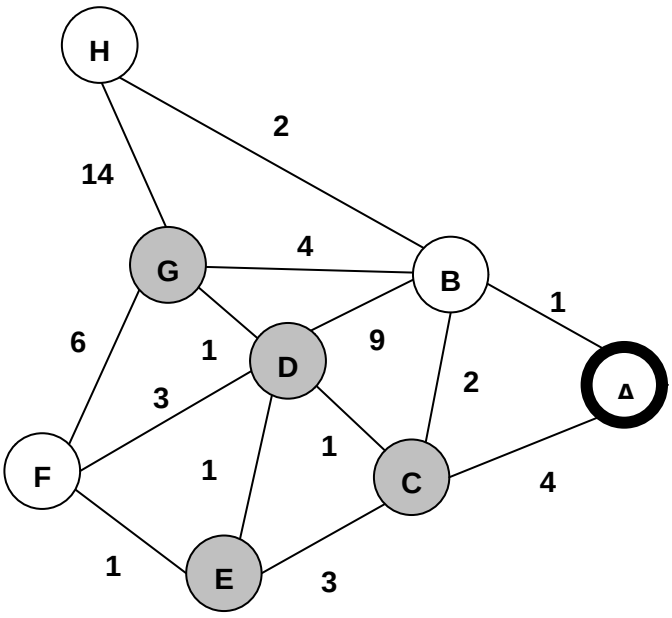
5. ALGORITHMES DE ROUTAGE MULTIDESTINATAIRE

Lorsque la source et les membres d'un groupe de multidiffusion n'appartiennent pas au même réseau physique, les routeurs doivent exécuter un protocole de gestion des groupes de multidiffusion tel que IGMP. Un tel protocole permet d'informer les routeurs s'il existe, sur les réseaux physiques qu'ils interconnectent, des ordinateurs hôtes abonnés à des groupes de multidiffusion. Les routeurs doivent également exécuter un protocole de routage multidestinataire qui construit un arbre de multidistribution (*multicast tree*). Cette structure regroupe l'ensemble des routes qui permettent aux routeurs de joindre les membres des groupes de multidiffusion *actifs* à un instant donné. Les routeurs situés aux embranchements d'un arbre de multidistribution ainsi construit relayent les paquets IP à (multi)diffuser en les dupliquant le long de cet arbre.

Les algorithmes potentiellement utilisables par les protocoles de routage multidestinataire sont les suivants :

- Flooding
- Spanning Tree
- Reverse Path Broadcasting (RPB)
- Truncated Reverse Path Broadcasting (TRPB)
- Reverse Path Multicasting (RPM)
- Core-Based Trees

5.1. La figure représente un réseau contenant 8 noeuds, étiquetés de A à H. Construire et représenter l'arbre recouvrant de coût minimal (*minimal cost spanning tree*) de racine A qui permet de joindre les nœuds C, D, E et G. Argumentez (de manière informelle) le fait que l'arbre recouvrant construit est bien celui de coût minimal. Pour ce faire, vous pourrez montrer qu'aucun des autres arbres ne permet à A de joindre les nœuds C, D, E et G avec un coût inférieur à celui que vous proposez.



5.1. Flooding (inondation)

Le moyen le plus simple pour remettre des paquets multidiffusés est d'implémenter un algorithme d'inondation (*flooding*). Un algorithme qui procède par inondation démarre lorsqu'un routeur reçoit un paquet à multidiffuser. Le routeur duplique ce paquet sur toutes ses interfaces autres que celle sur laquelle il a reçu le paquet. Une telle procédure garantit à tous les paquets de visiter l'ensemble des routeurs du réseau.

5.1.1. Pourquoi le routeur ne duplique-t-il pas les paquets sur toutes ses interfaces ? Cette précaution est-elle suffisante ?

5.1.2. Si un tel algorithme présente l'intérêt de ne pas installer de table de routage au niveau des routeurs, il ne passe pas le facteur d'échelle. Expliquer pourquoi.

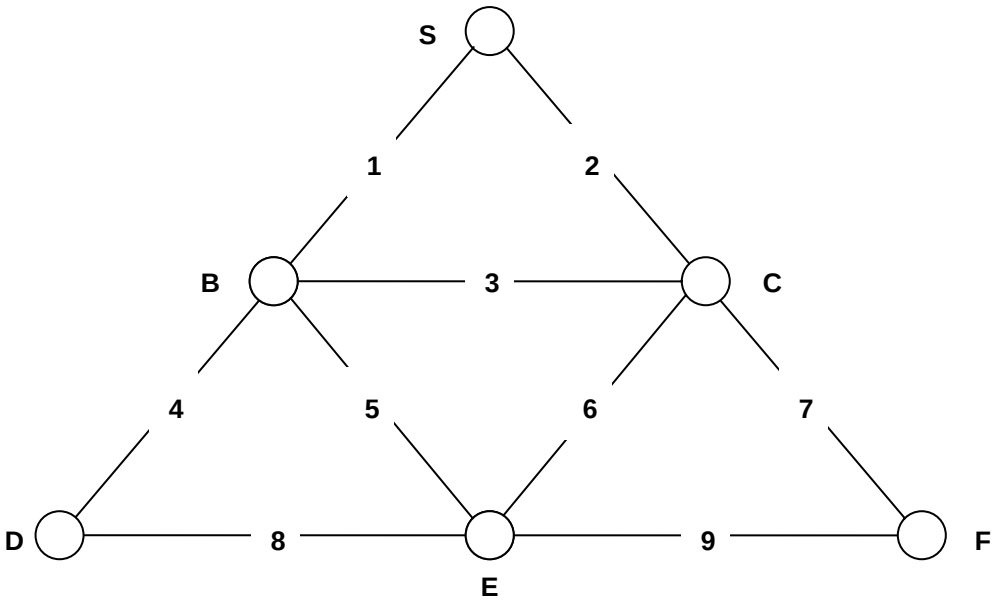
5.2. Reverse Path Broadcasting (RPB)

L'algorithme RPB (*Reverse Path Broadcasting*) construit un arbre de diffusion différent par couple (source, groupe de multidiffusion). Le sommet d'un tel arbre est la source tandis que parmi ses feuilles on trouve les membres du groupe. Cet algorithme fonctionne de la manière suivante : si un routeur reçoit un paquet sur une interface qu'il détermine comme étant celle qu'il aurait choisie pour joindre la source selon le plus court chemin, le routeur relaye alors le paquet sur l'ensemble de ses autres interfaces que nous appellerons ses interfaces *sortantes*. L'interface par laquelle sont reçus les paquets ainsi relayés sera quant à elle désignée comme étant l'interface *entrante* du routeur. Les paquets reçus par une autre interface que l'interface *entrante* sont rejetés.

5.1.2. L'algorithme tel que spécifié peut être amélioré en réduisant le nombre de paquets inutilement dupliqués dans le réseau. Cette amélioration requiert d'un routeur d'être en mesure de savoir s'il se trouve sur le chemin le plus court que chacun de ces voisins aurait choisi pour joindre la source du paquet qu'il s'apprête à leur relayer. En quoi consiste cette amélioration ?

5.1.3. Comment peut faire un routeur pour avoir cette information sur ces voisins immédiats ?

5.1.4. Exécuter l'algorithme RPB amélioré sur le réseau suivant : l'algorithme démarre alors que le routeur B reçoit sur le lien 1 un paquet multidiffusé issu du routeur qui connecte S.



5.1.5. Quels intérêts présente l'utilisation de l'algorithme RPB ? Quel est son principal inconvénient ?

5.3. Truncated Reverse Path Broadcasting (TRPB)

L'algorithme TRPB (*Truncated Reverse Path Broadcasting*) a été conçu pour venir à bout des limitations de l'algorithme RPB. Comme RPB, l'algorithme TRPB produit un arbre de multidiffusion différent par couple (source, groupe de multidiffusion). La principale différence entre ces deux algorithmes provient du fait est que dans TRPB, les routeurs utilisent les états installés par IGMP. Ces états permettent aux routeurs de découvrir la présence d'ordinateurs hôtes abonnés à des groupes sur chacun des réseaux physiques qu'ils connectent au reste de l'arbre de multidiffusion et de connaître l'identité de ces groupes. Un routeur évite ainsi de relayer des paquets dans des sous-réseaux où il n'existe aucun abonné aux groupes de multidiffusion en cours. On dit alors que les routeurs élaguent l'arbre de diffusion de ses feuilles qui n'ont pas lieu d'être.

5.3.1. L'algorithme TRPB vient-il complètement à bout des limitations de l'algorithme RPB ?

5.4. Reverse Path Multicasting (RPM)

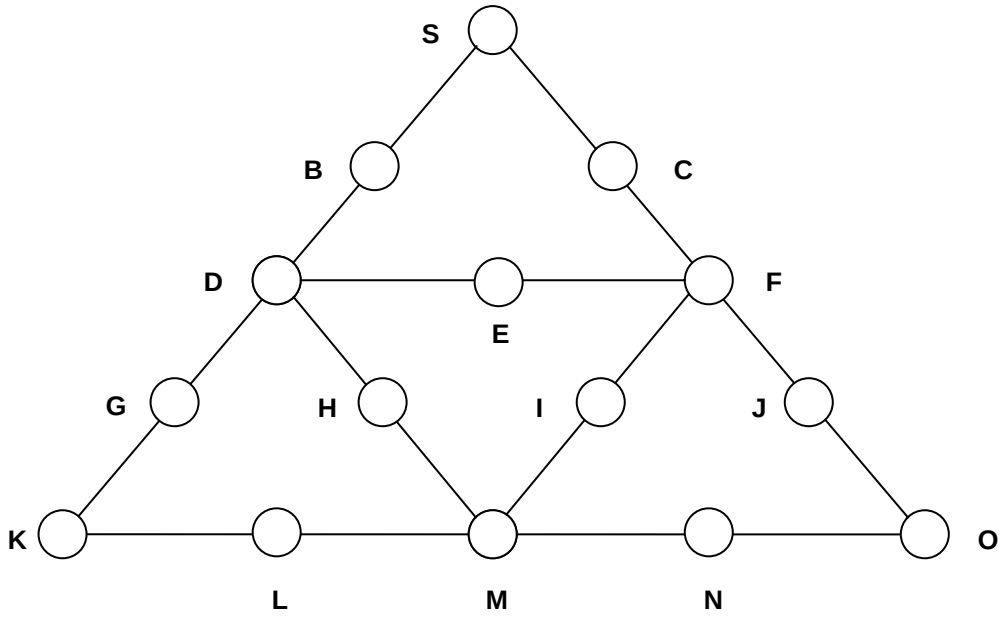
L'algorithme RPM (*Reverse Path Multicasting*) est une amélioration apportée aux algorithmes RPB et TRPB. Comme ces autres algorithmes, RPM crée un arbre de multidiffusion différent par couple (source, groupe de multidiffusion). RPM est un algorithme de type *reverse-path multicast* qui procède par inondation et élagage successifs. Deux protocoles de routage multidestinataires couramment déployés utilisent une telle approche : DVMRP (*Distance Vector Multicast Routing Protocol*) et PIM-DM (*Protocol Independant Multicast Dense-Mode*). Les propriétés de l'arbre que RPM construit pour un couple (S,G) donné sont les suivantes :

- les feuilles de cet arbre sont les routeurs qui connectent les ordinateurs hôtes abonnés à G ;
- les routes que regroupe cet arbre sont celles qui permettent à S de joindre chacune de ces feuilles selon le plus court chemin. Les routeurs et réseaux physiques que traversent ces routes connectent et hébergent respectivement des membres de G.

Pour construire un tel arbre, l'algorithme RPM opère de la manière suivante : lorsqu'un routeur reçoit de S un premier paquet dont la destination est G, le routeur le relaye le long d'un arbre de diffusion identique à celui que construit l'algorithme TRPB pour (S,G). Ce premier paquet est donc reçu par l'ensemble des routeurs feuilles de l'arbre TRPB. En se basant sur les états localement installés par IGMP, les routeurs feuilles découvrent la présence d'ordinateurs hôtes abonnés à G sur chacun des réseaux physiques qu'ils connectent au reste de l'arbre TRPB. Un routeur feuille qui ne connecte aucun membre de G retourne un message d'élagage dit *prune message* sur son interface *entrante* dans l'arbre TRPB. Il incite ainsi le routeur immédiatement situé en amont à ne plus relayer de paquets sur l'interface *sortante* par laquelle ce dernier reçoit le message *prune*. Si le routeur *amont* reçoit un message *prune* sur ses autres interfaces sortantes, celui-ci génère à son tour un message *prune*. Le processus de génération des messages *prune* permet d'élaguer toutes les branches inutiles de l'arbre TRPB : les branches qui subsistent à ce processus sont celles qui permettent à la source de joindre les seuls réseaux physiques qui hébergent des membres du groupe G.

5.4.1. Lorsqu'un routeur reçoit un message *prune*, il élague en conséquence la branche concernée et enregistre les informations relatives à cet élagage. Quelles sont d'après vous les informations enregistrées ?

5.4.2. Exécuter l'algorithme RPM sur l'arbre suivant : l'algorithme démarre alors que le routeur B reçoit un paquet multidiffusé issu du routeur S. Les états installés par RPM seront spécifiés pour chacun des routeurs.



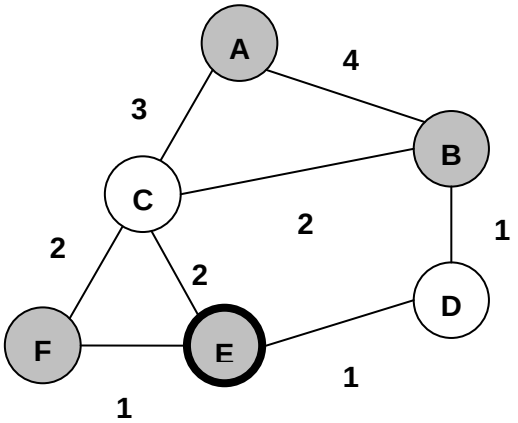
5.4.3. Les messages *prune* sont-ils conçus pour être relayés plus d'un saut en amont du routeur à l'origine du message *prune* ?

5.4.4. Que se passe-t-il lorsqu'un ordinateur hôte décide de s'abonner à un groupe G alors que la branche qui permettait à la source S de le joindre avait été auparavant élaguée ? Sachant que la topologie du réseau et la composition des groupes de multidiffusion changent au cours du temps, comment RPM opère-t-il pour prendre en compte et répercuter ces changements sur l'arbre de diffusion qu'il maintient ?

5.4.5. Etant donné les mécanismes mis en œuvre dans RPM et les propriétés des arbres de multidiffusion résultants, l'algorithme RPM résout-il les problèmes liés au facteur d'échelle ?

5.4.6. La figure suivante représente la topologie d'un réseau où pour chacun des liens est donné le coût. Les nœuds grisés indiquent des routeurs qui interconnectent des réseaux physiques où sont hébergés des ordinateurs hôtes abonnés au groupe de la session multidestinataire.

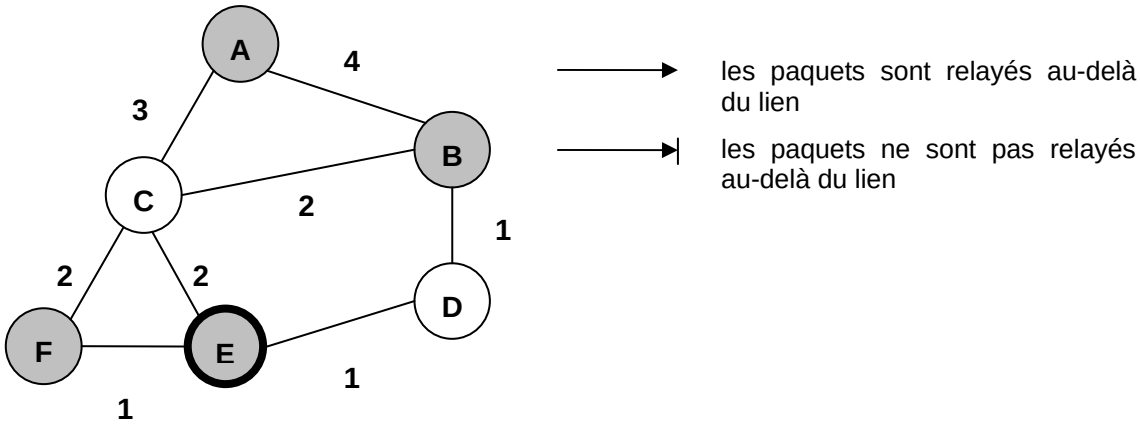
a. Supposons que le nœud E soit choisi comme source émettrice de la session multidestinataire. Construire l'arbre recouvrant de coût minimal correspondant à cette



session.

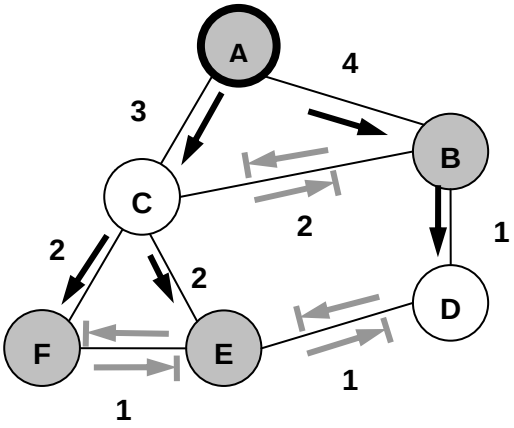
b. Reconstruire l'arbre recouvrant de coût minimal si le coût du lien entre B et D passe de 1 à 10.

5.4.7. Cet exercice reprend la topologie étudiée dans l'exercice précédent. La figure suivante



représente la topologie d'un réseau où pour chacun des liens est donné le coût. Les nœuds grisés indiquent des routeurs qui interconnectent des réseaux physiques où sont hébergés des ordinateurs hôtes abonnés à des groupes de multidiffusion.

En utilisant les deux flèches légendées dans la figure, matérialisez l'arbre de diffusion qui résulte de l'exécution de l'algorithme RPM (*Reverse Path Multicasting*) en prenant E comme source émettrice : vous indiquerez les liens le long desquels les paquets sont relayés ainsi que ceux au-delà desquels les paquets ne sont plus relayés. Indication : vous pourrez vous inspirer de la figure suivante qui matérialise l'arbre de diffusion que RPM construit quand le nœud A est la source émettrice de la session multidestinataire.



5.5. Core-Based Trees

Contrairement aux algorithmes RPB, TRPB et RPM qui construisent un arbre différent par couple (source,groupe de multidiffusion), l'algorithme CBT (*Core-Based Trees*) maintient un arbre de multidiffusion unique pour chaque groupe de multidiffusion, indépendamment de la localisation des sources potentielles. Les paquets destinés aux membres d'un même groupe de multidiffusion sont donc dupliqués le long du même arbre et ce quel que soit la source émettrice. Les arbres (de multidiffusion qui résulte de l'exécution de l'algorithme CBT sont construits autour d'un routeur appelé « cœur » (ou *core*). Plusieurs cœurs peuvent être spécifiés pour le même arbre CBT dans un souci de redondance.

Lorsqu'une machine hôte désire s'abonner à un groupe de multidiffusion G, son routeur CBT local fait correspondre l'adresse de G avec celle d'un des cœurs situés à la racine de l'arbre CBT maintenu pour le groupe G. Il envoie alors un message *join* directement adressé à ce cœur. Le message *join* est alors traité par chacun des routeurs qui sépare le futur abonné du coeur. Chacun d'eux marque l'interface par laquelle le message *join* est reçu et accuse positivement la réception du *join* au routeur précédemment visité. Une source qui désire diffuser des données aux membres du groupe G envoie ses paquets vers un des points de rendez-vous du cœur. Si ces paquets sont interceptés par un des routeurs de l'arbre que CBT maintient pour G, ils sont dupliqués sur l'ensemble des interfaces du routeur, exceptée celle par laquelle les paquets sont interceptés.

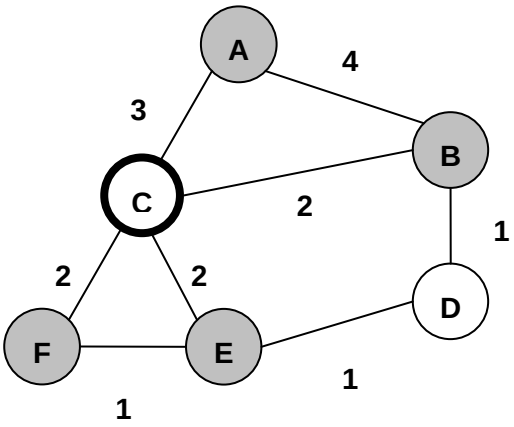
5.5.1. Quels avantages présente l'algorithme CBT du point de vue du facteur d'échelle ?

5.5.2. Comment fait un routeur de l'arbre CBT qui intercepte un paquet alors que ce dernier n'a encore atteint le cœur pour le dupliquer sur toutes ces interfaces, y compris son interface montante (entrante) dans l'arbre CBT ?

5.5.3. Que peut-on prévoir concernant la charge supportée par le cœur de l'arbre CBT ?

5.5.4. La figure suivante représente la topologie d'un réseau où pour chacun des liens est donné le coût. Les nœuds grisés indiquent des routeurs qui interconnectent des réseaux physiques où sont hébergés des ordinateurs hôtes abonnés à des groupes de multidiffusion.

Le noeud C est choisi comme cœur (ou *core*) par un algorithme de routage CBT (*Core-Based Trees*). En supposant que les messages *join* empruntent les plus courts chemins séparant les routeurs du cœur, dessinez l'arbre de multidistribution résultant de l'exécution de algorithme de routage CBT.



5.5.5. Cet exercice reprend la topologie étudiée dans l'exercice précédent. Rappelons que les nœuds grisés du réseau représentent les routeurs qui connectent des sous-réseaux sur lesquels se retrouvent les machines hôtes abonnées à la session multidestinataire en cours. Pour les besoins du présent exercice, nous supposons que les abonnés cumulent les rôles de récepteur et de source émettrice pour cette même session. Il s'agit en effet d'une session multipoint à multipoint. Nous supposons également que les routeurs qui connectent les abonnés à cette session reçoivent une unité de données par unité de temps. Cette unité de données doit alors être diffusée (acheminée) jusqu'aux autres abonnés que connectent les 3 autres routeurs grisés.

Le noeud C est choisi comme cœur (ou *core*) par un algorithme de routage CBT (*Core-Based Trees*). En supposant que les messages *join* empruntent les plus courts chemins séparant les routeurs du cœur, construisez l'arbre de multidistribution résultant de l'exécution de algorithme de routage CBT. Calculez la quantité de trafic (montant et descendant) transitant

sur chacun des liens du réseau. Cette quantité sera exprimée en unités de données par unité de temps.

Supposons à présent qu'on utilise l'algorithme RPM (*Reverse Path Multicasting*) pour construire les arbres de routage nécessaires pour acheminer les données échangées pendant la durée de la session multidestinataire. Construisez les 4 arbres de diffusion qui résultent de l'exécution de l'algorithme RPM en prenant A, B, E et F comme source émettrice. Calculez la quantité de trafic que génère dans ce cas la session muldestinataire. Déterminez lequel des deux algorithmes RPM et CBT construit des arbres qui tendent à concentrer le trafic ?

5.6. PIM-SM (Protocol Independent Multicast Sparse-Mode)

Les travaux sur les algorithmes de type CBT se sont poursuivis pour venir à bout de leurs limitations persistantes tout en gardant les bonnes propriétés des arbres partagés. Ces travaux ont donné naissance à PIM Sparse-Mode (RFC 2362). Dans PIM-SM, les phases de découverte des sources et de construction de l'arbre partagé sont rendues indépendantes. C'est en ce sens que PIM-SIM est reconnu comme une amélioration de CBT.

L'équivalent du cœur (*core*), le routeur autour duquel est construit un arbre CBT est appelé un point de rendez-vous (*Rendezvous Point*) ou RP dans PIM. Malgré une dénomination différente, le rôle d'un RP est quasiment identique à celui d'un cœur. Lorsqu'une source émettrice commence à diffuser des données, ses paquets ont pour destination l'adresse d'un groupe. Son routeur local fait correspondre cette adresse avec celle du RP, racine de l'arbre de diffusion construit pour joindre le groupe en question. Le routeur encapsule dès lors chaque paquet de données dans un autre paquet IP directement adressé au RP. Le RP dé(sen)capsule alors les paquets avant de les diffuser le long de l'arbre partagé

Lorsqu'un récepteur s'abonne au groupe, son routeur local génère un message *join* directement adressé au RP. Le message *join* est alors traité par chacun des routeurs qui sépare le futur abonné du coeur. Chacun d'eux instancie des informations d'acheminement qui reviennent à marquer l'interface par laquelle le message *join* est reçu. Il s'agit d'informations unidirectionnelles en ce qu'elles ne peuvent servir qu'à diffuser les données en aval (depuis le RP jusqu'aux récepteurs).

Les arbres partagés unidirectionnels résultant de l'exécution de PIM-SM ne sont pas forcément les plus efficaces. Ils constituent cependant le seul moyen d'indiquer aux récepteurs que des données commencent à arriver. Sur réception des premières données, le routeur local d'un récepteur peut décider de changer d'arbre en initiant la construction d'un arbre des plus courts chemins (*short path tree* ou SPT). Pour cela, le routeur envoie un message *join* directement adressé à la source. Dès que les données commencent à arriver sur l'arbre SPT nouvellement construit, un message *prune* peut alors être retourné au point de rendezvous pour éviter de recevoir les données diffusées en double.

Contrairement aux autres protocoles qui construisent des arbres de coût minimal tels que DVMRP et PIM-DM, les informations de routage créées par PIM-SM sont maintenues uniquement le long de des arbres SPT. Dans DVMRP et PIM-DM, les informations d'élagage sont maintenues le long des chemins à l'extrémité desquels il n'y a pas de récepteur abonné à la session en cours.

5.6.1. La localisation du point de rendezvous vous paraît-elle être aussi cruciale que celle du cœur dans CBT ?

5.6.2. Merge here

6. DISTANCE VECTOR MULTICAST ROUTING PROTOCOL (DVMRP)

6.1. DVMRP : un exemple de protocole de routage multidestinataire

DVMRP (*Distance Vector Multicast Routing Protocol*) est un protocole à vecteur de distance conçu pour permettre l'acheminent des paquets IP qui attendent du réseau un service de multidiffusion. Les arbres résultant de l'exécution de DVMRP ont pour sommet respectif les sources des sessions multidestinataires en cours. Les versions successives de DVMRP ont longtemps été déployées dans la plupart des routeurs du MBONE. Si depuis, DVMRP s'est vu par endroits remplacer, il reste toujours au cœur même du routage dans le MBONE.

Initialement définies dans le RFC 1075, les spécifications de DVMRP sont inspirées du protocole de routage unidestinataire RIP (*Routing Information Protocol*) : ces deux protocoles utilisent l'algorithme à vecteur de distance aussi connu sous le nom de Bellman-Ford. La principale différence entre RIP et DVMRP réside dans le fait que

- RIP calcule l'adresse du **saut suivant** sur chacune des routes qui permettent à un routeur de joindre suivant le plus court chemin, toutes les destinations potentielles dans le réseau ;
- DVMRP calcule les routes qui séparent le routeur des sources émettrices des sessions multidestinataires en cours. En effet, un routeur DVMRP détermine pour chaque couple (source, groupe) l'adresse du **saut précédent** sur la route qui le sépare de la source selon le plus court chemin.

Pour prendre en compte les changements de composition des groupes de multidiffusion, DVMRP utilise des variantes de l'algorithme RPB (*Reverse Path Broadcasting*) : dans ses spécifications initiales, DVMRP utilisait l'algorithme TRPB (cf. § 5.3) tandis que les dernières versions de *mrouted* (≥ 3.8) et les implémentations commercialisées de DVMRP ont étendu les spécifications du RFC 1075 afin de permettre l'utilisation de l'algorithme RPM (cf. §5.4) (*Reverse Path Multicasting*).

6.1.1. Sachant que DVRMP utilise des variantes de RPB pour construire des arbres dont les sommets respectifs sont les sources de sessions *multicast*, quelles sont les informations de routage nécessaires aux routeurs DVRMP pour relayer les paquets IP *multicast* conformément aux règles spécifiques à RPB ?

6.2. Interfaces physiques

Les interfaces des routeurs DVMRP sont configurées avec une métrique qui spécifie le coût du port correspondant et le TTL seuil (*Time To Live threshold*) qui permet de limiter la portée de la transmission multidiffusée. Un paquet multidiffusé est relayé sur une interface à condition que le champ TTL de l'en-tête du paquet contienne une valeur supérieure à celle que spécifie le TTL seuil configuré pour l'interface. Le tableau 1 liste les valeurs de TTL couramment utilisées pour restreindre la portée d'un paquet IP multidiffusé sur une interface.

0	Portée localement restreinte à l'équipement
1	Portée restreinte au même réseau physique
32	Portée restreinte au même site
64	Portée restreinte au sein de la même région
128	Portée restreinte au sein du même continent
255	Portée non restreinte

Tableau 1 *Valeurs du TTL seuil des interfaces*

6.2.1. Quelle est la portée d'un paquet IP injecté dans le réseau avec un TTL inférieure à 32 ?

6.3. Opérations de base

Les dernières versions de DVMRP utilisent l'algorithme RPM (*Reverse Path Multicasting*) modifié afin de prendre en compte les TTL seuil configurés : le premier paquet d'un couple (S,G) atteint l'ensemble des routeurs feuille du réseau à condition que la valeur de son champ d'entête *TTL* et le TTL seuil des interfaces sortantes des routeurs de l'arbre de diffusion (S,G) non élagué le permettent. Est alors mis en œuvre le processus de génération des messages *prune* (cf. 5.1.4). DVMRP implémente également un mécanisme d'embranchement rapide : dès qu'il reçoit un message *join* pour le couple (S,G), un routeur responsable de l'élagage de la branche de l'arbre (S,G) dont il était la feuille, retourne un message de greffe dit *graft message*. En recevant ce message, le routeur situé immédiatement en amont dans l'arbre de diffusion annule les états installés suite à la réception du dernier message *prune*. Ce mécanisme accélère le greffage de branches précédemment élaguées.

6.3.1. Est-il nécessaire de faire remonter un message *graft* jusqu'à la source ? Si non, quel est le premier routeur à en bloquer la remontée ?

S'il existe plus d'un routeur DVMRP sur un même réseau physique, seul un de ces routeurs est responsable de générer les messages *Host Membership Query* (IGMP). Il s'agit du *routeur dominant* (DR). A l'initialisation, un routeur part du principe qu'il est le DR du réseau physique local jusqu'à ce qu'il reçoive un message *Host Membership Query* (IGMP) provenant d'un routeur dont l'adresse IP est plus petite que la sienne.

6.3.2. Pourquoi est-il nécessaire d'élire un routeur *dominant* sur un même réseau physique ?

6.3.3. La figure 1 représente un réseau physique doté de 3 routeurs DVMRP dont le routeur C qui est le plus en aval dans l'arbre de diffusion. Quel sera le routeur dominant du réseau physique représenté dans la figure 1 ?

Figure 1 *Election d'un routeur dominant*

6.4. Table de routage DVMRP

Les **tables de routage** DVMRP indiquent aux routeurs comment transmettre les paquets selon les règles spécifiques au routage DVMRP. Pour constituer et mettre à jour leur table de routage, des

informations de routage sont régulièrement échangées entre routeurs DVMRP voisins. DVMRP utilise IGMP pour véhiculer les informations de mise à jour des tables de routage (RIP utilise UDP). Ces informations sont indépendantes de celles que génèrent les protocoles de type IGP (*Interior Gateway Protocol*) exécutés en parallèle dans le réseau. DVMRP repose sur l'utilisation des mises à jour avec coût infini utilisées pour traiter les horizons partagés avec retour empoisonné (*split horizon with poisoned reverse*). DVMRP emprunte cette méthode aux algorithmes à vecteur de distance pour permettre la détection des **feuilles** (*leaves*). Les feuilles désignent ici les interfaces de routeur qui ne connectent des réseaux physiques sur lesquels aucun *récepteur* n'existe. On distingue deux types de récepteurs : les ordinateurs hôtes abonnés à un groupe de diffusion et les routeurs *descendants*[2] de l'arbre de diffusion. Si une interface connecte un réseau physique sur lequel il n'existe aucun routeur DVMRP qui considère cette interface comme étant située sur le chemin le plus court à la source alors ce réseau physique est une feuille.

La technique de l'horizon partagé *simple* interdit à un routeur d'annoncer les routes à ceux de ses voisins par lesquels il les a découvertes. Les mises à jour retournées à un routeur voisin ne concernent pas les routes calculées à l'aide des mises à jours de ce dernier. La technique de l'horizon partagé *avec empoisonnement* inclut de telles routes mais en leur fixant un coût infini (leur métrique à l'infini).

DVMRP utilise la méthode de traitement des horizons partagés *par retour empoisonné* en imposant aux routeurs d'annoncer sur leur interface entrante (i.e. l'interface qui leur permet de joindre la source selon le plus court chemin) un coût infini pour la route qui les sépare de la source selon le chemin le plus court. Si durant la durée d'un temporisateur dit de retenue (*hold down timer*), un routeur ne reçoit pas de retour empoisonné pour une source S sur une de ses interfaces sortantes, le routeur en déduit que le réseau physique auquel le connecte cette interface est un réseau feuille pour la source S : ce réseau ne connecte aucun routeur descendant, i.e. un routeur pour lequel cette interface est située sur le chemin le plus court de S. Si en plus, DVRMP détermine grâce à la variante utilisée de l'algorithme RPB qu'il n'existe aucun membre du groupe G sur le réseau feuille, cette interface sortante est retirée de sa liste des interfaces d'acheminement de sa table de transmission (*forwarding table*).

Un exemple de table de routage DVMRP est donné dans la figure 2. Contrairement aux tables de routage unidestinataire telles que celles que construit RIP, chaque entrée de la table de routage maintenue par un routeur DVMRP concerne un couple (source,groupe). L'entrée créée pour le couple (S,G) décrit la route qui sépare selon le chemin le plus court le routeur de la source S dans l'arbre que l'algorithme RPB construit pour (S,G). Comme l'indique la figure 2, les tables de routage DVMRP ne reflètent pas les changements dans la composition des groupes de multidiffusion. Elles contiennent le préfixe (l'adresse du réseau) de la source S (*Source Subnets*) et l'adresse du routeur de saut précédent sur la route qui sépare selon le plus court chemin, le routeur de la source S (*From-Gateways*).

Source Subnet	Subnet Mask	From Gateway	Metric	Status	TTL	InPort	OutPorts
128.1.0.0	255.255.0.0	128.7.5.2	3	Up	200	1	2,3
128.2.0.0	255.255.0.0	128.7.5.2	5	Up	150	2	1
128.3.0.0	255.255.0.0	128.6.3.1	2	Up	150	2	1,3
128.4.0.0	255.255.0.0	128.6.3.1	4	Up	200	1	2

Tableau 2 *Table de routage DVMRP*

[2] Routeurs *multicast* situés immédiatement en aval dans l'arbre de diffusion

- La colonne `Source Subnet` indique l'adresse du réseau physique d'appartenance (i.e. le préfixe) de la source du couple (S,G) décrit par chaque entrée.
- La colonne `Subnet Mask` spécifie le masque du réseau physique d'appartenance de la source S.
- La colonne `From Gateway` indique l'adresse du routeur de saut précédent sur le chemin qui sépare le routeur de la source du couple (S,G) décrit par l'entrée.
- La colonne `TTL` est utilisée pour la gestion des entrées de la table de routage. La valeur spécifiée indique la durée exprimée en secondes au-delà de laquelle l'entrée sera retirée de la table.

6.4.1. Pourquoi est-il nécessaire de connaître l'interface sur laquelle sont reçues les informations de routage ?

6.4.2. Construire la portion de l'arbre RPB décrite par la table de routage représentée dans le tableau 2.

6.5. Table de transmission DVMRP

Pour tenir compte de la composition dynamique des groupes de multidiffusion, DVMRP maintient une seconde table appelée **table de transmission** (*forwarding table*). Les routeurs DVMRP construisent leur table de transmission en combinant les informations contenues dans leur table de routage et celles qu'ils obtiennent concernant la composition des groupes de diffusion (messages *join* et *prune*). Les tables de transmission représentent la vision locale que les routeurs DVMRP ont des arbres de multidiffusion que l'algorithme RPM construit pour chaque couple (source de la session, groupe des abonnés à cette session). Le tableau 3 donne un exemple de table de transmission DVMRP.

Source Subnet	Multicast Group	TTL	InPort	OutPorts
128.1.0.0	224.1.1.1	200	1 Pr	2p 3p
	224.2.2.2	100	1	2p 3
	224.3.3.3	250	1	2
128.2.0.0	224.1.1.1	150	2	2p 3

Tableau 3 *Table de transmission DVMRP*

Une entrée contient le préfixe de la source (`Source Subnet`), les adresses de groupe à destination desquels la source émet des données (`Multicast Group`), l'interface sur laquelle les paquets d'un couple (source,groupe) sont attendus (`InPort`) et les interfaces sur lesquelles ces paquets doivent être relayés pour atteindre des récepteurs du groupe (`OutPorts`).

- La colonne `Source Subnet` contient l'adresse du réseau physique d'appartenance des sources émettrices de données destinées aux groupes de diffusion dont les adresses respectives sont spécifiées dans la colonne suivante.
- La colonne `Multicast Group` contient l'adresse de classes D des groupes de diffusion auxquels sont destinées les données à relayer, selon l'adresse du réseau physique de la source émettrice de ces données. Est à noter qu'un même réseau physique peut contenir plusieurs sources émettrices de données destinées au même groupe ou à des groupes différents.
- La colonne `InPort` contient le port *entrant* (*parent port*) au travers duquel les données du couple (source, groupe) sont attendues. Le mot-clef `Pr` indique que le routeur a retourné sur ce port un message *prune* directement adressé au routeur situé immédiatement en amont.

- La colonne `OutPorts` contient les ports sortants (*child ports*) sur lesquels les paquets du couple (source, groupe) sont relayés. Un ‘p’ minuscule indique que le routeur a reçu un message *prune* sur le port concerné.

6.5.1. Construire la portion de l’arbre RPB avant et après élagage décrite par la table de transmission représentée dans le tableau 3.

7. QUESTIONS À CHOIX MULTIPLES

Q1. Quelle est la plage d'adresses IP Multicast disponibles ?

Q2. A quoi sert IGMP ?

Q3. Quel avantage présente IGMPv2 sur IGMPv1 ?

Q4. Quel avantage présente Shortest Path Trees comparé à Shared Trees ?

Q5. Pourquoi est-il avantageux d'utiliser Shared Trees ?

Q6. Quelle information un routeur utilise-t-il pour vérifier le RPF

Q7. Expliquer le I(ndependent) de PIM ?

Q8. Vrai ou faux ?

a. Vrai ou faux ? IGMP est le protocole utilisé pour la communication entre les hôtes pour la gestion du groupe.

b. Vrai ou faux ? Dans le reverse path forwarding, un noeud reçoit plusieurs copies d'un même paquet.

c. Vrai ou faux ? Dans le reverse path forwarding, un nœud peut envoyer plusieurs copies d'un paquet sur le même lien sortant.

TD Qualité de Service (QoS)

I- QoS : Généralités

Donnez une description des concepts suivants, et expliquez en quoi ces points sont importants pour la garantie de QoS :

- le marquage des paquets
- l'isolation des classes
- le multiplexage des paquets venant de diverses sources
- la procédure d'admission d'appels.

Exercice 1

a. Qu'est-ce que la QoS et avec quels paramètres peut-on la mesurer ?

b. Comment la qualité de service se traduit-elle pour l'utilisateur ?

c. Quelles applications nécessitent de la QoS ?

d. Comment la QoS est-elle implémentée actuellement dans les réseaux IP ?

e. Quels sont les problèmes liés à la QoS ?

f. Qu'est-ce que la classe de service et en quoi est-ce différent de la QoS?

g. Quels problèmes de sécurité la QoS peut-elle causer ?

Exercice 2

Que signifie l'interactivité pour l'audio et la vidéo streamés ?

Que signifie l'interactivité pour l'audio et la vidéo interactifs ?

Exercice 3

Décrivez différentes approches pour faire évoluer l'Internet afin qu'il supporte mieux les applications multimédias.

Exercice 4

Quelle est la différence entre le délai de bout-en-bout et la gigue ? Quelles sont les causes de la gigue ?

Exercice 5

Pourquoi un paquet reçu après l'instant prévu de diffusion est-il considéré comme perdu ?

II- Intserv / DiffServ / MPLS

Exercice 1

Les flots doivent-ils être définis comme le trafic entre deux hôtes ou bien comme le trafic entre deux processus ? Discutez des conséquences de chaque approche pour les programmes applicatifs. Comment ces mécanismes pourraient-ils être implémentés dans les routeurs ?

Exercice 2

(a) Expliquez comment le protocole de réservation RSVP fonctionne. Quels sont les principaux messages RSVP ? Quelles sont les actions des terminaux et des routeurs lors de la réception de ces messages ?

(b) Comment RSVP gère-t-il les changements de route IP ?

Exercice 3

Quelles sont les difficultés associées au modèle Intserv et à la réservation de ressources par flot ?

Exercice 4

En quoi Intserv/RSVP et DiffServ diffèrent-ils en termes de :

- passage à l'échelle
- réservation de ressources
- signalisation
- propriétés de multicast
- compatibilité avec IPv4 ?

Laquelle de ces solutions vous semble la plus probable dans les réseaux IP ? Pourquoi ?

Exercice 5

Quels sont les principaux avantages de DiffServ par rapport à Intserv et/ou RSVP ? Est-ce que DiffServ garantit de la QoS ?

Exercice 6

L'utilisateur A demande une connexion garantissant de la QoS pour une vidéoconférence. Une connexion vidéo de bonne qualité demande une bande passante comprise entre 2 et 8 Mbps. Quels sont les facteurs qui ont un impact sur le succès de la connexion dans le réseau, lorsque la QoS est implémentée avec DiffServ ?

Exercice 7

Qu'est-ce que MPLS et comment peut-il être utilisé pour implémenter la QoS ? A-t-il des propriétés communes avec Intserv et Diffserv ? Quels sont les principaux domaines où MPLS peut être appliqué en pratique ?

Exercice 8

Délai de paquetisation

On suppose qu'un message de données est découpé en paquet de longueur L. Ce message représente un volume total de B bits. Les paquets sont transmis en séquence et constitue une rafale d'une taille de B bits. Cette rafale de paquets emprunte un chemin de m liens de même capacité notée C exprimée en bit/s. On supposera le chemin vide de tout trafic. On négligera les temps de propagation et les erreurs de transmission. On pose T: le temps total de l'acheminement du message pour atteindre la destination ou temps de transfert.

1/ Donner l'expression de T.C (produit délai * bande passante) qui indique la latence binaire du chemin. La latence binaire exprime le nombre de bits pouvant être transmis pendant un temps équivalent à la remise complète d'un message à la destination. Le produit T.C sera exprimé en fonction de m, B et L. Expliquer votre raisonnement.

2/ De l'expression de T.C en déduire l'expression du délai T. Indiquer ce que représente chaque terme de cette expression.

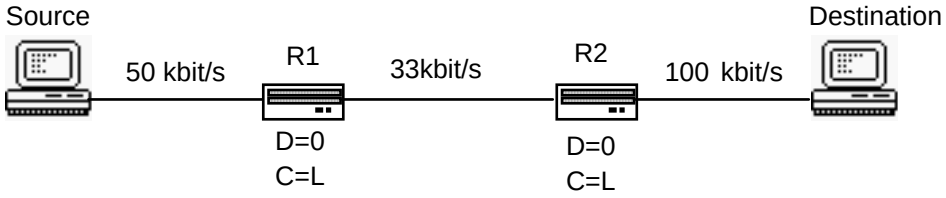
3/ Une réservation R est faite pour ce flot à l'aide d'un ordonnanceur WFQ dans les noeuds et la source. Donner l'expression du délai de transfert maximum.

4/ Pour ce flot, donner le délai de transfert minimum. Que faire pour diminuer le temps de transfert ?

5/ Le chemin n'est plus vide de trafic et la taille maximum des paquets dans ce réseau est de L. Quel est l'impact des autres trafics ("*cross traffics*") sur le temps de transfert de ce flot ? Dans ces conditions, donner l'expression du délai de transfert maximum pour ce flot ?

Exercice 9

Une source produit un flux CBR à un débit de 30 kbit/s (r=p) avec une taille de paquet de 1000 bits. Le délai acceptable pour le récepteur est de 80 ms. On supposera le réseau vide, ainsi chaque routeur introduit un délai égal à L (taille du paquet). Par hypothèse, les temps de propagations seront négligeables. Le chemin de la source au récepteur est le suivant :



1/ Représenter sur un graphe la courbe de l'enveloppe du trafic généré par la source ?

2/ Quel est le délai d'acheminement maximum pour une réservation au débit d'émission ?

3/ Quel est le débit minimum de réservation pour respecter la contrainte de délai ?

4/ Représenter sur le graphe de la question l'enveloppe de la réservation demandée ?

5/ La destination introduit une tolérance S de 30 ms (*Slack term*) pour sa réservation R. Donner la réservation demandée par le récepteur et faite dans chaque routeur ? Sachant que la relation (1) doit toujours être vérifiée:

$$S_{out} + \frac{b}{R_{out}} + \frac{C_{tot_i}}{R_{out}} \leq S_{in} + \frac{b}{R_{in}} + \frac{C_{tot_i}}{R_{in}} \qquad (1)$$

Avec C_{toti} le somme totale des termes d'erreur C pour tous les routeurs amont (vers la source) et incluant le routeur courant i. (R_{in}, S_{in}) est la réservation reçue et (R_{out}, S_{out}) représente la réservation demandée en amont par l'élement i. La formule (1) permet de calculer l'augmentation du délai de bout en bout quand $R_{out} < R_{in}$.

III- Conditionnement de trafic

Exercice 1

Donnez un exemple de discipline d'ordonnancement non conservative.

Exercice 2

(a) Expliquez comment le partage max-min (*max-min share*) est calculé.

(b) Considérons 10 flots avec des débits d'arrivée de 1, 2, …, 10 Mbps, qui traversent un lien à 45 Mbps. Calculer le fair share sur ce lien. Quel est le fair share si la capacité du lien est 60 Mbps ?

Exercice 3

Calculez l'allocation max-min fair pour les flots A, B, C, D et E, quand leurs demandes sont 2, 3, 4, 4, 5, leurs poids sont 2.5, 1, 0.5, 1, 2 et la taille des ressources est 15.

Exercice 4

a. Les token buckets et les leaky buckets sont deux manières de limiter le débit des données. Expliquer en quoi ils sont différents en terme de capacité à accepter les rafales dans le réseau, et garantir un débit de données moyen.

b. Le conditionnement de trafic peut être réalisé par connexion ou par hôte ; donner un exemple d'utilisation de chacune des deux manières.

Exercice 5

a. Considérons une source S dont le chemin vers la destination D est S-R-D, où R est un routeur. La courbe d'arrivée de S au routeur R est spécifiée par un token bucket de paramètres

(b = 500 Kb; r = 20 Kbps;R = 200 Kbps), où b représente la profondeur du bucket, r le débit moyen et R le débit crête.

Supposons que la source demande un délai d'au maximum 10s. Supposons également que le routeur alloue et garantit un débit fixe pour chaque flot (courbe de service linéaire). Quel est le débit minimum que le routeur doit allouer pour satisfaire cette contrainte de délai ? Quelle taille de buffer doit être allouée pour ce flot à ce débit ?

b. Supposons que dans la question a), le routeur est limité à un buffer de 300 Kb pour ce flot. Quel est le débit minimal que le routeur doit choisir pour satisfaire la contrainte de délai du flot aussi bien que sa contrainte de buffer ?

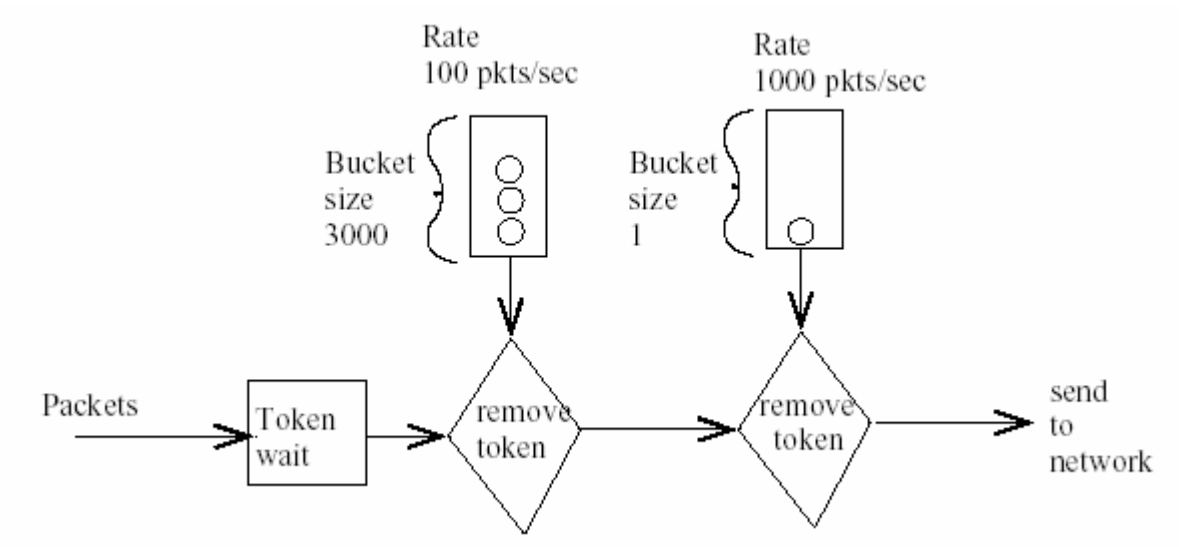
Exercice 6

Considérons un leaky bucket policer qui "police" le débit moyen et la taille des bursts d'un flot de paquets. On veut à présent policer aussi le débit crête, p. Montrez comment la sortie de ce leaky bucket policer peut être donné en entrée à un deuxième leaky bucket policer de façon à ce que les deux leaky buckets en série policent le débit moyen, le débit crête et la taille des bursts. Donnez la taille du bucket et le débit de génération des tokens pour le deuxième policer.

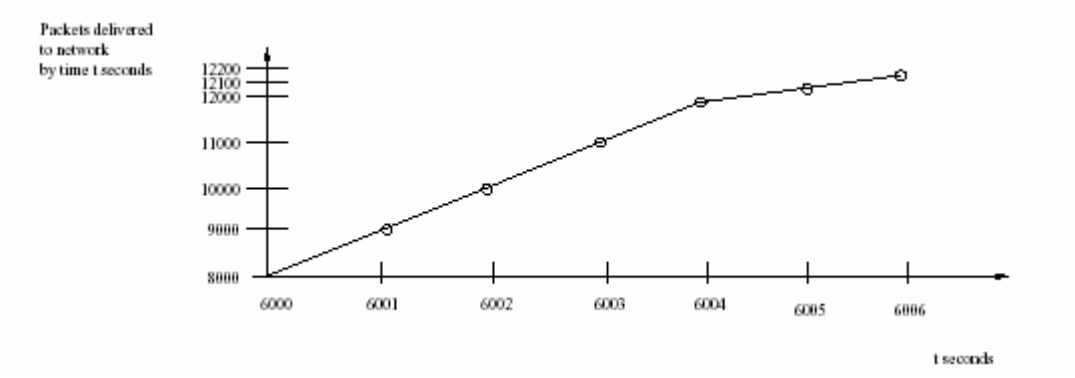
Exercice 7

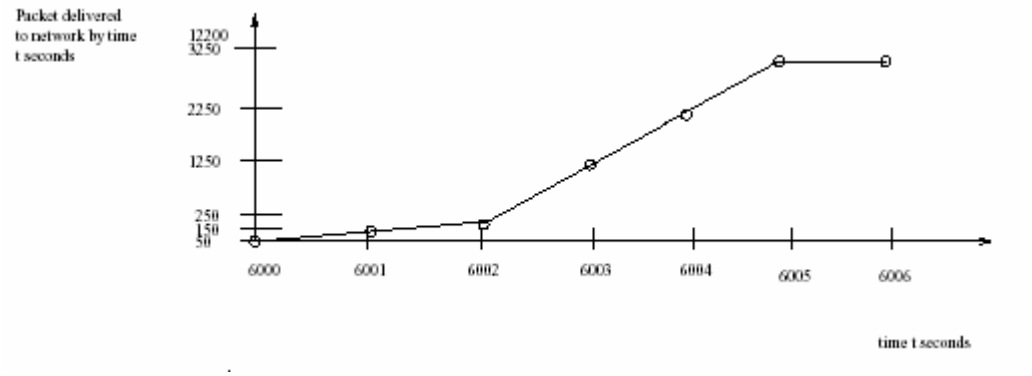
Leaky buckets

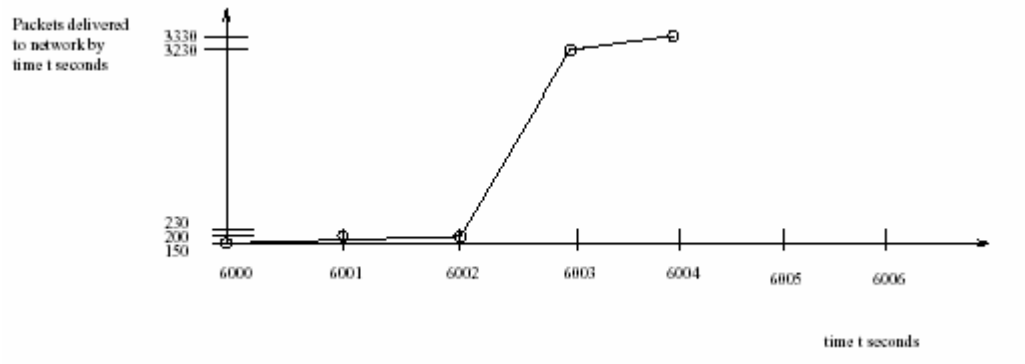
Considérons la figure suivante, montrant un flot passant à travers 2 leaky buckets avant de pénétrer dans le réseau.

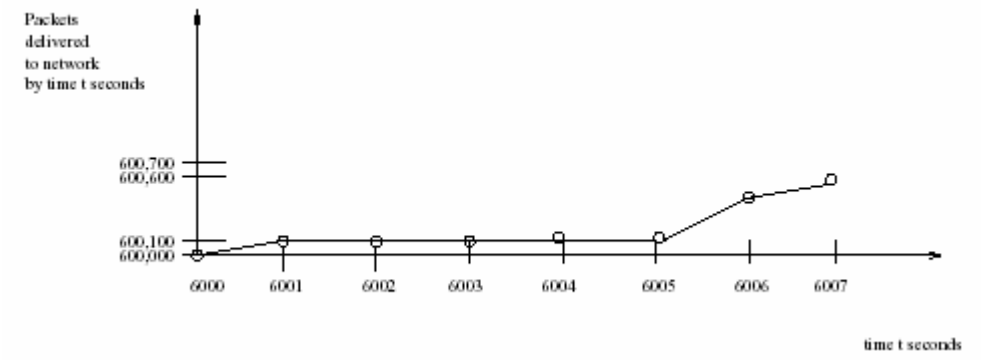


Observez les quatre comportements de flots suivants. Quels sont ceux qui auraient pu être générés par le tandem des leaky bucket policers ?









IV- Traitement des paquets

Exercice 1

Considérons un ordonnanceur qui ordonnance les paquets provenant de 4 flots. Ces 4 flots sont classés selon leur débit, avec $r_1 \leq r_2 \leq r_3 \leq r_4$, où r_i est le débit du ième flot. Le processus d'arrivée est poissonien. L'ordonnanceur est un ordonnanceur à priorités avec 4 niveaux de priorité (1 étant la priorité la plus élevée et 4 la plus faible).

Le tableau 1 spécifie la quantité de trafic de chaque flot pour chaque niveau de priorité. Par exemple, le trafic total du flot 2 de débit r2 est mis en file comme suit : une partie r_1 de son trafic est envoyée au niveau 1 et le reste au niveau 2.

Quelle politique d'ordonnancement cet ordonnanceur implémente-t-il ?

Table 1: Table for Problem 3

Priority Level	Flow 1	Flow 2	Flow 3	Flow 4
1	r_1	r_1	r_1	r_1
2	0	$r_2 - r_1$	$r_2 - r_1$	$r_2 - r_1$
3	0	0	$r_3 - r_2$	$r_3 - r_2$
4	0	0	0	$r_4 - r_3$

Exercice 2

Considérons l'ordonnanceur hiérarchique de la figure 1. Les valeurs des liens représentent la proportion de bande passante pour chaque lien. Les lignes pointant sur les feuilles représentent les liens arrivant aux feuilles et leur débit de trafic entrant (en Mbps) apparait en dessous. Quel débit est alloué aux flots d'origine A, B, C, F, G par l'ordonnanceur ?

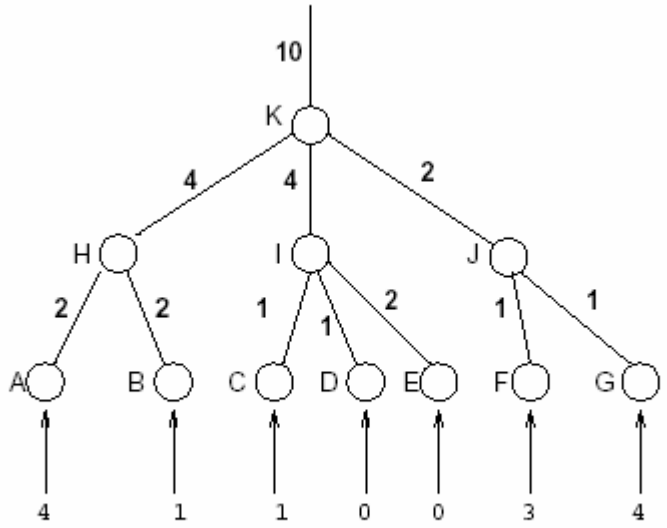


Figure 1: Hierarchical Fair Queueing Scheduler for Problem (5)

Exercice 3

RED

(a)

(a-i) si le trafic est constitué seulement de sources UDP, est-il probable que l'utilisation de RED résulte en une allocation de bande passante équitable (fair) entre les flots ?

(a-ii) Est-ce que RED évite les rafales de pertes ? Pourquoi ?

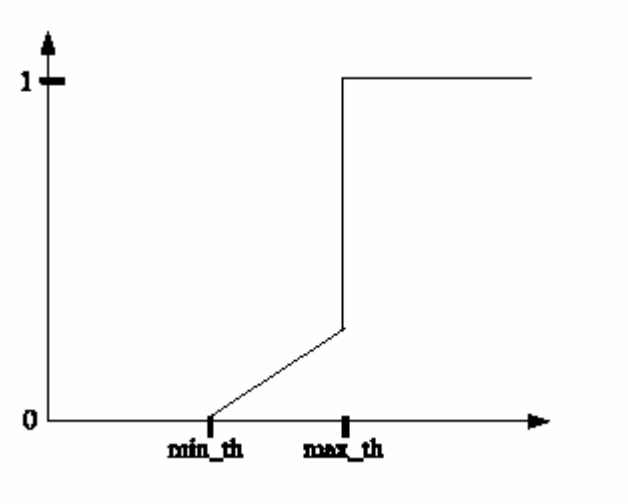
(b) Décrivez comment implémenter le contrôle de congestion avec Explicit Congestion Notification (ECN). En particulier,

(b-i) Quelles modifications (éventuelles) sont nécessaires dans RED ?

(b-ii) Quelles modifications (éventuelles) sont nécessaires dans la source TCP ?

(b-iii) Quelles modifications (éventuelles) sont nécessaires dans le récepteur TCP ?

(c) L'objectif de cette question est de décrire comment vous configureriez RED pour chacun des scénarios ci-dessous. Pour chaque scénario, vous pouvez changer seulement l'UN des paramètres de RED (min th, max th, ou le gain du filtre utilisé dans Exponentially-weighted Moving Average (EWMA) pour calculer la longueur moyenne de la file d'attente. Vous ne pouvez pas modifier l'algorithme RED lui-même. Expliquez bien, dans chaque cas, la motivation de votre changement de paramètre et en quoi ce changement améliore les performances.



(c-i) comment configureriez-vous RED pour avoir une meilleure utilisation du lien ?

(c-ii) comment configureriez-vous RED pour le rendre plus réactif à des rafales de courte durée ?

(c-iii) Comment feriez-vous pour que RED se comporte autant que possible comme le Drop-Tail queueing ?

Exercice 4

(a) Donnez deux avantages de RED sur le Drop Tail.

(b) Donnez un avantage de RED par rapport au Round-Robin et un avantage de Round-Robin par rapport à RED.

Exercice 5

(a) Décrivez le modèle de Service Assuré (assured service).

(b) Décrivez comment le mécanisme de gestion des buffers RIO (Red with In and Out) fonctionne, et comment il peut être utilisé pour implémenter le modèle de service assuré.

Exercice 6

Qu'est-ce qu'une file d'attente préemptive / non-préemptive ? Les files d'attente préemptives ont-elles un sens pour les réseaux informatiques ?

Exercice 7

Supposons que la politique d'ordonnancement WFQ est appliquée à un buffer qui supporte 3 classes ; supposons que les poids sont respectivement 0.5, 0.25 et 0.25 pour les trois classes.

a) Supposons que chaque classe a un grand nombre de paquets dans le buffer. Dans quel ordre les 3 classes pourraient-elles être servies pour avoir les poids WFQ ? (Pour l'ordonnancement Round-robin, un ordre naturel est 123123123…).

b) Supposons que les clases 1 et 3 ont un grand nombre de paquets dans le buffer, et qu'il n'y a pas de paquets de classe 2 dans le buffer. Dans quel ordre les 3 classes pourraient-elles être servies pour avoir les poids WFQ ?

Exercice 8

Montrez que tant que $r_1 < R \cdot w_i/(\Sigma w_j)$, alors d_{max} est le délai maximal que tout paquet du flot 1 peut subir dans la file WFQ.

Exercice 9

1) Laquelle de ces propositions est juste ?

a) RED réduit les risques de pertes successives d'une connexion.

b) RED modifie le mécanisme de retransmission de TCP.

c) RED modifie l'algorithme d'ajustement de fenêtre de TCP.

d) RED réduit le biais de TCP en faveur des connexions avec un petit round-trip time.

2) Quelles sont les définitions de TH_{min} et TH_{max} ?

3) Dans l'algorithme RED, un certain nombre de paramètres doit être fixé.

a) Quelle est la relation entre les valeurs choisies pour TH_{min} et le degré de sporadicité du trafic ?

b) Quelle est la relation entre la valeur ($TH_{max} - TH_{min}$) et le RTT typique vu par TCP ?

Exercice 10

Fair Queueing, Weighted Fair Queueing

Considérons un routeur qui gère 3 flots, sur lequel les paquets arrivent aux instants suivants :

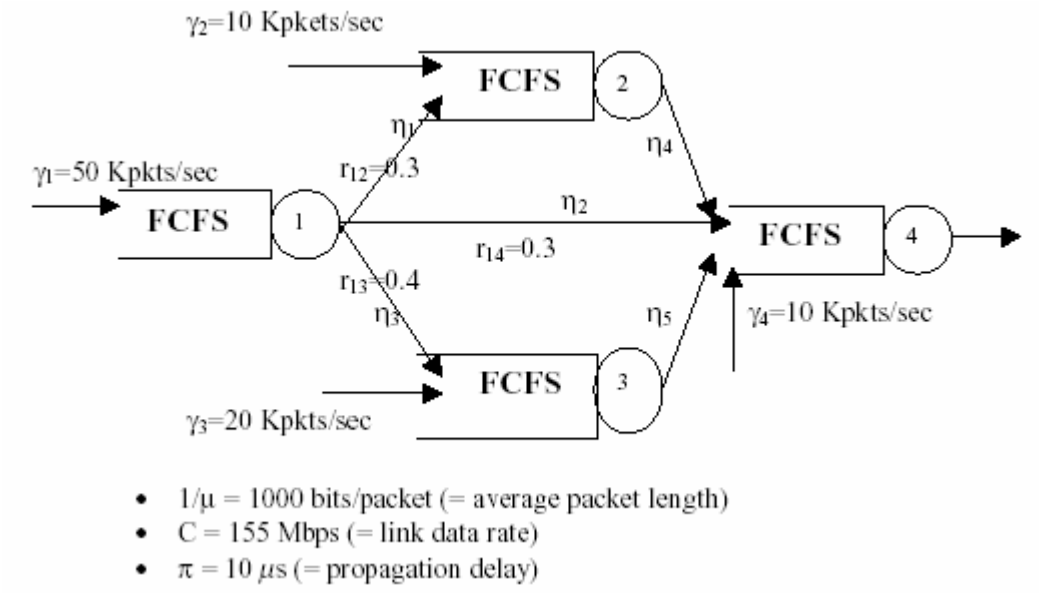
Flow A	1	3	4	5	6	7	8
Flow B	1	2	4	6	8	11	
Flow C	2	4	7	10			

(a) Supposons que le routeur implémente le fair queueing. Indiquez la séquence des paquets en sortie du routeur en fonction du temps. On suppose que le routeur dispose d'un buffer infini.

(b) Supposons que le routeur implémente le weighted fair queueing, où l'on donne aux flots A et B une part égale de la capacité, et où l'on donne au flot C deux fois la capacité de A. Indiquez la séquence des paquets en sortie du routeur en fonction du temps. On suppose que le routeur dispose d'un buffer infini.

Exercice 11

Réseaux de files d'attente



- $1/\mu$ = 1000 bits/packet (= average packet length)
- C = 155 Mbps (= link data rate)
- π = 10 μs (= propagation delay)

a) Soit r_{ij} la probabilité qu'un paquet quittant le noeud i soit destiné au noeud j. Calculez η_1,

η_2, η_3, η_4, η_5 en Kpkts/sec. Calculez également le taux d'arrivée moyen λ_i pour chaque noeud i.

b) Calculez T = délai moyen de bout en bout.

Exercice 12

QoS – ordonnancement

Le Fair queueing est différent du round-robin ; s'il y a n files d'attente, le round-robin revient à servir les files d'attente chacune leur tour, tant qu'il y a des paquets en attente. Les paquets peuvent être de taille différente. Si l'on a deux flots, que la taille des paquets d'un flot est de 100 et que la taille des paquets de l'autre flot est de 50, alors le round-robin donnera deux fois plus de bande passante au flot ayant les plus gros paquets.

Le fair queuing tient compte en plus de la taille des paquets quand il ordonnance les transmissions. Supposons qu'il y ait une file d'attente pour chaque flot. Au moment de l'arrivée de chaque paquet, l'instant de départ virtuel du paquet est déterminé. Ceci est déterminé en mettant le nouveau paquet à la fin de sa file d'attente. Supposons alors que le lien partagé goulot d'étranglement puisse servir toutes les files simultanément (i.e. en une unité de temps, il transmet la même quantité pour chaque file) et détermine l'instant de fin pour le nouveau paquet arrivé. Ainsi, chaque paquet dans le système a un instant virtuel de fin. L'ordonnancement fair queuing revient à servir les paquets dans l'ordre de leur instant virtuel de fin.

Par exemple, considérons 2 files d'attente. Trois paquets arrivent à peu près en même temps, mais dans l'ordre suivant :

a) le paquet 1 de taille 500 arrive sur la file 1
b) le paquet 2 de taille 200 arrive sur la file 1
c) le paquet 3 de taille 1000 arrive sur la file 2

L'ordonnancement fair queueing serait (a), (b) puis (c). Ceci est différent du round-robin, parce que l'instant virtuel de fin pour (b) est antérieur à celui de (c). Si l'ordre d'arrivée avait été (c), (a) et (b), alors l'ordre de service aurait aussi été (c), (a) et (b), puisqu'il n'y a pas de pré-emption.

Supposons qu'un routeur ait trois flots en entrée et un en sortie. Il reçoit les paquets listés dans le tableau ci-dessous à peu près tous au même instant, dans l'ordre indiqué, pendant une période où le port de sortie est occupé mais toutes les files sont autrement vides.

Packet	Size	Flow
1	100	1
2	100	1
3	100	1
4	100	1
5	190	2
6	200	2
7	110	3
8	50	3

Donnez l'ordre dans lequel les paquets sont transmis, en supposant que l'on utilise :

a) le fair queueing

b) le weighted fair queueing, avec le flot 2 ayant un poids de 2 et les deux autres avec un poids de 1.

Exercice 13

Un paquet de longueur 100 et 200 bits des flots A et B respectivement arrivent à l'instant t=0 à un ordonnanceur Fair Queuing vide. La bande passante du lien est de 100 bits/s.

1/ A quel instant (réel) se termine la transmission de chaque paquet ?
2/ Quel est le nombre de cycles R(t) quand le paquet de A termine sa transmission ?
3/ Si maintenant, un paquet du flot A avec une longueur de 10 bits arrive à t=1,5. Quelle sera son estampille (ou nombre de fin) notée F(A,1)? Quand le flot A devient inactif selon FQ ?
4/ Représenter en fonction du temps et sur un même graphe, les estampilles F(i,k) (k ième paquet du ième flot) des 3 paquets reçus, l'évolution de R(t) et la transmission des paquets.

TD Routage inter-domaine

Exercice 1 : rappel OSPF

a. Présentez rapidement OSPF et ses objectifs.

b. Comment l'initialisation des réseaux et les mises à jour des routes sont-elles effectuées dans OSPF ?

Exercice 2

Vrai ou faux ?

a) RIP est un protocole de routage à vecteurs de distances
b) BGP est un protocole « exterior gateway »
c) RIP utilise UDP pour échanger les annonces RIP(réponses)
d) Dans OSPF, un routeur de bordure de domaine doit être un routeur du backbone
e) BGP est un protocole à état des liens

Exercice 3 : BGP

a. Présentez rapidement BGP et ses objectifs.

b. Comment la découverte des voisins, l'acquisition d'informations de routage et l'évitement de boucles de routage sont-ils gérés dans BGP ?

Exercice 4

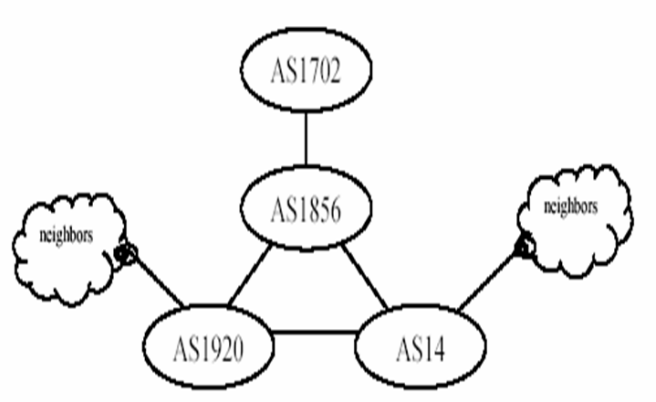
Le routage *Border Gateway Protocol (BGP)* est utilisé pour déterminer le meilleur chemin entre une source et une destination qui ne sont pas dans le même système autonome (*autonomous system, AS)*, c'est-à-dire pas dans le même domaine administratif. Etant donné que les chemins traversent souvent des réseaux gérés par des entreprises différentes, le choix des routes est souvent déterminé par une combinaison de facteurs, incluant éventuellement des préférences locales spécifiques au niveau de chaque routeur, des relations commerciales avec les voisins, et la longueur du chemin. BGP combine ces politiques de routage complexes qui ne sont pas uniformes à travers l'Internet.

Comme BGP effectue du routage *inter-domaine* (*i.e.*, entre différents domaines), il cherche avant tout à atteindre un AS contenant la destination, et non à atteindre l'hôte lui-même. Une fois que les paquets sont routés vers l'AS contenant la destination, leur remise à l'hôte spécifique est laissée au routage *intra-domaine* (*i.e.*, à l'intérieur d'un domaine). Par exemple, pour atteindre l'hôte 192.1.12.14 depuis un AS distant, il suffit de savoir que
1) cet hôte est contenu dans un AS, par exemple n°1702
2) l'AS distant connaît la route vers AS1702.
Les routeurs à l'intérieur de l'AS1702 peuvent alors router le paquet vers l'hôte 192.1.12.14.

Pour comprendre le routage inter-domaine, pensez à un AS comme à un nœud dans un réseau. Les nœuds qui lui sont connectés sont ses *voisins*. Les routeurs des ASs voisins sont connectés physiquement, ainsi les paquets peuvent passer de l'un à l'autre.

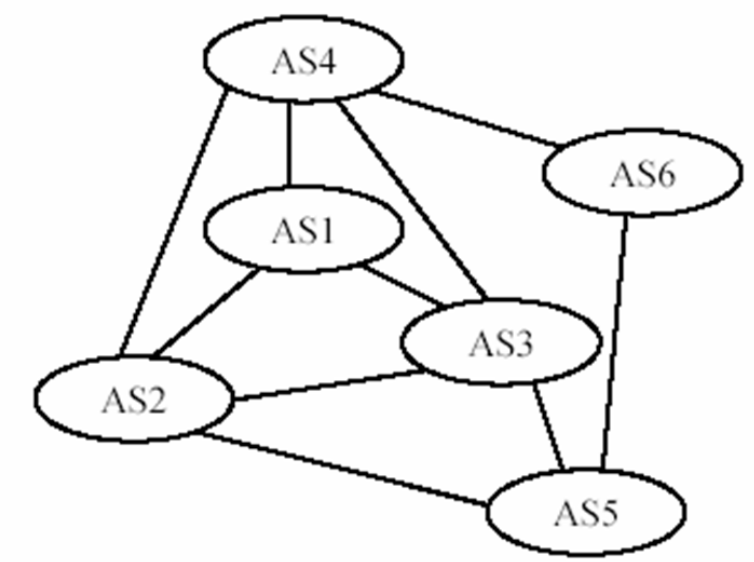
Voici comment les informations concernant les hôtes est transmise à travers le réseau :
1. Un AS "annonce" à ses voisins les hôtes qu'il contient.
2. Quand un AS reçoit de ses voisins des informations à propos d'une destination, il regarde le « meilleur » chemin vers cette destination. S'il n'y a pas de chemin vers cette destination ou si le nouveau chemin est meilleur, il enregistre ce nouveau chemin comme « meilleur » chemin.
3. Décider du "meilleur" chemin implique de prendre en compte de nombreux facteurs. Cette décision est prise différemment par chaque AS.
4. A chaque fois que le "meilleur" chemin d'un AS change à cause d'une mise à jour, il signale ce changement à ses voisins si ses politiques de routage le permettent.

Considérons l'exemple suivant :



1. l'AS1702 annonce les destinations 192.1.12.0 par 192.1.12.255.
2. son voisin AS1856 reçoit cette information pour la première fois et met à jour sa table pour indiquer qu'il a une connexion directe vers AS1702. Il annonce ensuite cette information à ses voisins, AS1920, AS14, et AS1702. (AS1702 a déjà un chemin vers lui-même ; alors il ignore cette nouvelle information.)
3. AS1920 et AS14 mettent à jour leur table pour tenir compte du chemin vers AS1856 et donc AS1702. Ces ASs annoncent ce chemin à leurs voisins.
4. Comme AS14 et AS1920 sont connectés, AS14 reçoit un chemin par AS1920 en plus de celui qu'il a déjà reçu de AS1856. Supposons que ses politiques lui disent de préférer les chemins venant de AS1920. Il changera alors sa table pour mettre le chemin AS14, AS1920, AS1856, AS1720 et annoncera cela à ses voisins (Notez que les boucles ne sont pas permises ; alors, AS1856 ne peut pas choisir le chemin AS1856, AS14, AS1920, AS1856, AS1702.)
5. Ce processus continuer tant que les ASs à travers le réseau choisissent leur meilleur chemin vers l'AS1702. Notez que les routes peuvent continuer à changer tant que de nouvelles informations sont reçues des ASs voisins.

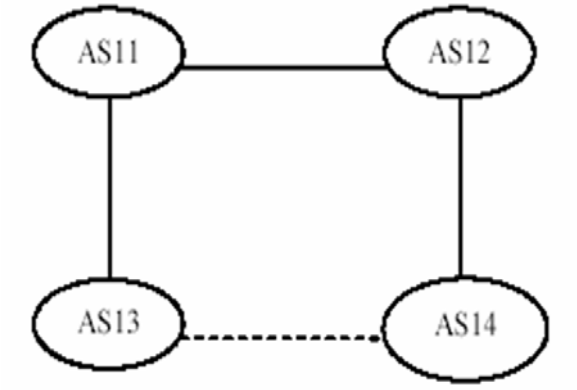
I. Considérons à présent le graphe d'ASs suivant, et considérons les routes vers une destination dans AS1. Voici les politiques de routage pour chaque AS (notez que le plus court chemin représente ici le chemin via le plus petit nombre possible d'ASs) :



1. AS2 n'utilisera pas le lien direct vers AS1 à moins que cela soit absolument nécessaire. Il préfère les chemins de l'AS4, puis ceux de l'AS3, puis ceux de l'AS5.
2. AS3 n'utilisera pas le lien direct vers AS1 à moins que cela soit absolument nécessaire. Il rejette tous les chemins via AS4.
3. AS4 choisit toujours le plus court chemin disponible.
4. AS5 préfère toujours les chemins par l'AS6 et sinon choisit le plus court chemin disponible.
5. AS6 choisit toujours le plus court chemin disponible.

A. Supposons que le processus commence avec AS1 annonçant ses destinations. A un certain point, les ASs auront des chemins vers l'AS1 qui ne changeront plus (*i.e.,* le graphe ci-dessus doit converger vers une solution). Trouvez ces « meilleurs » chemins vers AS1 pour chaque AS du réseau.

II. Les relations commerciales déterminent souvent les politiques de routage. On peut modéliser l'Internet en utilisant 2 sortes de relations : les *relations client-provider* et les *relations pair-à-pair*.
Dans une relation client-provider, un AS, le client, achète un service auprès d'un autre AS, le provider. Le client paye généralement le provider pour transporter du trafic et fournir l'accès vers les autres réseaux, tandis que le provider annonce sa capacité à atteindre le client, de telle sorte que ce client soit accessible par le réseau.
Dans une relation pair-à-pair, deux ASs établissent un accord mutuel pour transporter le trafic de l'autre et partager les routes, fournissant souvent des raccourcis à des routes plus longues devant passer par un seul provider.
Ces relations affectent les politiques de routage de la manière suivante : les ASs annonceront à leurs voisins toutes les routes vers leurs clients (c'est en partie ce pour quoi les client payent). D'un autre côté, les ASs n'annonceront à aucun voisin autre qu'un client les routes via un pair ou un provider. De plus, les routes via un client sont préférées, puis les routes via les pairs et finalement les routes via les providers.
Dans le graphe ci-dessous, supposons que AS11 et AS12 sont pairs, que AS13 est client de AS11 et que AS14 est client de AS12. Supposons que l'AS destination soit AS14.



B. Supposons que le lien entre AS13 et AS14 ne soit pas présent et que AS14 s'annonce auprès de ses voisins, et ainsi de suite. Quels sont les chemins de niveau AS de chaque AS vers AS 14 ?
Supposons maintenant que AS14 n'est pas sûr de son lien vers AS12 et décide d'acheter un lien auprès de AS13, *i.e.*, il devient client de AS13. Cependant, il dit à AS13 (par un accord spécial) que ce lien est un lien de backup et qu'il doit avoir une préférence inférieure à tout autre chemin que AS13 pourrait avoir vers AS14. Ainsi il n'y a pas de changement aux chemins AS de la question B, parce que cela est pris en compte quand AS13 annonce une « meilleure route » vers AS14.

C. Quand un lien est défaillant, la route correspondante est enlevée et doit être remplacée par une autre. Supposons que le lien entre AS12 et AS14 soit indisponible ; AS12 retire sa route vers AS14. AS13 a un lien de backup qu'il annoncera tant qu'il n'y aura pas d'autre route vers AS14. Etant donné ce changement, quels sont les nouveaux chemins de niveau AS de chaque AS vers AS14, une fois que les routes se sont stabilisées ?

D. Supposons que le lien entre AS12 et AS14 est restauré. En gardant en tête les règles de préférence pour les relations commerciales, quels sont à présent les chemins d'AS de chaque AS vers AS14 ?

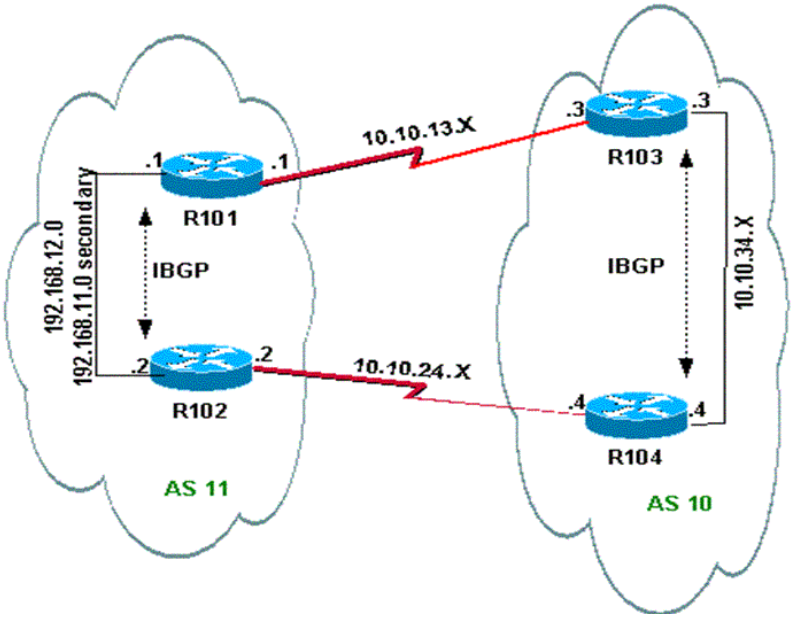
E. AS14 se plaint que l'AS13 reçoit encore du trafic par le lien de backup qui n'est plus censé être utilisé puisque d'autres routes sont disponibles. AS13 reboote ses routeurs, ce qui désactive temporairement le lien de AS13 vers AS14. Que sont les chemins AS après que AS13 redevient online ? Notez que AS13 devient reconnecté au réseau après que le chemin AS13–AS14 a été retiré.

Exercice 5 . Réseaux multihomés

Quels problèmes sont liés aux réseaux qui ont plusieurs connexions au monde extérieur ? Décrivez différents exemples et leur solution.

Exercice 6

On parle de Multi-homing lorsqu'un système autonome (AS) client a plusieurs liens BGP connectés à l'extérieur, comme le montre l'exemple suivant :



Supposons que l'AS 11 est le réseau client.

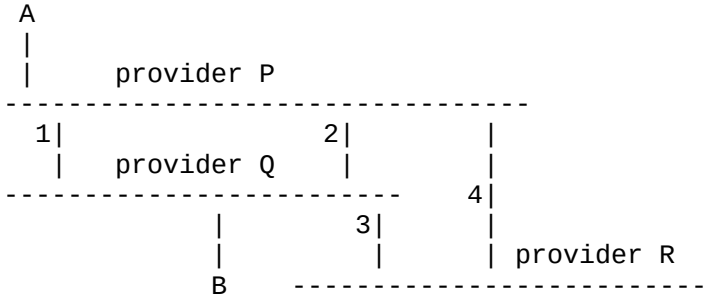
a) Expliquez comment vous configureriez BGP pour garder un lien de backup pour le trafic entrant et sortant.

b) En général, on considère comme du gâchis le fait de réserver un lien uniquement pour le backup. A la place, les deux liens sont configurés pour partager le trafic entrant et sortant. Expliquez comment vous configureriez BGP pour réaliser cela.

c) L'agrégation d'adresses est réalisée si un client obtient ses adresses réseau de la part de son provider. Par exemple, le provider a 138.39.0.0/16, et donne 138.39.1.0/24 à son client. Si l'AS client est multi-homé avec deux providers différents, comment l'agrégation d'adresses peut-elle affecter la capacité à faire du partage de charge ?

Exercice 7

Considérez le réseau ci-dessous, dans lequel les lignes horizontales représentent les liens des providers de transit et les lignes verticales numérotées représentent les liens inter-providers.



(a) Combien de routes vers P les speakers BGP de Q peuvent-ils recevoir ?

(b) Supposons que Q et P adoptent la politique consistant à router le trafic sortant (outbound) vers le lien le plus proche du provider de destination, minimisant ainsi leur propre coût. Quel chemin prendra le trafic de l'hôte A vers l'hôte B et de B vers A ?

(c) Que devrait faire Q pour que le trafic B -> A utilise le lien 1?

(d) Que devrait faire Q pour que le trafic B -> A passe par R?

Exercice 8

(a) Considérons un chemin P de niveau AS entre deux hôtes terminaux dans l'Internet (un chemin de niveau AS est une liste d'ASs sur le chemin entre les deux terminaux). Imaginons un lien entre deux ASs consécutifs sur le chemin. Un tel lien (A1 ;A2) est appelé
(a) un lien client-provider, si l'AS A1 est client de A2
(b) un lien provider-client si l'AS A2 est client de A1
(c) un lien pair-à-pair si les deux ASs A1 et A2 sont pairs l'un de l'autre.

Expliquez brièvement s'il est possible d'avoir plus d'un :
(i) lien client-provider sur ce chemin P ?
(ii) un lien provider-client sur ce chemin P ?
(iii) un lien pair-à-pair sur ce chemin P ?

(b) Considérons une organisation AS1 cliente de 2 providers de service Internet AS2 et AS3. AS2 a une relation de peering avec l'AS4, est client de l'AS5 et est provider de l'AS6. Supposons qu'il s'agisse des seules relations impliquant l'AS en question. Répondez aux questions suivantes :
(i) Est-ce que AS1 exporte les routes obtenues de AS2 vers AS3 ?
(ii) Est-ce que AS2 exporte les routes obtenues de AS5 vers AS1 ?
(iii) Est-ce que AS2 exporte les routes obtenues de AS5 vers AS4 ?
(iv) Est-ce que AS2 exporte les routes obtenues de AS6 vers AS1 ?
(v) Est-ce que AS2 exporte les routes obtenues de AS1 vers AS4 ?

Exercice 9

Considérons un réseau unique (i.e. un unique AS) dans lequel le routage par plus court chemin de Dijkstra est utilisé entre les terminaux. Supposons maintenant que le routage overlay (concernant seulement les terminaux) est utilisé pour router entre deux terminaux.

Avec cette métrique de performance, le routage overlay peut-il être plus performant que le routage au seul niveau réseau ? Justifiez votre réponse. En faisant toutes les hypothèses que vous voulez, quel est le surdébit additionnel entraîné par le routage overlay si les chemins sont deux sauts (logiques) dans le réseau overlay ? Etant données les réponses aux questions précédentes, identifiez un scénario dans lequel le routage overlay serait encore souhaitable dans un scénario à AS unique.

Considérons maintenant le cas de l'Internet, où à la fois le routage intra et inter-domaine sont utilisés, et où la métrique de performance est le nombre de sauts. Donnez un exemple

montrant comment le routage overlay peut être plus performant que le routage au niveau réseau uniquement.

Exercice 10

Supposez que vous conceviez un nouveau protocole de routage par la source. Une décision de conception importante est la manière dont les routeurs s'échangent les informations de routage à partir desquelles les décisions de routage seront prises.

Vous avez deux possibilités :

1. les routeurs séchangent leurs tables de routage comme cela est fait dans les annonces RIP.
2. Les routeurs s'échangent les mises à jour de l'état des liens comme cela est fait dans OSPF.

Quelle possibilité choisiriez-vous et pourquoi ?

Exercice 11

Il existe plusieurs protocoles pour le routage inter-AS et intra-AS. Une raison fréquemment donnée pour ne pas utiliser OSPF pour le routage inter-domaine est qu'il ne passe pas à l'échelle.

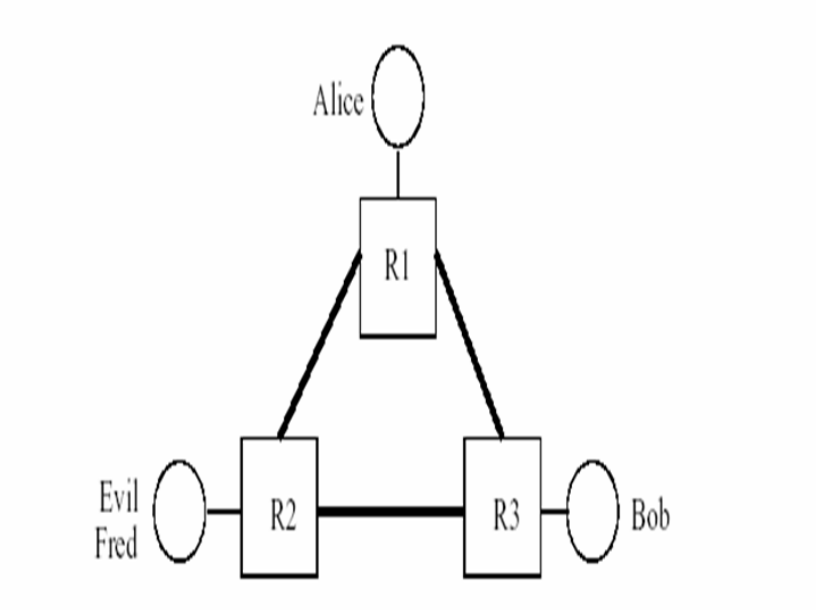
En supposant que vous puissiez éliminer complètement les problèmes de passage à l'échelle d'OSPF en faisant du routage hiérarchique (en utilisant des domaines), y a-t-il encore des raisons d'utiliser BGP plutôt qu'OSPF pour le routage inter-AS ?

Exercice 12

Supposons que deux ASs X et Y sont « ennemis ».
En utilisant BGP, est-il possible pour l'AS X d'implémenter la politique : « le trafic venant de mon AS ne doit pas traverser l'AS Y » ? Pourquoi ?
Est-il possible pour l'AS Y d'implémenter la politique : « Je ne veux pas transporter de trafic en transit depuis X » ? Pourquoi ?

Exercice 13

Considérez le réseau ci-dessous. Supposons que les routeurs utilisent le routage par état des liens à plus court chemin et que les tables de routage sont stabilisées. Le « méchant » Fred décide d'usurper Bob et envoie à Alice des paquets IP avec l'adresse de Bob comme adresse source. Comment le routeur d'Alice (routeur 1) peut-il détecter cela ?

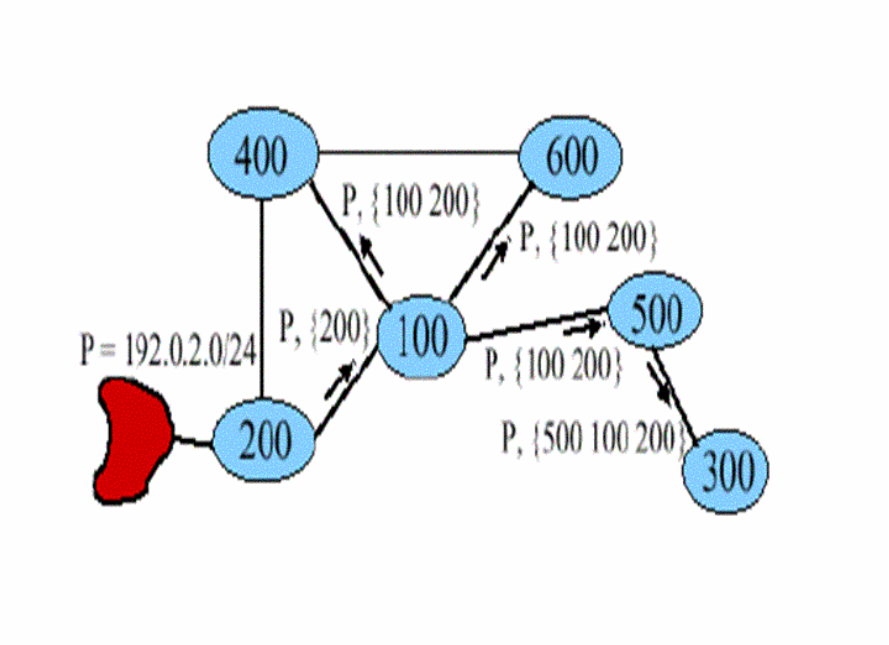


La réponse la question précédente est-elle encore vraie dans le cas d'un protocole à vecteurs de distance ?

Si Alice se trouve maintenant sur un autre réseau et que R1 est son routeur de bordure, peut-on utiliser une solution similaire ?

Exercice 14

La figure ci-dessous donne un exemple de la manière dont les routes vers un AS particulier peuvent être exportées. Les nœuds du graphe représentent les différents ASs et l'annotation sur l'arc orienté P,{i, j} montre l'annonce que l'AS i envoie à l'AS j concernant ses informations d'atteignabilité vers P (BGP est un protocole de vecteurs de chemins d'AS et chaque annonce doit contenir le chemin de niveau AS et le préfixe, dans notre exemple, l'AS P). BGP est un protocole reposant sur les politiques et un AS exporte une route vers un voisin seulement s'il le veut bien, et envoie le trafic pour ce préfixe depuis ce voisin.



Un tel graphe nous donne une idée sur les relations entre les ASs. A partir de ce graphe,

1. Que pouvez-vous déduire de la relation entre l'AS200 et l'AS 100 ?
2. Que pouvez-vous déduire de la relation entre l'AS200 et l'AS400 ?

Exercice 15

a. Qu'est-ce que le routage à base de politiques ?

b. Quels avantages offre-t-il ?

c. Quels problèmes peuvent y être associés ?

Exercice 16

Routage externe à base de politiques

a. Le routage à base de politiques est-il possible entre opérateurs, dans le routage externe ? Quels sont les problèmes liés à cela ?
b. Quel type de support BGP offre-t-il pour le routage à base de politiques ?
c. Quelles sont les possibilités et les limitations de routage externe à base de politiques ?

Exercice 17

Un attribut BGP est transitif s'il est passé aux autres pairs BGP, intransitif sinon.

a) MED n'est pas transitif. Pourquoi ?

b) MED n'est en général pris en considération que s'il s'agit d'une mise à jour venant d'un client. Pourquoi ?

Considérez les routes BGP suivantes :
Entrée 1: ASPATH {1,3}, MED 100, internal, IGP metric to NEXT_HOP 10
Entrée 2: ASPATH {2,3}, MED 150, internal, IGP metric to NEXT_HOP 5
Entrée 3: ASPATH {1,3}, MED 200, external

L'ordre dans lequel les routes BGP ont été reçues est entrée 1, entrée 2 et entrée 3 (l'entrée 3 est la plus ancienne dans la table BGP et l'entrée 1 est la plus récente). « Internal » signifie que la route est apprise par iBGP, et "external" signifie qu'elle est apprise de eBGP.

c) Quelle route un routeur BGP avec MED choisirait-il ? Pourquoi ?

d) Quelle route un routeur BGP avec MED choisirait-il ? Pourquoi ?

Exercice 18

Donnez un exemple de configuration de routeurs regroupés dans des systèmes autonomes de telle sorte que le chemin avec le moins de sauts entre A et B traverse 2 fois le même AS. Expliquez comment BGP réagirait dans une telle situation.

Exercice 19

Soit A le nombre de systèmes autonomes de l'Internet et soit D (diamètre) la longueur maximale d'un chemin d'AS.

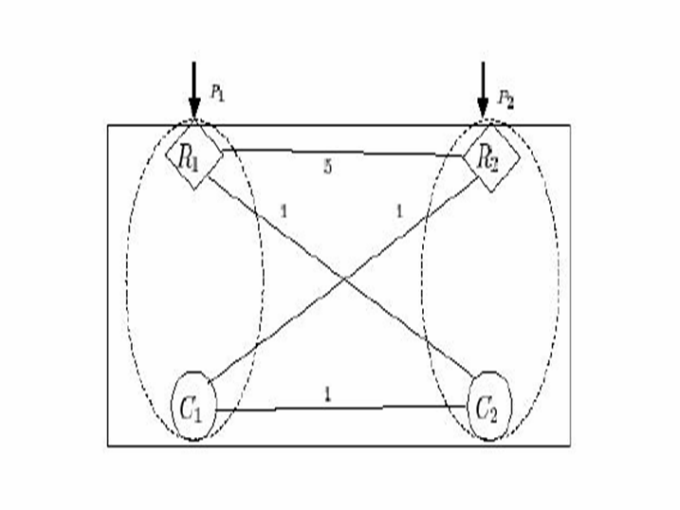
a) Donnez un modèle de connectivité pour lequel D est de l'ordre de logA et un autre pour lequel D est de l'ordre de racine carrée de A.
b) En supposant que chaque numéro d'AS est sur 2 octets et que chaque numéro de réseau est sur 4 octets, donnez une estimation de la quantité de données qu'un speaker BGP doit recevoir pour garder une trace de chaque chemin d'AS vers chaque réseau. Exprimez votre réponse en fonction de A, D et du nombre N de réseaux.

Exercice 20

I-BGP est utilisé pour échanger des informations d'atteignabilité externe avec un seul AS. Mais, à l'inverse des sessions E-BGP qui partagent un lien physique, les messages I-BGP circulent sur un lien logique. Par essence, les paquets I-BGP sont routés sur IGP ! Supposons que chaque routeur dans l'AS ait besoin d'une table de routage sans défaut. Alors, chaque routeur de l'AS doit participer aux sessions I-BGP. Une manière de faire cela qui passe à l'échelle est d'avoir une hiérarchie. L'ensemble des routeurs est partitionné en des ensembles disjoints et il y a un *Route Reflector* pour chaque ensemble (on suppose qu'il s'agit de l'un des routeurs de l'ensemble). Le maintien de la cohérence (à propos des informations d'atteignabilité) entre tous les routeurs peut alors être décomposé en deux problèmes :

1) Les Route Reflectors ont besoin de maintenir la cohérence entre eux, ce qu'ils font typiquement en établissant des sessions I-BGP avec les autres Route Reflectors.
2) Chaque ensemble de routeurs doit assurer la cohérence entre ses routeurs membres. Ceci est typiquement réalisé par chaque routeur qui maintient une session I-BGP avec le Route Reflector de son ensemble.

Les Route Reflectors suppriment également les informations d'atteignabilité qu'ils jugent redondantes.Par exemple, s'ils apprennent qu'il y a deux liens vers un préfixe, l'un avec un coût de 5 et l'autre avec un coût de 10, ils filtrent le chemin de coût 10.



La figure montre un AS X qui a 4 routeurs R1, R2, C1 et C2. R1 et R2 sont des routeurs gateways et également des Route Reflectors. Imaginez qu'il y a un autre AS Y, adjacent à l'AS X, ayant deux gateways R3 et R4. R1 maintient une session E-BGP avec R3 et R2 maintient une session E-BGP avec R4. Par son interaction avec R3, R1 connaît P1 et un AS-path en un saut vers l'AS Y. De la même façon, R2 connaît P2 et un autre chemin d'AS en un saut vers l'AS Y. Chaque cercle représente un ensemble (R1 maintient 2 sessions I-BGP, l'une avec C1, l'autre avec R2. Idem pour R2). Les arcs du graphe montrent les liens réels entre les routeurs, et l'étiquette d'un arc représente le coût IGP du lien. Notez que le graphe a un axe de symétrie, par conséquent le raisonnement sur R1, P1 et C1 s'applique aussi à R2, P2 et C2.

1) Quel chemin, P1 ou P2, R1 annonce-t-il à C1 ?
2) Si C1 doit atteindre R1, quel chemin prendra-t-il ?
3) Supposons que C1 veuille joindre un hôte sur l'AS Y. Que se passe-t-il ?

LE PROTOCOLE IPV6

Le protocole IPv6 résulte de travaux entrepris en 1992 au sein de l'IETF (*Internet Engineering Task Force*), l'organisme de standardisation de l'Internet. Ces travaux ont principalement été motivés pour résoudre certains des problèmes révélés par l'utilisation à grande échelle d'IPv4 tels que l'épuisement des adresses disponibles ou l'explosion des tables de routage. Le protocole IP a subi un toilettage reprenant l'expérience acquise au fil des ans avec IPv4, mais sans pour autant renier les principes fondamentaux qui ont fait le succès de l'Internet tels que la communication de « bout en bout » et le « meilleur effort » (*Best Effort*) pour l'acheminement.

Parmi les nouveautés essentielles, on peut citer :

- l'augmentation de 2^{32} à 2^{128} du nombre d'adresses disponibles ;
- des mécanismes de configuration et de renumérotation automatique ;
- IPsec, QoS et le multicast « de série » ;
- la simplification des en-têtes de paquets, qui facilite notamment le routage.

Certains de ces points sont détaillés dans la suite de ce document présenté sous la forme de travaux dirigés (TD).

Après plus de 10 ans d'efforts de standardisation, les spécifications de base du protocole et les règles d'attribution des adresses sont clairement définies. La plupart des routeurs et des systèmes d'exploitation incluent cette nouvelle version du protocole IP et la transition vers IPv6 devient jour après jour réalité.

1 ADRESSES IPV6

1.1 Format des adresses

Une adresse IPv6 est longue de 128 bits (contre 32 pour IPv4). On dispose ainsi d'environ 3,4 × 1038 adresses, soit 340 282 366 920 938 463 463 374 607 431 768 211 456, soit encore, pour reprendre l'image usuelle, plus de 67 milliards de milliards par millimètre carré de surface terrestre.

1.2 Notation d'une adresse IPv6

On abandonne la notation décimale pointée employée pour les adresses IPv4 (par exemple 172.31.128.1) au profit d'une écriture hexadécimale, où les 8 groupes de 16 bits sont séparés par un signe deux-points :

```
1fff:0000:0a88:85a3:0000:0000:ac1f:8001
```

La notation canonique complète ci-dessus comprend exactement 39 caractères. Cependant, la notation décimale pointée est autorisée uniquement pour les 2 derniers blocs représentant les 32 derniers bits de l'adresse IPv6. Ainsi l'adresse ci-dessus est équivalente à :

```
1fff:0000:0a88:85a3:0000:0000:172.31.128.1
```

Il est permis d'omettre de 1 à 3 chiffres zéros non significatifs dans chaque groupe de 4 chiffres hexadécimaux. Ainsi, l'adresse IPv6 ci-dessus est équivalente à :

`1fff:0:a88:85a3:0:0:ac1f:8001 (1fff:~~000~~0:0a88:85a3:~~000~~:~~000~~0:ac1f:8001)`

ou

`1fff:0:a88:85a3:0:0:172.31.128.1`

Par contre, il n'est pas permis de supprimer un signe deux-points entre deux groupes de 1 à 4 chiffres hexadécimaux.

De plus, une unique suite de un ou plusieurs groupes consécutifs de 16 bits tous nuls peut être omise, en conservant toutefois les signes deux-points de chaque côté de la suite de chiffres omise. Ainsi, l'adresse IPv6 ci-dessus peut être abrégée en :

`1fff:0:a88:85a3::ac1f:8001`

ou

`1fff::a88:85a3:0:0:ac1f:8001`

voire aussi

`1fff:0:a88:85a3::172.31.128.1`

ou.

`1fff::a88:85a3:0:0:172.31.128.1`

En revanche les écritures suivantes NE SONT PAS valides

~~`1fff::a88:85a3::ac1f:8001`~~

~~`1fff::a88:85a3::172.31.128.1`~~

car elle contiennent chacune plusieurs substitutions (dont les longueurs binaires respectives sont ici ambiguës) : il ne peut exister qu'une seule occurrence de la séquence « `::` » dans la notation d'une adresse IPv6, qui contiendra nécessairement de 2 à 7 signes deux-points (ou 3 points séparateurs et de 2 à 6 signes deux-points).

L'adresse IPv6 indéterminée peut ainsi être abrégée en « `::0.0.0.0` » ou « `::` ».

1.2.1 Donnez une écriture pour l’adresse suivante :

`FEDC:0000:0000:0000:400:A987:6543:210F`

1.2.2 L'abréviation «::» peut-elle apparaître plus d’une fois dans une même adresse ? Pourquoi ?

La représentation des préfixes IPv6 est similaire à la notation CIDR (RFC 1519) utilisée pour les préfixes IPv4. Un préfixe IPv6 est donc représenté par la notation :

`adresse_ipv6/longueur_du_préfixe_en_bits`

Exemple de notation combinant l'adresse d'une interface et la longueur du préfixe réseau associé :

`3EDC:BA98:7654:3210:945:1321:ABA8:F4E2/64`

1.2.3 Donnez la forme abrégée de l’adresse suivante sachant que la longueur du préfixe est de 8 octets

```
3EDC:BA98:7654:3210:0000:0000:0000:0000
```

1.2.4 Proposez une autre façon d'écrire le préfixe suivant :

```
3EDC:BA98:7654:3::/56
```

1.3 Structure, allocation et routage d'une adresse IPv6

Les 64 premiers bits de l'adresse IPv6 (préfixe) servent généralement à l'adresse de sous-réseau, tandis que les 64 bits suivants identifient l'hôte à l'intérieur du sous-réseau : ce découpage joue un rôle un peu similaire aux masques de sous-réseau d'IPv4.

Cependant, la réservation et l'enregistrement d'adresses IPv6 routables sur Internet se fait par blocs dont le préfixe a une taille maximale de 64 bits. Toutes les adresses d'un même bloc sont routées de la même façon au travers des répartiteurs Internet et ne font l'objet d'aucune autre demande de réservation spécifique. L'usage des 64 derniers bits au sein de ce bloc reste privé, et ces bits peuvent ainsi encapsuler (de façon privée) une adresse MAC (48 bits), une adresse IPv4 (32 bits) voire même un nom d'hôte (jusqu'à 8 caractères), ce qui permet ainsi une configuration facile de dispositifs de traduction d'adresses IPv6 vers les hôtes d'un réseau local ne supportant pas IPv6.

1.4 Types d'adresses

Différentes sortes d'adresses IPv6 jouent des rôles particuliers. Ces propriétés sont indiquées par le début de l'adresse, appelé préfixe. IPv6 reconnaît trois types d'adresses : *unicast*, *multicast* et *anycast*.

Le type *unicast*, est le plus simple. Une adresse de ce type désigne une interface unique. Un paquet envoyé à une telle adresse, sera donc remis à l'interface identifiée par cette adresse. Parmi les adresses *unicast*, on peut distinguer celles qui ont une portée globale, c'est-à-dire désignant sans ambiguïté une machine sur le réseau Internet et celles qui ont une portée locale (lien ou site). Ces dernières ne sont pas routées sur l'Internet.

1.4.1 Unicast Global

L'Internet IPv6 est défini comme étant le sous-réseau 2000::/3 (les adresses commençant par un 2 ou un 3). Il s'agit des adresses unicast de portée globale. Seules ces adresses peuvent être routées. Toutes les autres adresses ne peuvent être utilisées que localement sur un même réseau physique (de niveau 2), ou par un accord privé de routage mutuel.

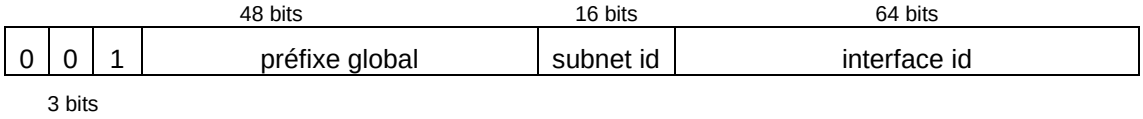


Figure 1 *Format des adresses de type plan agrégé*

Les adresses globales de type plan agrégé, proposée dans le RFC 3587, précise la structure d'adressage IPv6 définie dans le RFC 3513 en précisant les tailles de chacun des blocs. Une adresse intègre trois niveaux de hiérarchie :

- une topologie publique codée sur 48 bits, allouée par le fournisseur d'accès ;
- une topologie de site codée sur 16 bits. Ce champ permet de coder les numéros de sous réseau du site ;

- un identifiant d'interface codée sur 64 bits distinguant les différentes machines sur le lien.

Il existe plusieurs instanciations du plan d'adressage agrégé. Historiquement la première (préfixe 3FFE::/16) a servi aux réseaux expérimentaux, puis une seconde (préfixe 2001::/16) est définie par les autorités régionales pour les réseaux dits de production et enfin une troisième est dédiée (préfixe 2002::/16) au mécanisme de transition 6to4. Ces instanciations se différencient par la valeur du préfixe initial de 16 bits.

Global Unicast Prefix	Assignment	Date			
2001:0000::/23	IANA	01 Jul 99	2001:4800::/23	ARIN	24 Aug 04
2001:0200::/23	APNIC	01 Jul 99	2001:4A00::/23	RIPE NCC	15 Oct 04
2001:0400::/23	ARIN	01 Jul 99	2001:4C00::/23	RIPE NCC	17 Dec 04
2001:0600::/23	RIPE NCC	01 Jul 99	2001:5000::/20	RIPE NCC	10 Sep 04
2001:0800::/23	RIPE NCC	01 May 02	2001:8000::/19	APNIC	30 Nov 04
2001:0A00::/23	RIPE NCC	02 Nov 02	2001:A000::/20	APNIC	30 Nov 04
2001:0C00::/23	APNIC	01 May 02	2002:0000::/16	6to4	01 Feb 01
2001:0E00::/23	APNIC	01 Jan 03	2003:0000::/18	RIPE NCC	12 Jan 05
2001:1200::/23	LACNIC	01 Nov 02	2400:0000::/19	APNIC	20 May 05
2001:1400::/23	RIPE NCC	01 Feb 03	2400:2000::/19	APNIC	08 Jul 05
2001:1600::/23	RIPE NCC	01 Jul 03	2400:4000::/21	APNIC	08 Aug 05
2001:1800::/23	ARIN	01 Apr 03	2404:0000::/23	APNIC	19 Jan 06
2001:1A00::/23	RIPE NCC	01 Jan 04	2600:0000::/22	ARIN	19 Apr 05
2001:1C00::/22	RIPE NCC	01 May 04	2604:0000::/22	ARIN	19 Apr 05
2001:2000::/20	RIPE NCC	01 May 04	2608:0000::/22	ARIN	19 Apr 05
2001:3000::/21	RIPE NCC	01 May 04	260C:0000::/22	ARIN	19 Apr 05
2001:3800::/22	RIPE NCC	01 May 04	2610:0000::/23	ARIN	17 Nov 05
2001:3C00::/22	RESERVED	11 Jun 04	2800:0000::/23	LACNIC	17 Nov 05
2001:4000::/23	RIPE NCC	11 Jun 04	2A00:0000::/21	RIPE NCC	19 Apr 05
2001:4200::/23	ARIN	01 Jun 04	2A01:0000::/26	RIPE NCC	15 Dec 05
2001:4400::/23	APNIC	11 Jun 04	3FFE:0000::/16	6BONE	01 Dec 98
2001:4600::/23	RIPE NCC	17 Aug 04			

Tableau 1 *Attribution des préfixes d'adresses Unicast Global*

Parmi les adresses de 2000::/3, on distingue donc :

- Les adresses permanentes (2000::/16) allouées transitoirement avant l'ouverture du registre officiel.
- Les adresses permanentes (2001::/16) ouvertes à la réservation depuis 2001.
- Les adresses 6to4 (2002::/16) permettant d'acheminer le trafic IPv6 via un ou plusieurs réseaux IPv4.
- Les adresses du 6bone (3FFE::/16) pour l'expérimentation des interconnexions de réseaux IPv6.

- Toutes les autres adresses routables (plus des trois quarts) sont actuellement réservées pour usage ultérieur.

1.4.2 Lien-Local

Les adresses de type lien-local (*link local use address*) appartiennent à FE80::/64. Ces adresses sont locales dans le sens où elles sont non routables. Leur utilisation est restreinte au même réseau local de niveau 2. Il s'agit d'adresses dont la validité est restreinte à un lien. Une telle adresse identifie un ensemble d'interfaces directement connectées sans routeur intermédiaire : des machines branchées sur un même Ethernet, des machines reliées par une connexion PPP, ou aux extrémités d'un tunnel. Les adresses lien-local sont configurées automatiquement à l'initialisation de l'interface et permettent la communication entre noeuds voisins. L'adresse est obtenue en concaténant le préfixe FE80::/64 aux 64 bits de l'identifiant d'interface.

Parmi elles, les adresses du bloc FE80::/96 correspondent bit-à-bit aux adresses IPv4 et ne nécessitent aucune configuration (cependant il s'agit quand même d'une interface logique différente). Par exemple, l'adresse IPv6 fe80::172.16.1.2 sera assignée automatiquement à la même interface physique que l'adresse IPv4 172.16.1.2.

Les adresses lien-local sont utilisées par les protocoles de configuration d'adresse globale, de découverte de voisins (*neighbor discovery*) et de découverte de routeurs (*router discovery*). Ce sont de nouveaux dispositifs, le premier supplantant en particulier le protocole ARP (*Address Resolution Protocol*), qui permettent à un réseau local de se configurer automatiquement (voir Découverte de voisins).

Le fait que ces adresses aient une portée très faible les limite dans la pratique au cas où un démarrage automatique (*bootstrap*) est nécessaire. Leur usage ne doit pas être généralisé dans les applications classiques en régime stabilisé.

1.4.3 Identifiant d'interface

Les types d'adresses global ou lien-local utilisent un identifiant sur 64 bits pour désigner une interface connectée sur un lien. Si cette longueur n'est pas directement imposée par la norme d'adressage d'IPv6 RFC 3513, elle bénéficie d'un fort consensus car elle permet de garantir facilement une unicité sur le lien et par conséquent de faciliter l'auto-configuration des équipements.

Plusieurs techniques ont été élaborées à l'IETF. La plus répandue est basée sur l'utilisation d'une valeur unique par construction comme l'adresse MAC de la machine. Mais l'on peut également choisir une valeur aléatoire pour garantir plus de confidentialité ou au contraire la dériver d'une clé publique pour mieux authentifier l'émetteur du message. La taille de 64 bits permet de réduire à une valeur proche de zéro la probabilité de conflits. Enfin dans certains cas l'affectation manuelle de cette valeur peut se justifier.

1.4.4 Unique Local Address

Les adresses de type site-local étant supprimées du standard IPv6 (RFC 3879), le RFC 4193 définit un nouveau format d'adresse unicast : les adresses uniques locales (ULA : *Unique Local Address*). Ces adresses sont destinées à une utilisation locale. Elles ne sont pas définies pour être routées dans l'Internet, mais seulement au sein d'une zone limitée telle qu'un site ou entre un nombre limité de sites. Les adresses uniques locales ont les caractéristiques suivantes :

- Préfixe globalement unique.

- Préfixe clairement définit facilitant le filtrage sur les routeurs de bordure.
- Permet l'interconnexion de sites sans générer de conflit d'adresse et sans nécessiter de renumérotation.
- Indépendantes des fournisseurs d'accès à l'Internet et ne nécessitent donc pas de connectivité.
- Pas de conflit en cas de routage par erreur en dehors d'un site.
- Aucune différence pour les applications, qui peuvent les considérer comme des adresses globales unicast standard.

Les adresses uniques locales sont créées en utilisant un identifiant global (Global ID) généré pseudo-aléatoirement. Ces adresses suivent le format suivant :

- Prefix (7 bits) : FC00::/7 préfixe identifiant les adresses IPv6 locales (ULA)
- L (1 bit) : Positionné à 1, le préfixe est assigné localement. La valeur 0 est réservée pour une utilisation future.
- Global ID (40 bits) : Identifiant global utilisé pour la création d'un préfixe unique (Globally Unique Prefix).
- Subnet ID (16 bits) : Identifiant d'un sous réseau à l'intérieur du site.
- Interface ID (64 bits) : L'indentifiant d'interface tel que définit dans Identifiant d'interface.

1.4.5 Adresse indéterminée

L'adresse indéterminée (*unspecified address*) est utilisée comme adresse source par un nœud du réseau pendant son initialisation, avant d'acquérir une adresse. Sa valeur est `0:0:0:0:0:0:0:0` (en abrégé ::).

Cette adresse est utilisée uniquement par des protocoles d'initialisation, elle n'est jamais attribuée à un nœud et n'apparaît jamais comme adresse destination d'un paquet IPv6.

1.4.6 Adresse de bouclage

L'adresse de bouclage (*loopback address*) vaut `0:0:0:0:0:0:0:1` (en abrégé ::1). C'est l'équivalent de l'adresse 127.0.0.1 d'IPv4. Elle est utilisée par un nœud pour s'envoyer à lui-même des paquets IPv6. Un paquet IPv6 transitant sur le réseau ne peut donc avoir l'adresse de bouclage comme adresse source ni comme adresse destination.

1.4.7 Multicast

Une adresse de type multicast désigne un ensemble d'interfaces. Elle est caractérisée par le préfixe FF00 ::/8. Son format est le suivant :

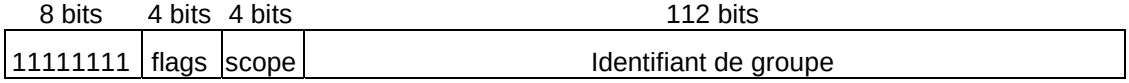


Figure 2 *Format des Adresses de multicast*

Le champ « flags » est un mot de 4 bits. Les 3 premiers bits sont réservés et doivent être initialisés à zéro. Le dernier bit, nommé T, s'il est positionné à 0, indique que la validité de l'adresse est permanente (auquel cas elle est nécessairement attribuée par une autorité compétente de l'Internet),

sinon l'adresse est temporaire (*transient address*). Le champ « scope », également de longueur de 4 bits, est le niveau de diffusion (scope) de l'adresse considérée. Les valeurs actuellement définies sont :

- 1 - node-local
- 2 - link-local
- 3 - subnet-local
- 4 - admin-local
- 5 - site-local
- 8 - organisation-local
- E - global
- Les portées 0 et F sont réservées.

Les adresses « multicast sollicité » sont construites à partir d'une adresse unicast ou anycast en concaténant le préfixe FF02::1:FF00:0/104 aux 24 deniers bits extraits de celle-ci. Un équipement, à partir de chacune de ses adresses IPv6 construit une adresse de multicast sollicité et écoute les paquets émis vers cette adresse. Les autres équipements sur le lien qui ne connaissent que l'adresse IPv6 de cet équipement sur ce lien peuvent utiliser son adresse de multicast sollicité pour le joindre et obtenir son adresse MAC. Ils construisent son adresse de multicast sollicité de la même manière que l'équipement lui-même. De cette adresse de multicast sollicité est alors déduite l'adresse MAC de multicast en concaténant le préfixe 0x3333 et les 4 octets de poids faible de l'adresse multicast IPv6. Sur le lien-local, les paquets IPv6 multicast sont alors encapsulés dans une trame Ethernet dont le champ « Adresse de destination » contient une adresse MAC multicast : le 8ème bit de poids fort de cette adresse est positionné à 1. L'utilisation des adresses IPv6 « multicast sollicité » rend obsolète l'utilisation de la diffusion généralisée utilisée par des protocoles tel que ARP en IPv4.

1.4.7.a Pourquoi avoir aboli l'utilisation de la diffusion généralisée dans IPv6 ?

1.4.7.b Quelle est l'adresse MAC multicast correspondant à l'adresse multicast IPv6 FF1E::12:AC21:6521 ?

1.4.8 Anycast

Le type d'adresses anycast est une officialisation de propositions faites pour IPv4 RFC 1546. Comme dans le cas du multicast, une adresse de type anycast désigne un groupe d'interfaces, la différence étant que lorsqu'un paquet a pour destination une telle adresse, il est acheminé à un des éléments du groupe et non pas à tous. C'est, par exemple, le plus proche au sens de la métrique des protocoles de routage. Cet adressage est principalement expérimental.

2 SUPPORT DE TRANSMISSION

2.1 Encapsulation des datagrammes IPv6

La méthode de transport d'un datagramme IPv6 entre deux machines directement reliées entre elles par un lien physique est la même que pour IPv4 : le datagramme est tout d'abord routé vers une interface d'émission qui l'encapsule dans une trame ; cette trame est transmise sur le lien vers l'adresse physique de la machine destination ; la machine destination reçoit la trame sur son interface, la décapsule et la traite.

2.1.1 Donnez les principales différences entre la méthode de transport des datagrammes IPv6 et IPv4.

2.2 Réseau à diffusion : Ethernet

Les datagrammes IPv6 utilisent l'encapsulation standard Ethernet V2, chaque trame contenant un seul datagramme. L'en-tête de la trame Ethernet contient les adresses Ethernet source et destination, et le champ type de protocole vaut 0x86DD. La structure d'une trame est donnée à la figure Encapsulation Ethernet.

La taille maximale d'un datagramme pouvant être transmis directement par une interface Ethernet (MTU) est normalement de 1 500 octets. Une valeur différente peut être forcée par configuration manuelle ou en utilisant l'option MTU des annonces de routeurs.

Pour la construction des adresses lien-local et des adresses auto-configurées, l'identifiant d'interface est celui dérivé de l'adresse MAC IEEE 802 constructeur de l'interface Ethernet. L'IEEE a défini un identificateur global à 64 bits (format EUI-64) pour les réseaux IEEE 1394 qui vise une utilisation dans le domaine de la domotique. L'IEEE décrit les règles qui permettent de passer d'un identifiant MAC codé sur 48 bits à un EUI-64.

Il existe plusieurs méthodes pour construire l'identifiant :

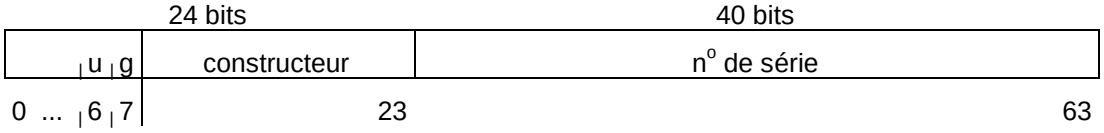


Figure 3 *Identificateur global EUI-64*

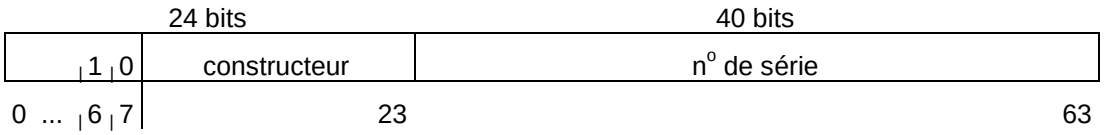


Figure 4 *Identificateur d'interface dérivé d'une EUI-64*

- Si une machine ou une interface possède un identificateur global IEEE EUI-64, celui-ci a la structure décrite par la figure Identificateur global IEEE EUI-64. Les 24 premiers bits de l'EUI-64, comme pour les adresses MAC IEEE 802, identifient le constructeur et les 40 autres bits identifient le numéro de série (les adresses MAC IEEE 802 n'en utilisaient que 24). Les 2 bits u (septième bit du premier octet) et g (huitième bit du premier octet) ont une signification spéciale :
  - u (Universel) vaut 0 si l'identifiant EUI-64 est universel,
  - g (Groupe) indique si l'adresse est individuelle (g = 0), c'est-à-dire désigne un seul équipement sur le réseau, ou de groupe (g = 1), par exemple une adresse de multicast.

  L'identifiant d'interface à 64 bits est dérivé de l'EUI-64 en inversant le bit u (cf. figure Identificateur d'interface dérivé d'une EUI-64). En effet, pour la construction des adresses IPv6, on a préféré utiliser 1 pour marquer l'unicité mondiale. Cette inversion de la sémantique du bit permet de garder la valeur 0 pour une numérotation manuelle, autorisant à numéroter simplement les interfaces locales à partir de 1.

- Si une interface possède une adresse MAC IEEE 802 à 48 bits universelle (cas des interfaces Ethernet ou FDDI), cette adresse est utilisée pour construire des identifiants d'interface sur 64 bits, comme indiqué sur la figure ci-dessous.

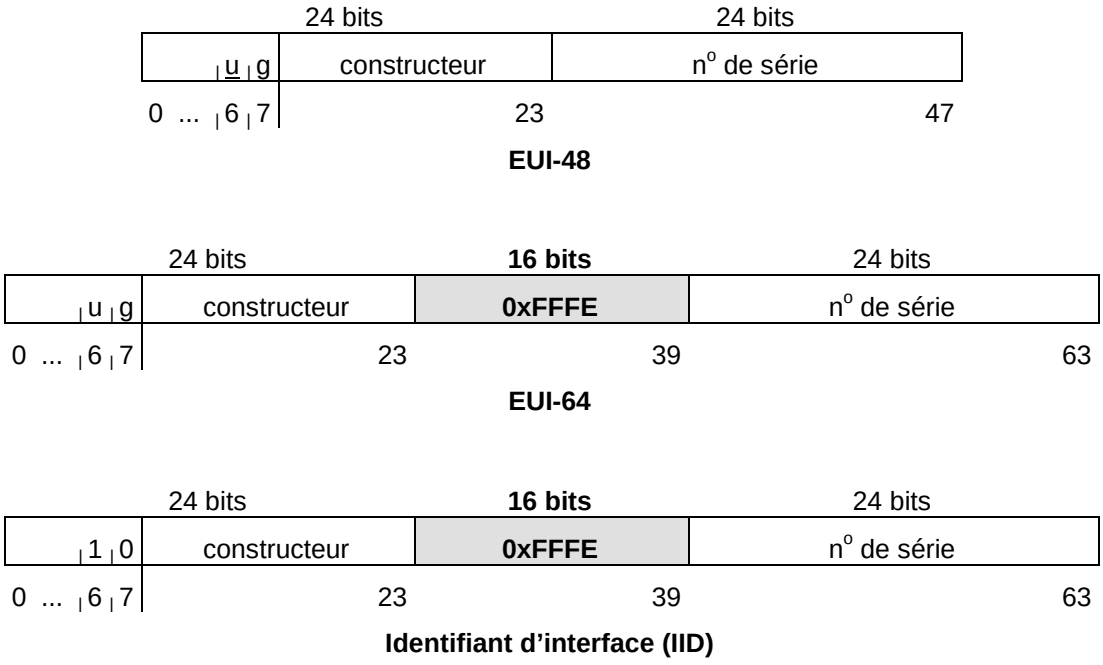


Figure 5 *Transformation d'une adresse MAC en identifiant d'interface*

2.2.1 Quels sont l'identifiant d'interface et l'adresse lien-local d'une interface physique de type Ethernet ayant pour adresse constructeur 34:56:78:9A:BC:DE ? Quelle est l'adresse « multicast sollicité » associé à cette interface ainsi que l'adresse MAC de multicast correspondante ?

3 DATAGRAMME IPV6

3.1 Format de l'en-tête des datagrammes IPv6

Avec la représentation des adresses employées, le format des datagrammes est la modification la plus visible pour l'utilisateur expérimenté et l'ingénieur de réseau. La modification de la taille des adresses conduit à une taille d'en-tête de 40 octets (le double de l'en-tête IPv4 sans les options). Hormis ce changement de taille, le format des en-têtes IPv6 a été simplifié ce qui permet aux routeurs de meilleures performances dans leurs traitements.

Le format d'en-tête d'un paquet IPv6 est donné par le RFC 2474 :

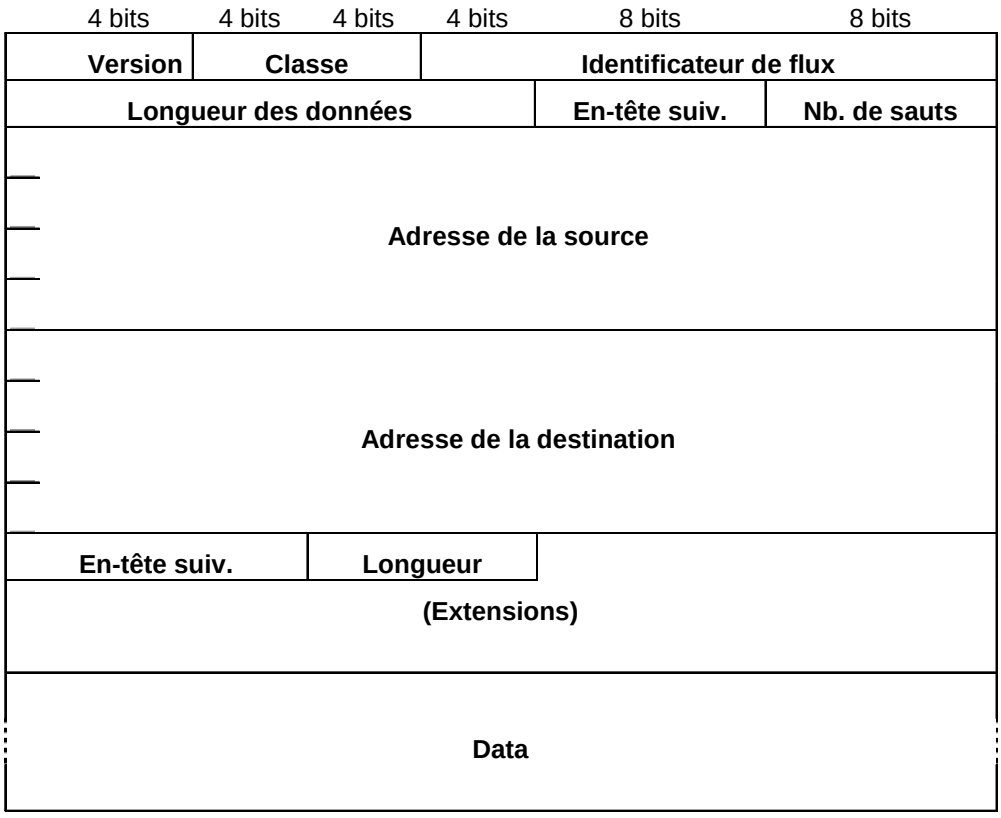


Figure 6 *Format d'un datagramme IPv6*

- Le champ « Version » est le seul champ qui occupe la même position dans l'en-tête des paquets IPv6 et des paquets IPv4. Sa valeur est 6.
- Le champ « Classe de trafic » (cf. RFC 2460) permet la différenciation de services conformément aux spécifications du RFC 2474. Dans les paquets IPv4, il prend la place du champ ToS, initialement défini dans la spécification d'IPv4. Le champ « Classe de trafic » est aussi appelé octet DiffServ (DS). Le champ DS est découpé en deux parties. Le sous-champ DSCP (*DiffServ Code Point*) contient les valeurs des différents comportements. Les deux derniers bits du champ sont actuellement non utilisés, mais devraient servir aux routeurs pour indiquer un risque de congestion en combinaison avec l'algorithme RED (*Random Early Detection*).

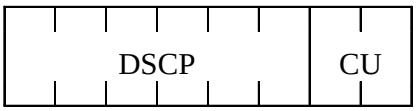


Figure 7 *Champ « Classe de trafic »*

- Le champ « Identificateur de flux » contient un numéro unique choisi par la source, qui a pour but de faciliter le travail des routeurs et la mise en oeuvre des fonctions de qualité de service comme RSVP. Cet indicateur peut être considéré comme une marque à un contexte dans le routeur. Le routeur peut alors faire un traitement particulier : choix d'une route, traitement en "temps-réel" de l'information.
- Contrairement à IPv4, le champ « Longueur des données utiles » (*payload*), codé sur deux octets, ne contient que la taille des données utiles, sans prendre en compte la longueur de

l'en-tête. Pour des paquets dont la taille des données serait supérieure à 65 535 ce champ vaut 0 et l'option *jumbogramme* de l'extension de « proche-en-proche » est utilisée.

- Le champ « En-tête suivant » a une fonction similaire au champ « Protocole » du paquet IPv4. Il identifie le prochain en-tête. Il peut s'agir d'un protocole (ICMP ou de niveau supérieur tels que UDP, TCP, etc.) ou de la désignation d'extensions. Les extensions contiennent aussi ce champ pour permettre un chaînage.

Valeur	Extension
0	Proche-en-proche
6	TCP
17	UDP
41	IPv6
43	Routage
44	Fragmentation
50	Confidentialité
51	Authentification
58	ICMPv6
59	Fin d'en-têtes
60	Destination
132	SCTP
135	Mobilité
136	UDP-lite

Tableau 2 *Valeurs du champ « En-tête suivant »*

- Le champ « Nombre de sauts » contient une valeur décrémentée à chaque nœud traversé. Un datagramme retransmis par un routeur est rejeté avec l'émission d'un message d'erreur ICMPv6 vers la source si la valeur après décrémentation atteint 0.

3.1.1 En vous basant sur le format des entêtes IPv6, listez les différences avec celui des entêtes IPv4.

3.1.2 Quel impact a la disparition du champ *checksum* dans l'entête IPv6 sur les protocoles de niveau supérieur ?

3.1.3 Le paquet IPv6 suivant a été capturé au cours d'une connexion FTP. Procédez au décodage de son entête.

```
0000: 60 00 00 00 00 28 06 40 3f fe 03 02 00 12 00 02
0010: 00 00 00 00 00 00 00 13 3f fe 03 05 10 02 00 01
0020: 02 00 c0 ff fe 11 cb a0 ff b3 00 15 55 4d fd d1
0030: 00 00 00 00 a0 02 40 00 7d 76 00 00 02 04 05 a0
0040: 01 03 03 00 01 01 08 0a 00 9a 17 44 00 00 00 00
```

4 LES EXTENSIONS

Les extensions d'IPv6 peuvent être vues comme un prolongement de l'encapsulation d'IP dans IP. À part l'extension de proche-en-proche traitée par tous les routeurs intermédiaires, les autres extensions ne sont prises en compte que par les équipements destinataires du paquet.

Une extension a une longueur multiple de 8 octets. Elle commence par un champ « en-tête suivant » codé sur un octet. La valeur prise par ce champ définit le type de données qui vient à la suite de cette extension : il peut s'agir d'une autre extension ou d'un protocole de niveau 4. Pour les extensions à longueur variable, l'octet suivant contient la longueur de l'extension exprimée en mots de 8 octets, le premier n'étant pas compté (une extension de 16 octets aura donc un champ longueur de 1).

Si plusieurs extensions sont présentes dans un même en-tête, leur ordre d'apparition est le suivant (cf. RFC 2460) :

- Proche-en-proche (doit toujours être en première position)
- Destination (sera aussi traité par les routeurs listés dans l'extension de routage par la source)
- Routage par la source
- Fragmentation
- Authentification
- Destination (traité uniquement par l'équipement terminal)

4.1 Extension Proche-en-proche

L'extension **proche-en-proche** (*hop-by-hop*) se situe toujours en première position et est traitée par tous les routeurs que le paquet traverse. Le type associé (contenu dans le champ « En-tête suivant » de l'en-tête précédent) est 0 et le champ longueur de l'extension contient le nombre de mots de 8 octets (64 bits) moins 1. Cette extension est composée d'options. Pour l'instant, seules quatre options, dont deux de bourrage, sont définies. Chaque option est une suite d'octets. Le premier octet de ces 4 options est un type, le deuxième (sauf pour l'option 0) contient la longueur de l'option en octets moins 2. Les deux premiers bits de poids fort du type définissent le comportement du routeur quand il rencontre une option inconnue :

- 00 : le routeur ignore l'option ;
- 01 : le routeur rejette le paquet ;
- 10 : le routeur rejette le paquet et retourne un message ICMPv6 d'inaccessibilité ;
- 11 : le routeur rejette le paquet et retourne un message ICMPv6 d'inaccessibilité si l'adresse de destination n'est pas multicast.

Le bit suivant (i.e. le 3^{e} de poids fort) du type indique que le routeur peut modifier le contenu du champ option (valeur à 1) ou non (valeur à 0).

Les quatre options de proche-en-proche sont :

1. Pad1 (type 0). Cette option est utilisée pour introduire un octet de bourrage.
2. Padn (type 1). Cette option est utilisée pour introduire plus de 2 octets de bourrage. Le champ longueur indique le nombre d'octets qui suivent.

3. L'option *Jumbogramme* (type 194 ou 0xc2, RFC 2675). Cette option est utilisée quand le champ longueur des données du paquet IPv6 n'est pas suffisant pour coder la taille du paquet. Cette option est essentiellement prévue pour la transmission à grand débit entre deux équipements. Si l'option jumbogramme est utilisée, le champ longueur des données utiles dans l'en-tête IPv6 vaut 0.
4. L'option *Router Alert* (type 5 RFC 2675) demande à un routeur d'examiner le contenu des données qu'il relaie (Router Alert existe également en IPv4, RFC 2113). En principe, le processus de relayage (recopier le paquet sur une interface de sortie en fonction de l'adresse destination et des tables de routage) doit être le plus rapide possible. Mais pour des protocoles comme la gestion des groupes de multicast avec MLD (*Multicast Listener Discovery*) ou la signalisation des flux avec RSVP, tous les routeurs intermédiaires doivent tenir compte des données.

   L'émetteur envoie les données à la destination, mais s'il précise l'option Router Alert, les routeurs intermédiaires vont analyser les données, voire modifier leur contenu avant de relayer le paquet. Ce mécanisme est efficace puisque les routeurs n'ont pas à analyser le contenu de tous les paquets d'un flux. Le champ « valeur » de l'option contient :

   – 0 : pour les messages du protocole MLD de gestion des groupes multicast ;
   – 1 : pour les messages RSVP ;
   – 2 : pour les Réseaux Actifs ;
   – les autres valeurs sont réservées.

4.1.1 Comment se comporte un routeur lorsqu'il reçoit un paquet contenant l'extension proche-en-proche avec l'option *Jumbogramme* qu'il ne sait pas traiter ? Même question avec l'option *Router Alert*.

4.1.2 Pour quelle option de l'extension Proche-en-proche, un routeur peut-il modifier le contenu l'option ?

4.2 Extension Destination

L'extension **destination**, dont le format est identique à l'extension de proche-en-proche (cf. figure Format des extensions « proche-en-proche » et « destination »), contient des options qui sont traitées par l'équipement destinataire. Pour l'instant, à part les options de bourrage (voir Pad1 et Padn) et de mobilité, la seule autre option concerne le tunnelage dans des paquets IPv6 (RFC 2473). Cette option permet de limiter le niveau d'encapsulation dans des tunnels des paquets IPv6.

4.3 Extension Routage

Cette extension permet d'imposer à un paquet une route différente de celle offerte par les politiques de routage présentes sur le réseau. Pour l'instant seul le routage par la source (type = 0), similaire à l'option *Loose Source Routing* d'IPv4, est défini.

Dans IPv4, le routage peut être strict (le routeur suivant présent dans la liste doit être un voisin directement accessible) ou libéral (*loose*) (un routeur peut utiliser les tables de routage pour joindre le routeur suivant servant de relais). Dans IPv6, seul le routage libéral est autorisé. En effet, le routage strict était initialement mis en place surtout pour des raisons de sécurité. La source devait être absolument sûre du chemin pris par les paquets. Cette utilisation a maintenant disparu du réseau.
Le principe du routage par la source ou *Source Routing* dans IPv4 est le même pour IPv6. L'émetteur met dans le champ destination du paquet IPv6, l'adresse du premier routeur servant de

relais, l'extension contient la suite de la liste des autres routeurs relais et le destinataire. Quand un routeur reçoit un paquet qui lui est adressé comportant une extension de routage par la source, il permute son adresse avec l'adresse du prochain routeur et réémet le paquet vers cette adresse suivante. La figure suivante donne le format de l'extension de routage par la source :

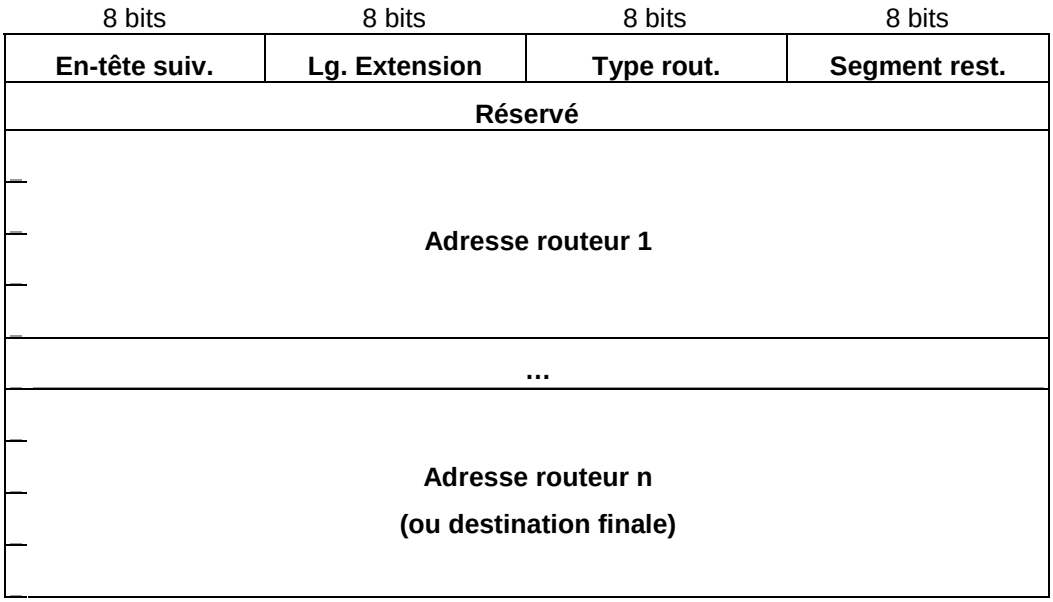


Figure 8 *Format de l'extension de routage*

- Le champ « longueur de l'en-tête » indique le nombre de mots de 64 bits qui composent l'extension. Pour l'extension de type 0, cela correspond au nombre d'adresses présentes dans la liste, multiplié par 2.
- Le champ « type routage » indique la nature du routage. Pour l'instant, seul le routage par la source, de type 0 est spécifié. La suite de l'en-tête est propre à ce type.
- Le nombre de « segments restant » est décrémenté après la traversée d'un routeur. Il indique le nombre d'équipements qui doivent encore être traversés. Il permet de trouver dans la liste l'adresse qui devra être substituée.
- Les 32 bits suivants (« réservé ») sont inutilisés pour préserver l'alignement.

La liste comprenant les routeurs à traverser et le destinataire finale est fournie. Ces adresses ne peuvent pas être des adresses de multidiffusion (*multicast*).

4.3.1 Le paquet suivant a été capturé lors de l'ouverture d'une connexion telnet. La commande telnet permet de spécifier des paramètres de routage par la source. Ainsi telnet @routeur1 @destination permet un routage libéral vers la destination en passant par le routeur intermédiaire routeur1.

```
0000: 60 00 00 00 00 40 2b 40 3f fe 03 02 00 12 00 02
0010: 00 00 00 00 00 00 00 13 3f fe 03 02 00 12 00 05
0020: 02 a0 c9 ff fe aa 22 01|06 02 00 01 00 00 00 00
0030: 3f fe 03 05 10 02 00 01 02 00 c0 ff fe 11 cb a0|
0040: ff b1 00 17 17 10 7e 57 00 00 00 00 a0 02 40 00
0050: 35 6e 00 00 02 04 05 a0 01 03 03 00 01 01 08 0a
0060: 00 9a 1d 04 00 00 00 0b
```

a. Décoder la trace en donnant le maximum d'informations.

b. Quelle est l'adresse de la destination finale de ce paquet ?

4.4 Extension Fragmentation

Le format de l'extension de fragmentation est donné dans la figure suivante. La signification des champs est identique à celle d'IPv4 :

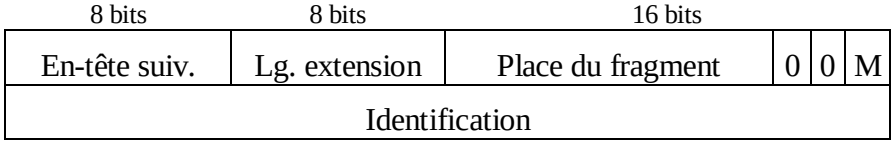


Figure 9 *Format de l'extension de fragmentation*

- Le champ « Place du fragment » indique lors du réassemblage où les données doivent être insérées. Ceci permet de parer les problèmes dus au déséquencement dans les réseaux orientés datagrammes. Comme ce champ est sur 13 bits, la taille de tous les segments, sauf du dernier, doit être multiple de 8 octets.
- Le bit « M » s'il vaut 1 indique qu'il y aura d'autres fragments émis.
- Le champ « Identification » permet de repérer les fragments appartenant à un même paquet initial. Il est différent pour chaque paquet et recopié dans ses fragments.
- Le bit « DF » (Don't Fragment) n'est plus nécessaire puisque, si un paquet est trop grand, il y aura rejet du paquet par le routeur.

4.4.1 Pourquoi avoir prévu une extension fragmentation alors que le mécanisme de découverte du PMTU permet d'adapter la taille des paquets IPv6 à l'émission ? Au niveau de quel type d'équipements fragmentation et réassemblage ont-ils lieu ?

4.4.2 Les paquets suivants correspondent à l'envoi d'un datagramme de longueur 3 500 octets en UDP alors que le MTU de l'interface est 1 500.

a. Décoder les deux traces en donnant le maximum d'informations.

Trace 1

```
0000: 60 00 00 00 05 b0 2c 40 3f fe 03 02 00 12 00 02
0010: 00 00 00 00 00 00 00 13 3f fe 03 02 00 12 00 05
0020: 02 a0 c9 ff fe aa 22 01|11 00 00 01 00 00 00 8e|
0030: f3 8e 00 0d 0d b4 c2 27 30 31 32 33 34 35 36 37
0040: 38 39 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e
0050: 4f 50 51 52 53 54 55 56 57 58 59 5a 61 62 63 64
0060: 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74
0070: ...
```

Trace 2

```
0000: 60 00 00 00 05 b0 2c 40 3f fe 03 02 00 12 00 02
0010: 00 00 00 00 00 00 00 13 3f fe 03 02 00 12 00 05
0020: 02 a0 c9 ff fe aa 22 01|11 00 05 a9 00 00 00 8e|
0030: 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54
```

```
0040: 55 56 57 58 59 5a 61 62 63 64 65 66 67 68 69 6a
0050: 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78 79 7a
0060: 30 31 32 33 34 35 36 37 38 39 41 42 43 44 45 46
0070: 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56
0080: ...
```

b. Ces deux paquets sont-ils suffisants pour envoyer le datagramme UDP en question ? S'ils ne le sont pas, donner le format du ou des autres paquets.

4.5 Extension Sécurité

Deux extensions de sécurité -- les extensions d'authentification AH (*Authentication Header*) et de confidentialité ESP (*Encapsulating Security Payload*) -- sont définies par l'IETF. Elles permettent de protéger les communications passées sur les réseaux IPv6 mais aussi IPv4 en assurant les services de confidentialité, authentification, intégrité et détection de rejeux. Le chapitre Sécurité du RFC 2402 donne une description détaillée de ces extensions, et présente les modes de protection existants.

5 ICMPv6

Le protocole de contrôle d'IP a été prévu. Dans IPv4, ICMP (Internet Message Control Protocol) sert à la détection d'erreurs (par exemple : équipement inaccessible, durée de vie expirée,...), au test (par exemple ping), à la configuration automatique des équipements (redirection ICMP, découverte des routeurs). Ces trois fonctions ont été mieux définies dans IPv6. De plus ICMPv6 (RFC 2463) intègre les fonctions de gestion des groupes de multicast (MLD : Multicast Listener Discovery) qui sont effectuées par le protocole IGMP (Internet Group Message Protocol) dans IPv4. ICMPv6 reprend aussi les fonctions du protocole ARP utilisé par IPv4.

Le protocole se voit attribuer le numéro 58. Le format générique des paquets ICMPv6 est le suivant :

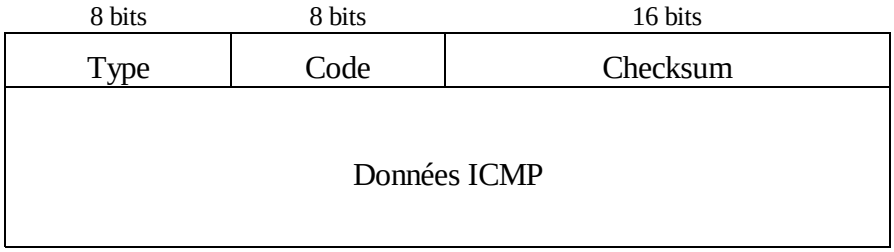


Figure 10 *Format générique des paquets ICMPv6*

- Le champ « Type » code la nature du message ICMPv6. Contrairement à IPv4 où la numérotation ne suivait aucune logique, les valeurs inférieures à 127 sont réservées aux messages d'erreur. Les autres valeurs réservées aux messages. d'information, parmi lesquels se trouvent ceux utilisés par le protocole découverte des voisins (*neighbor discovery*) pour la configuration automatique des équipements.
- Le champ « Code » précise la cause du message ICMPv6.
- Le champ « Checksum » permet de vérifier l'intégrité du paquet ICMP. Ce champ est calculé avec le pseudo-en-tête tel que celui utilisé au niveau transport.

Les messages ICMPv6 de compte rendu d'erreur contiennent dans la partie « données » le paquet IPv6 ayant provoqué l'erreur. Pour éviter des problèmes de fragmentation puisqu'il est difficilement envisageable de mettre en œuvre la découverte du MTU, la longueur du message ICMPv6 est limitée à 1 280 octets et par conséquent le contenu du paquet IPv6 peut être tronqué.

Type	**Code**	**Nature**
Gestion des erreurs		
1	Destination inaccessible	
	0	aucune route vers la destination
	1	la communication avec la destination est administrativement interdite
	2	Hors portée de l'adresse source
	3	l'adresse est inaccessible
	4	le numéro de port est inaccessible
2	Paquet trop grand	
3	Temps dépassé	
	0	Limite du nombre de sauts atteinte
	1	Temps de réassemblage dépassé
4	Erreur de paramètre	
	0	champ d'en-tête erroné
	1	champ d'en-tête suivant non reconnu
	2	option non reconnue
Information		
128		Demande d'écho
129		Réponse d'écho
Gestion des groupes multicast (MLD, RFC 2710)		
130		Requête d'abonnement
131		Rapport d'abonnement
132		Fin d'abonnement
Découverte de voisins (RFC 2461)		
133		Sollicitation du routeur
134		Annonce du routeur
135		Sollicitation d'un voisin
136		Annonce d'un voisin
137		Redirection
Renumérotation des routeurs (expérimental, RFC 2894)		

138		renumérotation des routeurs :
	0	commande
	1	Résultat
	255	remise à zéro du numéro de séquence
Recherche d'information sur un noeud (expérimental)		
139		Demande d'information
140		Réponse
Découverte de voisins inverse (RFC 3122)		
141		Sollicitation
142		Annonce
Gestion des groupes multicast (MLDv2, RFC 3810)		
143		Rapport d'abonnement MLDv2
Mobilité (RFC 3775)		
144		Découverte d'agent mère (requête)
145		Découverte d'agent mère (réponse)
146		Sollicitation de préfixe mobile
147		Annonce de préfixe mobile
Découverte de voisins sécurisée (SEND, RFC 3971)		
148		Sollicitation de chemin de certification
149		Annonce de chemin de certification
Mobilité (expérimental)		
150		Protocoles de mobilité expérimentaux, tels que Seamoby

Tableau 3 *Valeurs des champs « type » et « code » d'ICMPv6*

5.1 Destination inaccessible

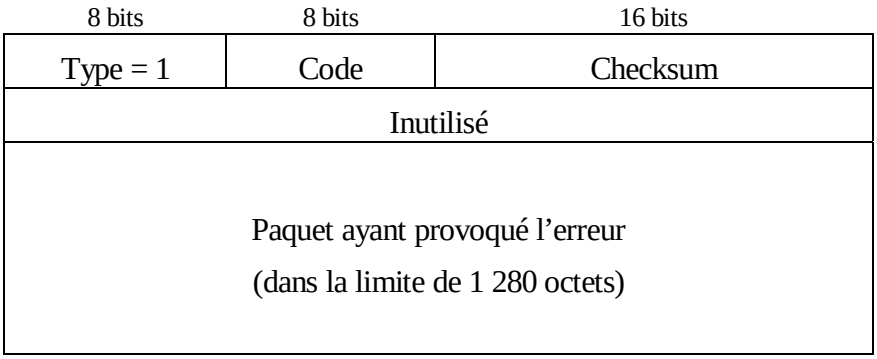


Figure 11 *Format d'un message ICMPv6 Destination inaccessible*

Ce message est émis par un routeur intermédiaire quand le paquet ne peut pas être transmis parce que soit :

- le routeur ne trouve pas dans ses tables la route vers la destination (code = 0) ;
- le franchissement d'un équipement de type firewall est interdit (« raison administrative », code = 1) ;
- l'adresse destination ne peut être atteinte avec l'adresse source fournie, par exemple si le message est adressé à un destinataire hors du lien, l'adresse source ne doit pas être une adresse lien-local (code = 2) ;
- toute autre raison comme par exemple la tentative de router une adresse locale au lien (code = 3) ;
- le destinataire peut aussi émettre un message ICMPv6 de ce type quand le port destination contenu dans le paquet n'est pas affecté à une application (code = 4).

5.2 Paquet trop grand

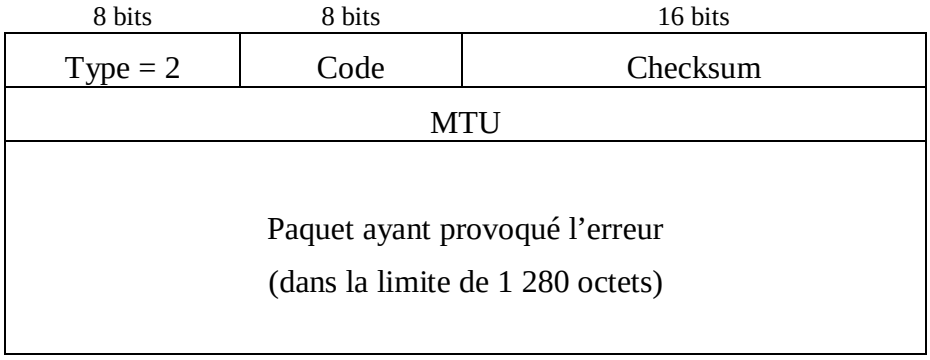


Figure 12 *Format d'un message ICMPv6 Paquet trop grand*

Ce message ICMPv6 est utilisé par le protocole de découverte du MTU pour trouver la taille optimale des paquets IPv6 pour qu'ils puissent traverser les routeurs. Ce message contient la taille du MTU acceptée par le routeur pour que la source puisse efficacement adapter la taille des données.

5.3 Temps dépassé

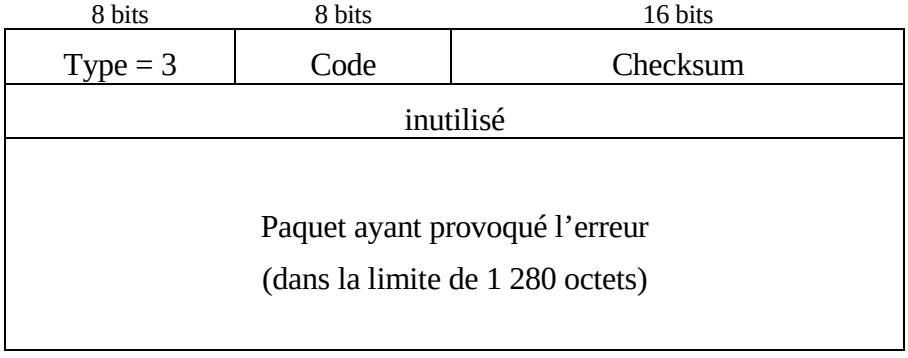


Figure 13 *Format d'un message ICMPv6 Temps dépassé*

Ce message indique que le paquet a été rejeté par le routeur :

- soit parce que le champ nombre de sauts a atteint 0 (code = 0) ;
- soit qu'un fragment s'est perdu et le temps alloué au réassemblage a été dépassé (code = 1).

Ce message sert aussi à la commande *traceroute* pour déterminer le chemin pris par les paquets.

5.4 Erreur de paramètre

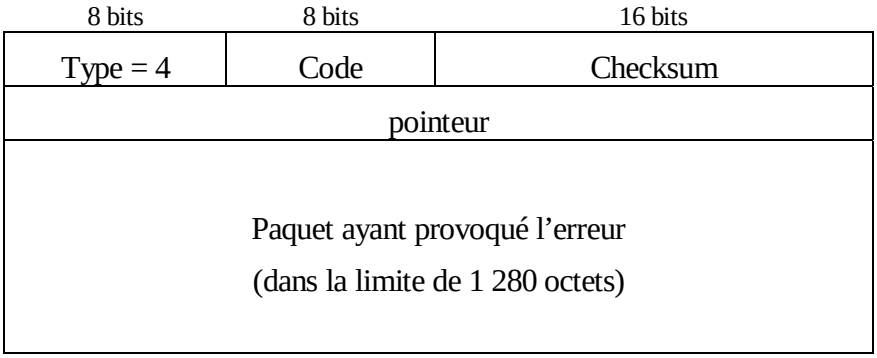


Figure 14 *Format d'un message ICMPv6 Erreur de paramètre*

Ce message est émis par un nœud ayant détecté une erreur de syntaxe dans l'en-tête du paquet IP ou des extensions. Le champ code révèle la cause de l'erreur :

- la syntaxe de l'en-tête n'est pas correcte (code = 0) ;
- le numéro en-tête suivant n'est pas reconnu (code = 1) ;
- une option de l'extension (par exemple proche-en-proche ou destination) n'est pas reconnue et le codage des deux bits de poids fort oblige à rejeter le paquet (code = 2).

Le champ pointeur indique l'octet où l'erreur est survenue dans le paquet retourné.

5.5 Demande et réponse d'écho

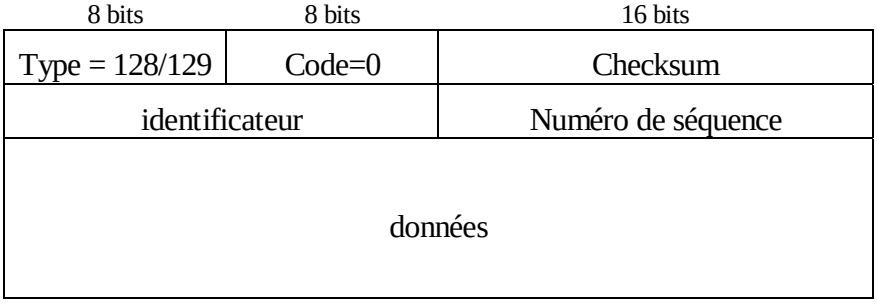


Figure 15 *Format d'un message ICMPv6 demande et réponse d'écho*

Ces deux messages servent en particulier à la commande *ping* permettant de tester l'accessibilité d'une machine. Le principe de fonctionnement est le même que pour IPv4, une requête (type 128) est envoyée vers l'équipement dont on veut tester le fonctionnement, celui-ci répond par le message réponse d'écho (type 129). Le champ identificateur permet de distinguer les réponses dans le cas où

plusieurs commandes ping seraient lancées simultanément sur la machine. Le champ numéro de séquence permet d'associer la réponse à une requête pour mesurer le temps d'aller et retour dans le cas où les demandes sont émises en continu et que le délai de propagation est élevé. Le champ données permet d'augmenter la taille du message pour les mesures.

6 DECOUVERTE DE VOISINS

Le protocole de découverte des voisins (*neighbor discovery*) permet à un équipement de s'intégrer dans l'environnement local, c'est-à-dire le lien sur lequel sont physiquement transmis les paquets IPv6. Il permet de dialoguer avec les équipements connectés au même support (stations et routeurs). Il ne s'agit pas pour un équipement de connaître exactement la liste de tous les autres équipements connectés sur le lien, mais uniquement de gérer ceux avec lesquels il dialogue.

Le protocole utilise cinq types de messages ICMPv6 (voir le tableau des *Valeurs des champs « type » et « code » d'ICMPv6*). Le champ nombre de sauts de l'en-tête IPv6 contient la valeur 255

6.1.1 Pourquoi utiliser la valeur 255 pour des paquets qui ne doivent pas être routés hors du lien physique ?

Le protocole réalise les différentes fonctions :

- Résolution d'adresses
- Détection d'inaccessibilité des voisins (ou NUD *Neighbor Unreachability Detection*)
- Configuration automatique
- Découverte des routeurs
- Découverte des préfixes
- Détection des adresses dupliquées
- Découverte des paramètres
- Indication de redirection

6.2 Données véhiculées par les messages

L'intérêt du protocole de découverte des voisins est d'unifier différents protocoles qui existent dans IPv4. En particulier, la plupart des données utilise un format d'options commun, ce qui simplifie la mise en œuvre du protocole. Le format contient les champs type, longueur en mots de 64 bits, données. La faible précision du champ longueur va introduire une perte de place. En contrepartie, elle va permettre aussi un alignement des options sur des mots de 64 bits, ce qui optimise leur traitement. Le tableau suivant décrit les cinq options générales utilisées dans les messages de découverte des voisins.

	Sollicitation du routeur	Annonce du routeur	Sollicitation du voisin	Annonce du voisin	Indication de rédirection
adresse physique de la source	présent	Présent	présent		
adresse physique de la cible				présent	présent
information sur le préfixe		>= 1			
en-tête redirigée					présent
MTU		possible			

Figure 16 *Utilisation des options dans les messages de découverte des voisins*

6.2.1 Adresse physique de la source/cible

La figure Format de l'option adresse physique source/cible donne le format de ces options. Le type 1 est réservé à l'adresse physique de la source et le type 2 à l'adresse de la cible.

Le champ «longueur» est la taille en mots de 64 bits de l'option. Dans le cas d'une adresse MAC, d'une longueur de 6 octets, il contient donc la valeur 1.

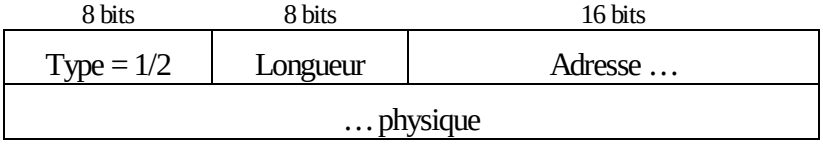
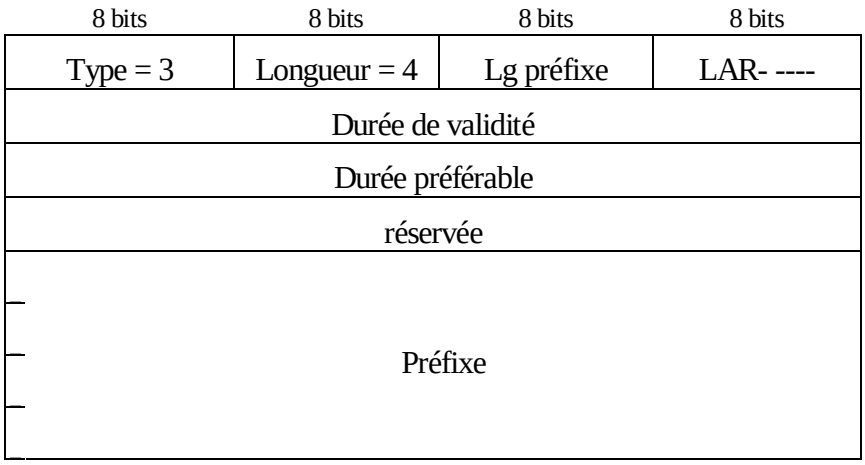


Figure 17 *Format de l'option adresse physique source/cible*

6.2.2 Information sur le préfixe

Cette option contient les informations sur le préfixe pour permettre une configuration automatique des équipements. Le champ type vaut 3 et le champ longueur vaut 4. La figure Format de l'option information sur le préfixe donne le format de l'option :

Figure 18 *Format de l'option information sur le préfixe*



- Le champ lg.préfixe indique combien de bits sont significatifs pour le préfixe annoncé dans un champ suivant.
- Le bit L indique, quand il est à 1, que le préfixe permet d'indiquer que tous les autres équipements partageant le même préfixe sont sur le même lien. L'émetteur peut donc directement les joindre. Dans le cas contraire, l'équipement émet le paquet vers le routeur. Si ce dernier sait que l'équipement émetteur peut joindre directement le destinataire, il émettra un message ICMPv6 d'indication de redirection.
- Le bit A indique, quand il est à 1, que le préfixe annoncé peut être utilisé pour construire l'adresse de l'équipement.
- Le bit R, indique, quand il est à 1, que le champ préfixe contient l'adresse globale d'un routeur «agent mère». Les bits de poids fort peuvent toujours être utilisés pour construire un préfixe.
- Le champ durée de validité indique en secondes la durée pendant laquelle le préfixe est valide.
- Le champ durée préférable indique la durée en secondes pendant laquelle une adresse construite avec le protocole de configuration sans état demeure «préférable» (cf. Durée de vie des adresses).

Pour ces deux champs, une valeur de 0xffffffff représente une durée infinie. Ces champs peuvent servir dans la phase de passage d'un fournisseur d'accès à un autre ; c'est-à-dire d'un préfixe à un autre.

- Le champ réservé permet d'aligner le préfixe sur une frontière de mot de 64 bits.
- Le champ préfixe contient la valeur de préfixe annoncé sur le lien. Pour maintenir un alignement sur 64 bits pour le reste des données du paquet, ce champ a une longueur fixe de 128 bits.

6.2.3 En-tête redirigée

Cette option est utilisée par le message d'indication de redirection. Elle permet d'encapsuler les premiers octets du paquet IPv6 qui a provoqué l'émission de ce message comme dans le cas des messages ICMPv6 d'erreur.

Le type vaut 4 et la taille de cette option ne doit pas conduire à un paquet IPv6 dépassant 1280 octets (cf. figure Format de l'option en-tête redirigée). Par contre le paquet doit contenir le maximum d'information possible.

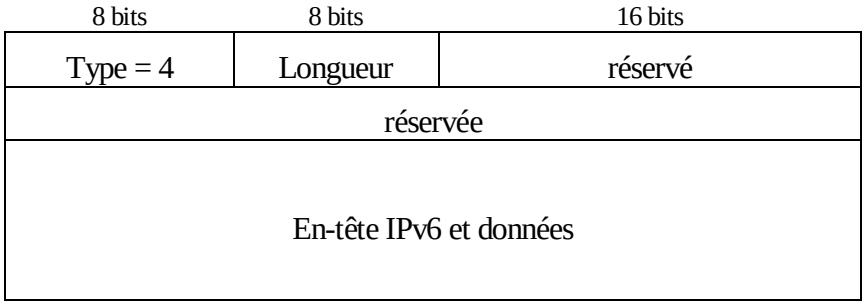


Figure 19 *Format de l'option En-tête redirigée*

6.2.4 MTU

Cette option permet d'informer les équipements sur la taille maximale des données pouvant être émises sur le lien. La figure Format de l'option MTU donne le format de cette option. Il n'est pas nécessaire de diffuser cette information si l'équipement utilise toujours la taille maximale permise. Par exemple, sur les réseaux Ethernet, les équipements utiliseront la valeur 1 500. Par contre pour les réseaux anneau à jeton ou FDDI, il est souvent nécessaire de préciser si les équipements doivent utiliser la valeur maximale permise ou une valeur inférieure pour autoriser l'utilisation de ponts.

Le champ type vaut 5 et le champ longueur 1.

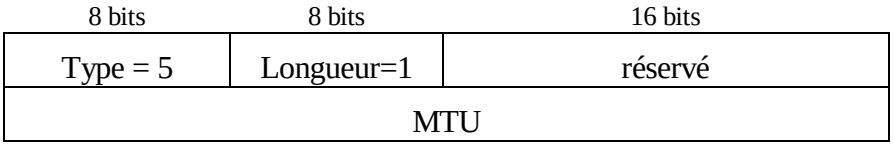


Figure 20 *Format de l'option MTU*

6.3 Messages de découverte de voisins

Les différentes fonctionnalités de découverte des voisins utilisent 5 messages : 2 pour le dialogue entre un équipement et un routeur, 2 pour le dialogue entre voisins et un dernier pour la redirection. Chacun de ces messages peut contenir des options.

- **Sollicitation du routeur**

Le message de sollicitation d'un routeur est émis par un équipement au démarrage pour recevoir plus rapidement des informations du routeur. Ce message est émis à l'adresse IPv6 de multicast réservée aux routeurs sur le même lien ff02::2. Si l'équipement ne connaît pas encore son adresse source, l'adresse non spécifiée est utilisée. Le champ option contient normalement l'adresse physique de l'équipement.

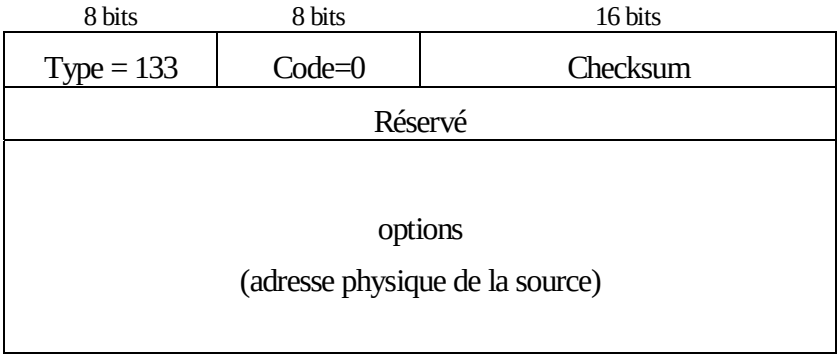


Figure 21 *Format des paquets de sollicitation du routeur*

- **Annonce du routeur**

  Ce message (cf. figure Format des paquets d'annonce du routeur) est émis périodiquement par les routeurs ou en réponse à un message de sollicitation d'un routeur par un équipement. Le champ adresse source contient l'adresse locale au lien du routeur, le champ destination contient soit l'adresse de l'équipement qui a émis la sollicitation, soit l'adresse de toutes les stations (ff02::01).

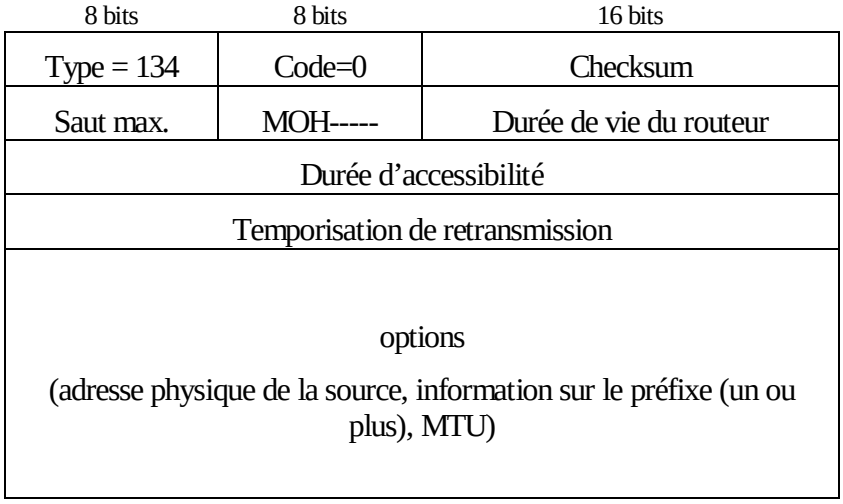


Figure 22 *Format des paquets d'annonce du routeur*

- **Sollicitation d'un voisin**

  Ce message permet d'obtenir des informations d'un équipement voisin, c'est-à-dire situé sur le même lien physique (ou connecté via des ponts). Le message peut lui être explicitement envoyé ou émis sur une adresse de diffusion. Dans le cas de la détermination de l'adresse physique, il correspond à la requête ARP du protocole IPv4.

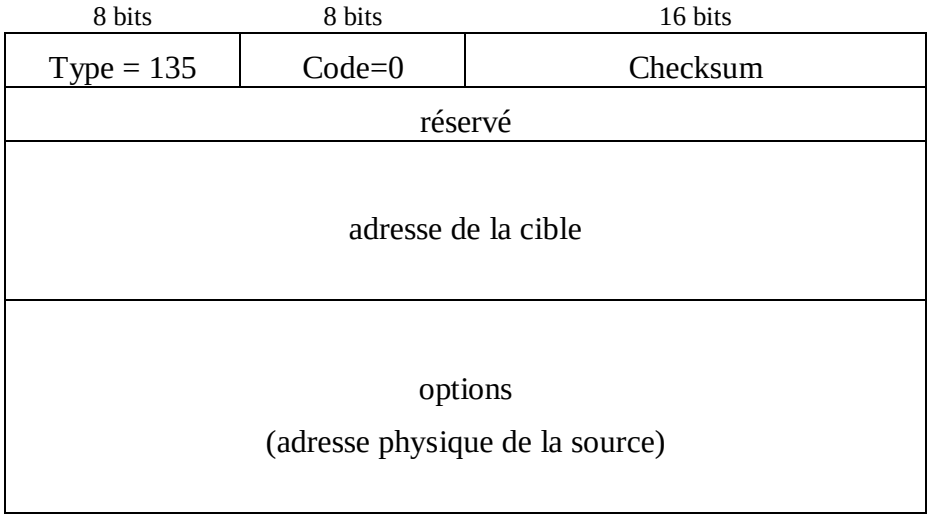


Figure 23 *Format des paquets de sollicitation d'un voisin*

- **Annonce d'un voisin**

  Ce message (cf. figure Format des paquets d'annonce d'un voisin) est émis en réponse à une sollicitation, mais il peut aussi être émis spontanément pour propager une information de changement d'adresse physique, ou de statut «routeur». Dans le cas de la détermination d'adresse physique, il correspond à la réponse ARP pour le protocole IPv4.

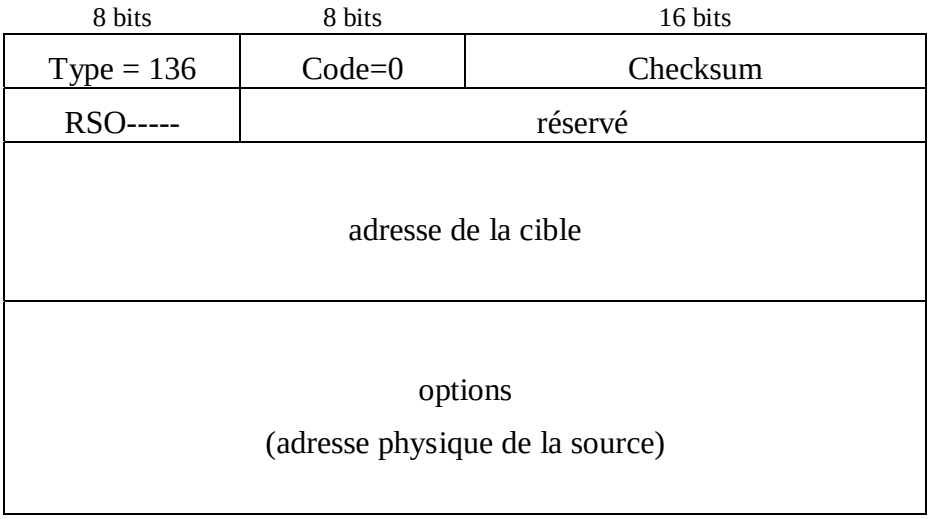


Figure 24 *Format des paquets d'annonce d'un voisin*

- **Indication de redirection**

  La technique de redirection est la même que dans IPv4. Un équipement ne connaît que les préfixes des réseaux auxquels il est directement attaché et l'adresse d'un routeur par défaut. Si la route peut être optimisée, le routeur par défaut envoie ce message pour indiquer qu'une route plus courte existe. En effet, avec IPv6, comme le routeur par défaut est appris

automatiquement, la route n'est pas forcément la meilleure (cf. figure Routage par défaut non optimal).

Un autre cas d'utilisation particulier à IPv6 concerne des stations situées sur un même lien physique mais ayant des préfixes différents. Ces machines passent dans un premier temps par le routeur par défaut. Ce dernier les avertit qu'une route directe existe.

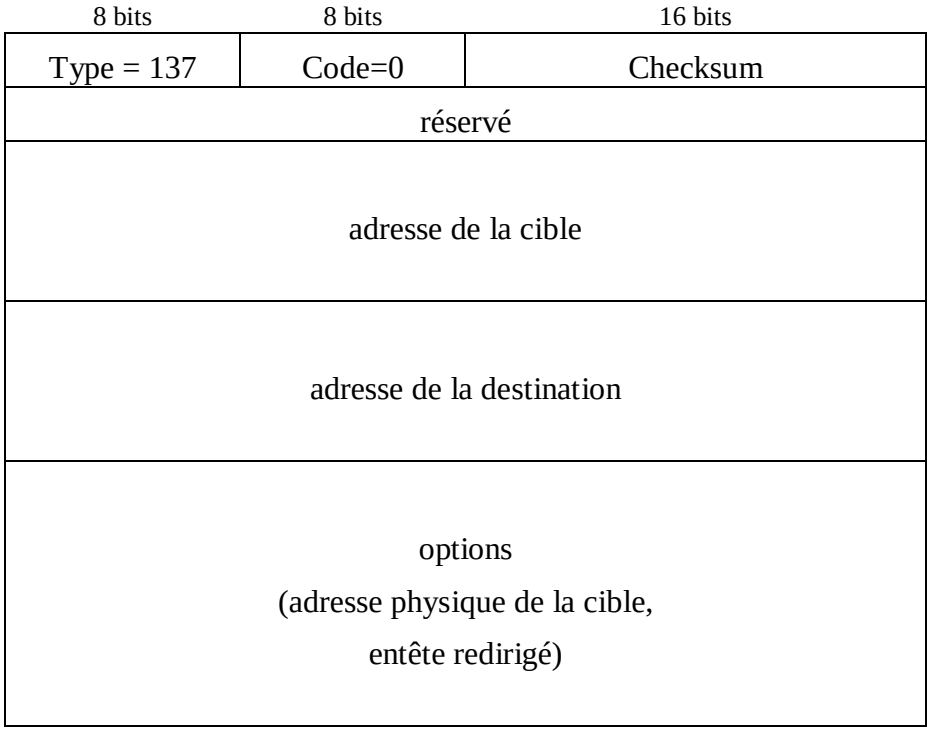


Figure 25 *Format des paquets d'indication de redirection*

6.4 Ping et résolution d'adresses

6.4.1 Les paquets suivants ont été obtenus lors d'un *ping* entre deux machines IPv6 situées sur le même réseau physique de type Ethernet. Avant de pouvoir émettre un paquet IPv6 ICMPv6 de demande d'écho entre deux stations IPv6 situées sur le même réseau physique de type Ethernet, l'émetteur a besoin de connaître l'adresse physique de l'équipement destinataire. Il utilise le protocole de découverte des voisins et émet un message de sollicitation d'un voisin donné par la trace 1. La trace 2 correspond au message d'annonce d'un voisin retourné en réponse.

a. Décoder les deux traces en donnant le maximum d'informations.

Trace 1 :

```
0000: 6f 00 00 00 00 20 3a ff 3f fe 03 02 00 12 00 03

0010: 0a 00 20 ff fe 0a aa 6d ff 02 00 00 00 00 00 00

0020: 00 00 00 01 ff 00 00 03|87 00 4d 7f 00 00 00 00

0030: 3f fe 03 02 00 12 00 03 00 00 00 00 00 00 00 03|

0040: 01 01 08 00 20 0a aa 6d
```

Trace 2 :

```
0000: 6f 00 00 00 00 20 3a ff fe 80 00 00 00 00 00 00

0010: 18 00 20 ff fe 0c 7a 34 3f fe 03 02 00 12 00 03

0020: 0a 00 20 ff fe 0a aa 6d|88 00 d7 fb e0 00 00 00

0030: 3f fe 03 02 00 12 00 03 00 00 00 00 00 00 00 03|

0040: 02 01 1a 00 20 0c 7a 34
```

b. Donnez la valeur des champs « adresse de la source », « adresse de la destination » et « type » de la trame qui encapsule ce paquet. Quelle l'adresse physique (Ethernet) de la source de cette même trame ?

c. Quelle l'adresse IPv6 de la source du paquet 2 ? En vous basant sur l'option utilisée dans le message ICMPv6 encapsulé dans ce même paquet, donnez l'adresse physique (Ethernet) de la source de la trame qui encapsule ce paquet ?

d. L'émetteur envoie alors un premier message ICMPv6 « Demande d'écho » que le destinataire acquitte en retournant un message « Réponse d'écho ». Donner les valeurs des champs d'en-tête des deux paquets IPv6 et des messages encapsulés.

e. Les échanges ICMP Demande d'écho et Réponse d'écho continuent ensuite toutes les secondes. Si les échanges continuent assez longtemps, les deux machines vérifieront périodiquement que le correspondant est toujours correct (il a pu tomber en panne ou être remplacé avec changement d'adresse Ethernet) en utilisant le protocole NUD. Aussi de temps en temps, chaque machine va émettre un message de sollicitation d'un voisin. Une réponse (annonce de voisin avec le bit S) montre que le correspondant est toujours valide. Donner le format du message de sollicitation envoyé par le destinataire.

6.5 Configuration de la route par défaut

En IPv6 seuls les routeurs utilisent des protocoles de routage pour définir leurs tables de routage. Le routage des autres machines repose sur la notion de route par défaut. Comme avec IPv4, l'envoi de messages de redirection est utilisé pour installer de meilleures routes. Périodiquement les routeurs envoient des Annonces du routeur qui permettent aux machines sur le câble de choisir un routeur par défaut, et aussi de calculer leur adresse (dans le mode d'autoconfiguration sans état ou *stateless*).

6.5.1 Un même câble Ethernet relie 3 machines :

- deux routeurs
- et une machine hôte.

Les routeurs émettent périodiquement sur le réseau des messages d'annonce de routeur. Voici la trace de l'un de ces messages.

```
0000: 6f 00 00 00 00 38 3a ff fe 80 00 00 00 00 00 00

0010: 18 00 20 ff fe 0c 7a 34 ff 02 00 00 00 00 00 00

0020: 00 00 00 00 00 00 00 01|86 00 77 3c 00 00 17 70

0030: 00 00 00 00 00 00 00 00|01 01 1a 00 20 0c 7a 34|

0040: 03 04 40 c0 ff ff ff ff ff ff ff ff 00 00 00 00

0050: 3f fe 03 02 00 12 00 03 00 00 00 00 00 00 00 00
```

a. Décoder et interpréter cette trace en donnant le maximum d'informations.

b. Donnez la table de routage de la machine hôte connecté au câble Ethernet après réception de ce message.

7 MECANISME DE DECOUVERTE DU PMTU

Pour des considérations d'efficacité, il est généralement préférable que les informations échangées entre équipements soient contenues dans des datagrammes de taille maximale. Cette taille dépend du chemin suivi par les datagrammes et est égale à la plus grande taille autorisée par l'ensemble des liens traversés. Elle est de ce fait appelée PMTU, ou *Path Maximum Transmission Unit* (unité de transfert de taille maximale sur le chemin).

Initialement, l'équipement émetteur fait l'hypothèse que le PMTU d'un certain chemin est égal au MTU du lien auquel il est directement attaché. S'il s'avère que les paquets transmis sur ce chemin excèdent la taille maximale autorisée par un lien intermédiaire, alors le routeur situé aux abords de ce lien détruit ces paquets et retourne un message d'erreur ICMPv6 de type «paquet trop grand», en y indiquant le MTU accepté. Fort de ces informations, l'équipement émetteur réduit le PMTU supposé pour ce chemin. Plusieurs itérations peuvent être nécessaires avant d'obtenir un PMTU permettant à tout paquet d'arriver à l'équipement destinataire sans jamais excéder le MTU de chaque lien traversé. Le protocole IPv6 garantit que le MTU de tout lien ne peut descendre en dessous de 1 280 octets, valeur qui constitue ainsi une borne inférieure pour le PMTU.

7.1 Comment sont réparées les pertes enregistrées au cours des premières itérations du protocole de découverte du PMTU ?

7.2 Une fois la PMTU déterminée pour un chemin donné, dans quelles situations est-il envisageable de solliciter à nouveau le protocole de découverte du PMTU ?

7.3 Les traces suivantes sont celles des trois paquets échangés lors d'une ouverture de connexion TCP.

Paquet 1

```
0000: 60 00 00 00 00 28 06 40 3f fe 03 02 00 12 00 02

0010: 00 00 00 00 00 00 00 13 3f fe 03 04 01 15 83 00

0020: 02 c0 4f ff fe 61 21 4c|ff ad 13 89 5c 3e 06 6a

0030: 00 00 00 00 a0 02 40 00 9c 2e 00 00 02 04 05 a0

0040: 01 03 03 00 01 01 08 0a 00 9e 7b c4 00 00 00 00
```

Paquet 2

```
0000: 60 00 00 00 00 28 06 3c 3f fe 03 04 01 15 83 00

0010: 02 c0 4f ff fe 61 21 4c 3f fe 03 02 00 12 00 02

0020: 00 00 00 00 00 00 00 13|13 89 ff ad e3 59 9c 1a

0030: 5c 3e 06 6b a0 12 42 f0 14 5a 00 00 02 04 11 3a

0040: 01 03 03 00 01 01 08 0a 00 41 f9 83 00 9e 7b c4
```

Paquet 3

```
0000: 60 00 00 00 00 20 06 40 3f fe 03 02 00 12 00 02
0010: 00 00 00 00 00 00 00 13 3f fe 03 04 01 15 83 00
0020: 02 c0 4f ff fe 61 21 4c|ff ad 13 89 5c 3e 06 6b
0030: e3 59 9c 1b 80 10 43 80 4b 14 00 00 01 01 08 0a
0040: 00 9e 7b c4 00 41 f9 83
```

a. Quelle la taille de segment négociée à l'issue de cette ouverture de connexion TCP ?

Les traces suivantes sont celles des trois premiers paquets échangés sur la connexion précédemment établie :

Paquet 4

```
0000: 60 00 00 00 05 b4 06 40 3f fe 03 02 00 12 00 02
0010: 00 00 00 00 00 00 00 13 3f fe 03 04 01 15 83 00
0020: 02 c0 4f ff fe 61 21 4c|ff ad 13 89 5c 3e 06 6b
0030: e3 59 9c 1b 80 10 43 80 40 a9 00 00 01 01 08 0a
0040: 00 9e 7b c4 00 41 f9 83|20 21 22 23 24 25 26 27
...suite des 1440 octets de données...
```

Paquet 5

```
0000: 60 00 00 00 02 18 3a fe 3f fe 03 02 00 11 00 01
0010: 00 00 00 00 00 00 00 08 3f fe 03 02 00 12 00 02
0020: 00 00 00 00 00 00 00 13|02 00 e8 f7 00 00 05 c8|
0030: 60 00 00 00 05 b4 06 3e 3f fe 03 02 00 12 00 02
0040: 00 00 00 00 00 00 00 13 3f fe 03 04 01 15 83 00
0050: 02 c0 4f ff fe 61 21 4c|ff ad 13 89 5c 3e 06 6b
0060: e3 59 9c 1b 80 10 43 80 40 a9 00 00 01 01 08 0a
0070: 00 9e 7b c4 00 41 f9 83|20 21 22 23 24 25 26 27
suite du paquet IPv6 tronqué à 1280-48 octets ...
```

Paquet 6

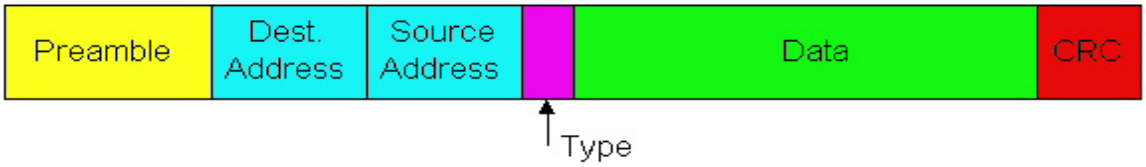
```
0000: 60 00 00 00 05 a0 06 40 3f fe 03 02 00 12 00 02
0010: 00 00 00 00 00 00 00 13 3f fe 03 04 01 15 83 00
0020: 02 c0 4f ff fe 61 21 4c|ff ad 13 89 5c 3e 06 6b
0030: e3 59 9c 1b 80 10 43 80 8b b3 00 00 01 01 08 0a
0040: 00 9e 7b c4 00 41 f9 83|20 21 22 23 24 25 26 27
0050: 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37
suite des 1420 octets de données...
```

b. En vous basant sur la taille du premier paquet de données envoyé sur la connexion TCP, déterminez la valeur de la PMTU initialement utilisée.

c. Décodez la trace du paquet 5. Vous déterminerez quel est l'équipement d'où est issu le paquet IP ainsi que la nouvelle valeur négociée de la PMTU.

Annexes ING

La trame Ethernet



Signification:

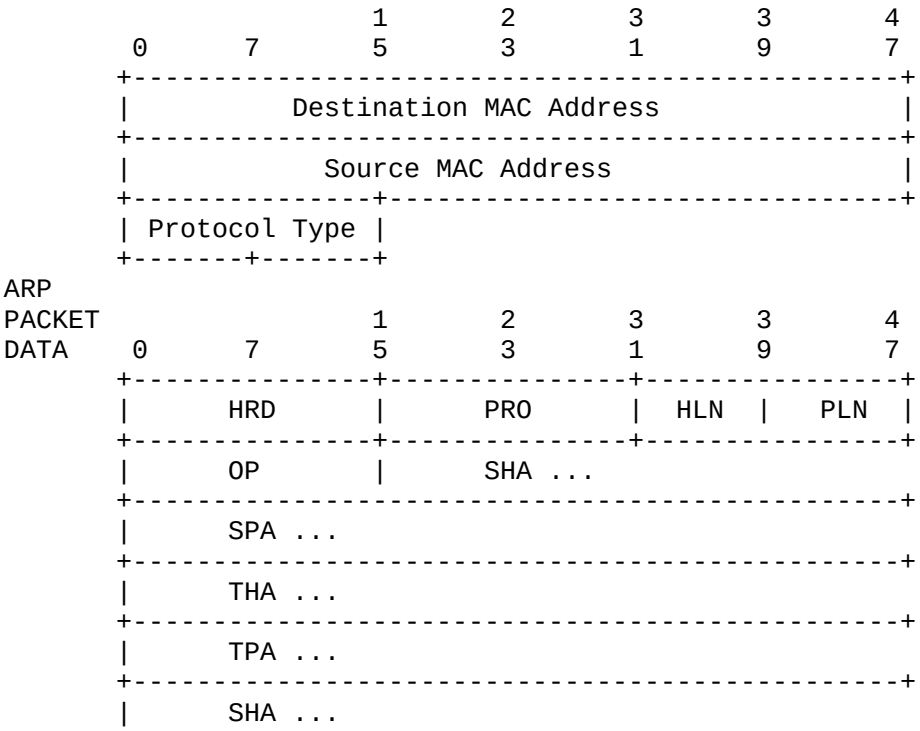
- préambule indique le début de la trame. Sert à la synchronisation trame.
- type définit le type de données encapsulées par la trame.

type (héxa)	utilisation
0200	XEROX PUP
0201	PUP Address Trans.
0600	XEROX NS IDP
0800	DoD Internet
0801	X.75 Internet
0802	NBS Internet
0803	ECMA Internet
0804	ChaosNet
0805	X.25 niveau 3
0806	ARP
0807	XNS
6001 à 6006	DEC
8035	RARP
8098	Appletalk
86DD	IPv6

- CRC (*Cyclic Redundancy Check*) ou checksum sert au contrôle d'intégrité de la trame.

Le paquet ARP/RARP

TRANSMISSION LAYER for 802.x protocols

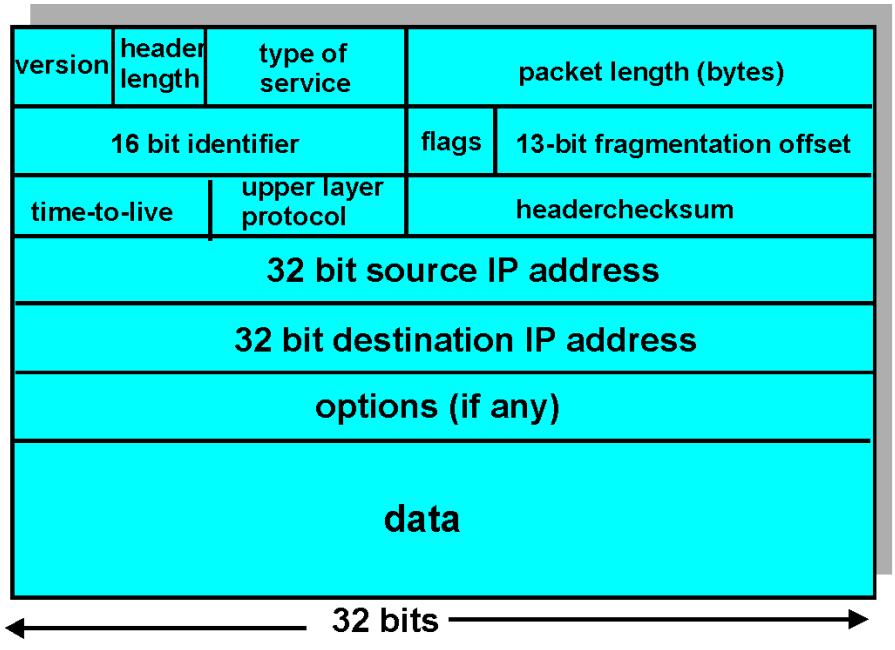


ARP PACKET DATA

```
Ethernet transmission layer
(not necessarily accessible to the user):
  DESTINATION     48.bit: Destination Address
  SOURCE          48.bit: Source Address
  PROTOCOL TYPE   16.bit: Protocol type
                          (set to ARP).

Ethernet packet data:
HRD   16.bit: Hardware address space
              (e.g., Ethernet, Packet Radio Net.)
PRO   16.bit: Protocol address space.
              For Ethernet hardware, this is from
              the set of type fields ether_typ$.
HLN    8.bit: Hardware Address Length (0-255 Bytes)
PLN    8.bit: Protocol Address Length (0-255 Bytes)
OP    16.bit: Opcode: either request or reply
SHA   nbytes: Sender Hardware Address (this packet),
              n from the HLN field.
SPA   mbytes: Sender Protocol Address (this packet),
              m from the PLN field.
THA   nbytes: Target Hardware Address (this packet),
              (if known).
TPA   mbytes: Target Protocol Address
```

Le datagramme IP



Signification:

- IHL (*Internet Header Length*). longueur de l'en-tête exprimé en mots de 32 bits.

- identification donnée par la source. Sert au destinataire en cas de fragmentation.

- TTL. Durée de vie du datagramme. Décrémentée à chaque saut et à la destination chaque seconde en cas de fragmentation.

- Protocol. Protocole émetteur et destinataire des données du datagramme.

Code (déc)	Abréviation	Nom du protocole	Reference
0		Reserved	
1	ICMP	Internet Control Message	[RFC792]
2	IGMP	Internet Group Management	[RFC1112]
3	GGP	Gateway-to-Gateway	[RFC823]
4	IP	IP in IP (encapsulation)	
5	ST	Stream	[RFC1190]
6	TCP	Transmission Control	[RFC793]
7	UCL	UCL	
8	EGP	Exterior Gateway Protocol	[RFC888]
9	IGP	any private interior gateway	
10	BBN-RCC-MON	BBN RCC Monitoring	
11	NVP-II	Network Voice Protocol	[RFC741]
12	PUP	PUP	
13	ARGUS	ARGUS	
14	EMCON	EMCON	
15	XNET	Cross Net Debugger	
16	CHAOS	Chaos	
17	UDP	User Datagram	[RFC768]
[18..35]	...		
36	XTP	XTP	
37	DDP	Datagram Delivery Protocol	
[38..44]	...		
45	IDRP	Inter-Domain Routing Protocol	
46	RSVP	Reservation Protocol	
47	GRE	General Routing Encapsulation	
48	MHRP	Mobile Host Routing Protocol	
[49..53]	...		
54	NHRP	NBMA Next Hop Resolution Protocol	
55-60		Unassigned	
[61..100]	...		
101-254		Unassigned	
255		Reserved	

- header checksum. valeur de contrôle ne portant que sur l’entête.

- options. champ de taille variable. Les options sont codées sur le principe TLV (type, longueur, valeur). La longueur indique la taille complète de l'option en octet.

Type (déc)	Nom	Reference
0 EOOL	- End of Options List	[RFC791]
1 NOP	- No Operation	[RFC791]
130 SEC	- Security	[RFC1108]
131 LSR	- Loose Source Route	[RFC791]
68 TS	- Time Stamp	[RFC791]
133 E-SEC	- Extended Security	[RFC1108]
7 RR	- Record Route	[RFC791]
136 SID	- Stream ID	[RFC791]
137 SSR	- Strict Source Route	[RFC791]

Structure de l'option d'enregistrement de route (Record Route)

```
Class  Number  Length
 0       7       var.   routeur address
```

- padding. octets de bourrage afin d'aligner l'en-tête sur un nombre entier de mots de 32 bits.
-

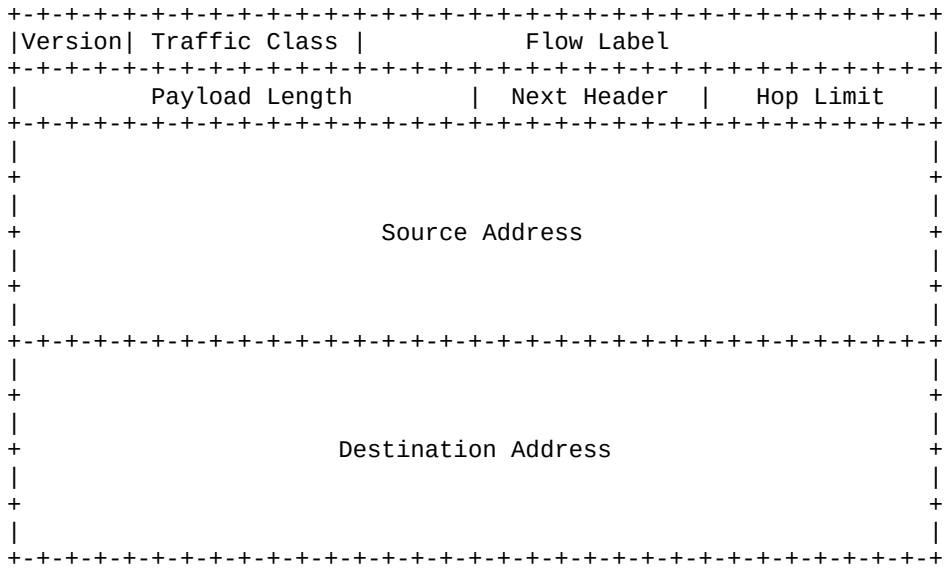
Le message ICMP

type (déc)	signification du message
8	*Echo Request* demande d'écho
0	*Echo Reply* réponse en écho
11	*Time Exceeded for a Datagram* temps de vie d'un datagramme dépassé
12	*Parameter Problem on a Datagram* datagramme mal formé
3	*Destination Unreachable* destination inaccessible
5	*Redirect* redirection, changement de route
4	*Source Quench* interruption de la source
13	*Timestamp Request* demande de date d'estampillage
14	*Timestamp Reply* réponse à une demande d'estampillage
15	*Information Request* demande d'information
16	*Information Reply* réponse à une demande d'information
17	*Address Mask Request* demande de masque d'adresse
18	*Address Mask Reply* réponse à une demande de masque d'adresse

Messages ICMP

ICMP type	code	description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Format datagramme IPv6



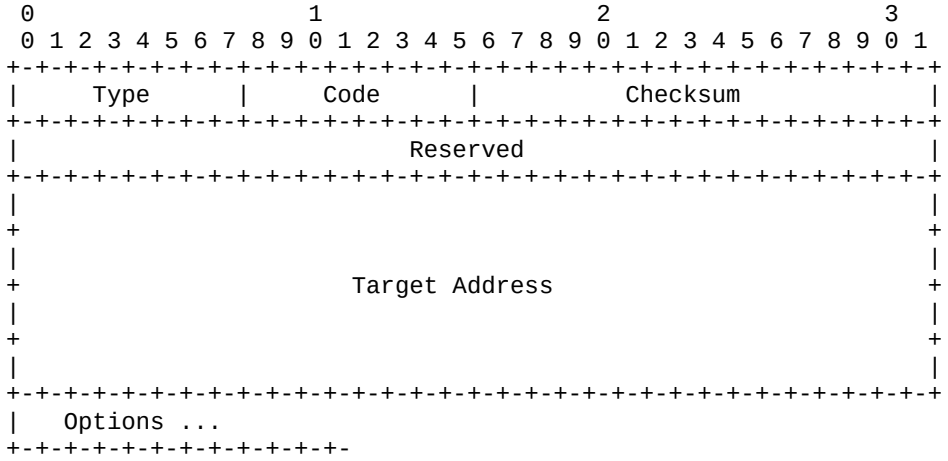
Valeur du Next Header:

```
00      Extension hop by hop
43      Extension Routage
44      Extension Fragmentation
60      Extension Destination
06      TCP
17      UDP
41      IPv6
58      ICMPv6
```

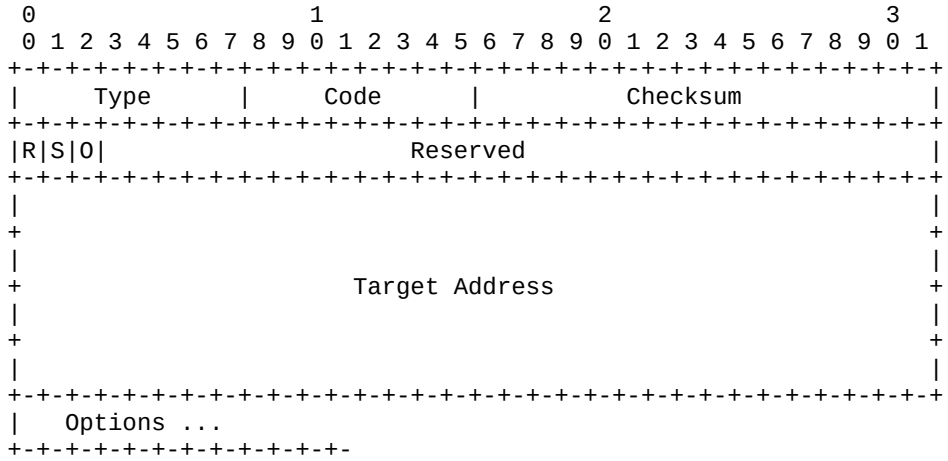
ICMPv6

Type	Nature
128	Demande d'écho
129	Réponse d'écho
133	Sollicitation du routeur
134	Annonce du routeur
135	Sollicitation d'un voisin
136	Annonce d'un voisin
137	Redirection

Solicitation d'un voisin



Annonce d'un voisin



Flag:

R (Router flag): Vaut 1 quand l'émetteur est un routeur

S (Solicited flag): Vaut 1 quand le message d'annonce est envoyé en réponse à un message de solicitation

O (Override flag): vaut 1 pour indiquer la mise à jour de l'adresse physique du cache.

Format d'option

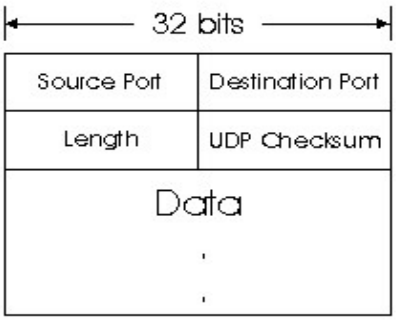
Source/Target Link-layer Address

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |    Length     |    Link-Layer Address ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

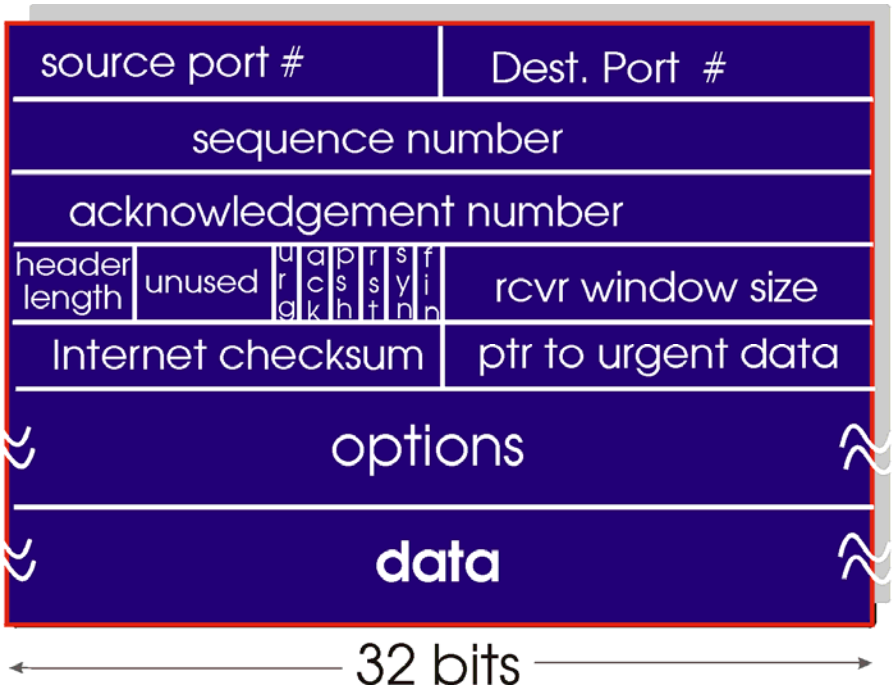
Le champ length inclus lui-même et le champ type. Il s'exprime en unité de 8 octets. Par exemple, la longueur d'une adresse IEEE 802 vaut 1.

Option Name	Type
Source Link-Layer Address	1
Target Link-Layer Address	2
Prefix Information	3
Redirected Header	4
MTU	5

Le datagramme UDP



Le segment TCP



Numéro des ports "well known"

La liste complète est disponible à l'URL:
`ftp://ftp.isi.edu/in-notes/iana/assignments/port-numbers`.

```
Port Assignments:
Keyword         Decimal    Description
References
-------         -------    -----------                        ---------
-
                  0/tcp    Reserved
                  0/udp    Reserved
[...]
daytime          13/tcp    Daytime
daytime          13/udp    Daytime
[...]
ftp-data         20/tcp    File Transfer [Default Data]
ftp-data         20/udp    File Transfer [Default Data]
ftp              21/tcp    File Transfer [Control]
ftp              21/udp    File Transfer [Control]
#                22/tcp    Unassigned
#                22/udp    Unassigned
telnet           23/tcp    Telnet
telnet           23/udp    Telnet
[...]
smtp             25/tcp    Simple Mail Transfer
smtp             25/udp    Simple Mail Transfer
[...]
nameserver       42/tcp    Host Name Server
nameserver       42/udp    Host Name Server
nicname          43/tcp    Who Is
nicname          43/udp    Who Is
[...]
domain           53/tcp    Domain Name Server
domain           53/udp    Domain Name Server
[...]
tftp             69/tcp    Trivial File Transfer
tftp             69/udp    Trivial File Transfer
#                          David Clark <ddc@LCS.MIT.EDU>
gopher           70/tcp    Gopher
gopher           70/udp    Gopher
[...]
www-http         80/tcp    World Wide Web HTTP
www-http         80/udp    World Wide Web HTTP
[...]
nntp            119/tcp    Network News Transfer Protocol
nntp            119/udp    Network News Transfer Protocol
[...]
ntp             123/tcp    Network Time Protocol
ntp             123/udp    Network Time Protocol
[...]
snmp            161/tcp    SNMP
snmp            161/udp    SNMP
snmptrap        162/tcp    SNMPTRAP
snmptrap        162/udp    SNMPTRAP
```