

IP, fragmentation, routage, commutation

Avec Correction

1. Fragmentation des paquets IP

1.1 Soit un réseau constitué de 5 routeurs IP (R1 ... R5) et de trois stations A, B et C qui doivent communiquer (Figure 1).

Chaque liaison entre hôtes (station ou routeur) est étiquetée par son MTU (Maximum Transmission Unit). Le MTU définit la taille maximale d'un paquet IP qui peut être véhiculé dans les trames d'un réseau physique particulier. Ce paramètre est rattaché à une interface réseau du hôte (de numéro p_i et d'adresse IP de classe B) pour fragmenter les données avant leur transmission sur la liaison.

On suppose que A doit émettre 1520 octets de données vers B.

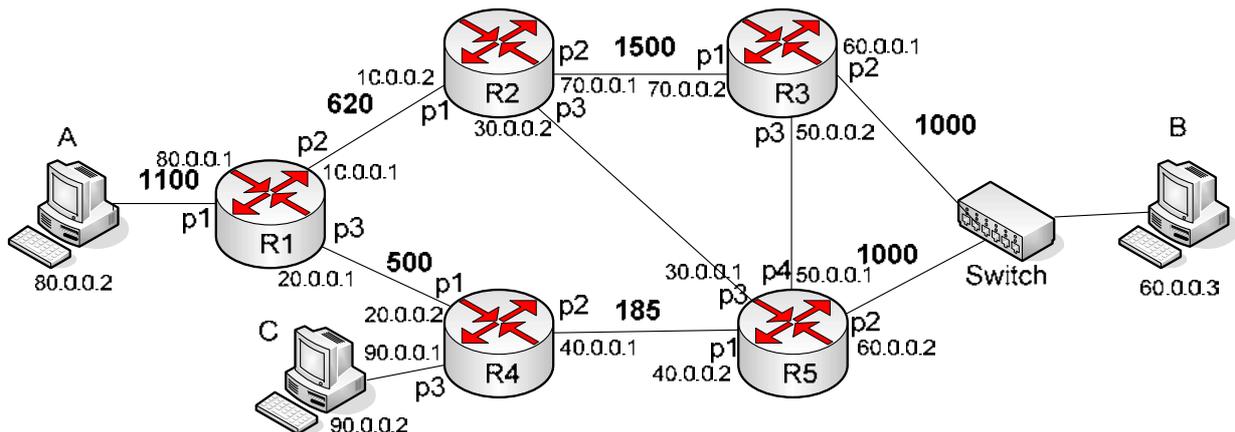
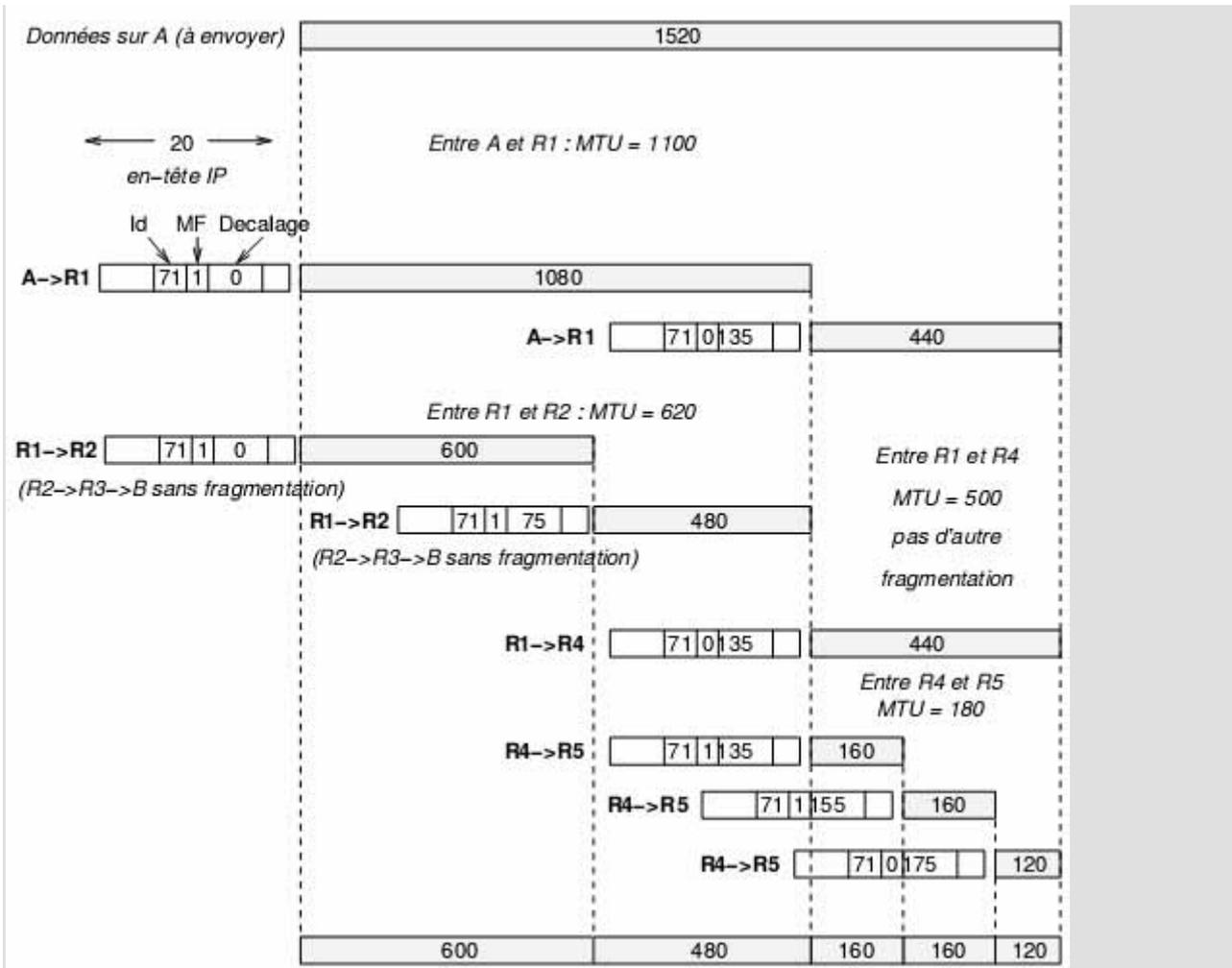


Figure 1 – Architecture de réseau

1.1 Décrivez les fragmentations réalisées pour la transmission d'un paquet IP émis par A à destination de B, en supposant que le routeur R1 transmet alternativement les trames qu'il reçoit vers R2-R3 puis vers R4-R5. Précisez pour chaque fragment de paquet, les valeurs des champs (Identification, More Fragment Flag, Offset). On supposera que la valeur initiale de l'Identifiant du paquet est 71.



1.2 Justifiez le fait que la régénération des paquets fragmentés n'a lieu que sur la station destinataire et non sur les routeurs.

Certains routeurs ne voient pas passer la totalité des fragments d'un paquet IP, ils ne peuvent donc pas reconstruire le datagramme original. De plus, l'opération de défragmentation nécessite de la mémoire et du calcul. La philosophie de IP est de simplifier les fonctions dans le réseau (c'est-à-dire exécutées par les routeurs) et de repousser les traitements complexes (contrôle de flux, contrôles d'erreurs, chiffrement, défragmentation, etc ...) aux extrémités du réseau (c'est-à-dire exécutés par les stations).

1.3 Afin d'éviter la fragmentation, la notion de **Path-MTU** (MTU sur un chemin) a été introduit. Il représente la valeur minimale des MTU des réseaux traversés. Proposer une solution pour trouver cette valeur.

Une solution, proposée par exemple dans IPv6, est de placer le bit « Don't fragment » à la valeur 1 dans l'en-tête des paquets IP émis vers la destination (champ FLAG). Si le paquet arrive sur un routeur qui doit le fragmenter parce que le MTU de l'interface de sortie est trop petite, alors il sera détruit et le routeur enverra un message ICMP (Destination Unreachable) à la station émettrice. Une extension du protocole ICMP pour IPv6 (ICMPv6) propose d'indiquer dans ce message d'erreur ICMP la taille du MTU requis. Dans la pratique seule les stations en IPv6 exploitent cette extension. Les émetteurs IPv4 choisissent de réduire la taille des paquets émis arbitrairement et à l'aveugle, jusqu'à trouver une taille de paquet qui est inférieure ou égale à la valeur Path-MTU.

2. Routage des paquets IP

Soit le réseau de la Figure 1 (question 1 ci-dessus). A chaque liaison, supposée symétrique, est associée une distance égale à 1. On supposera que les routeurs mettent en œuvre un protocole de routage de type vecteur de distance avec l'algorithme Bellman-Ford.

2.1 Quelle est la longueur maximale d'une route dans un réseau IP utilisant le protocole de routage RIP ? Comment s'assurer qu'un paquet IP ne ?

La route la plus longue traverse au maximum 16 routeurs. Les paquets IP transmis par une source sont initialisés avec le champ TTL (Time To Live) égal à 16. A chaque passage dans un routeur, le champ TTL du paquet est décrémenté de 1. Lorsqu'un paquet arrive dans un routeur avec une valeur TTL égale à 1, le routeur détruit ce paquet, limitant les routes à 16 nœuds traversés au maximum.

2.2 On supposera que le réseau vient d'être mis en service par l'administrateur et que chaque routeur n'a qu'une connaissance locale de la topologie du réseau (il ne connaît que ses routeurs voisins et ses sous-réseaux voisins). Donner les tables de routage initiales des différents routeurs telles que configurées par l'administrateur, en suivant le format de table ci-dessous.

Adresse IP du réseau destination	adresse IP du prochain routeur	numéro de l'interface	Métrique (Hop Count) nbre de sauts

A T=0

R1

Dest. network	Next hop	Interf .	Metric (HC)
10.0.0.0	direct	P2	0
20.0.0.0	direct	P3	0
90.0.0.0	direct	P1	0

R2

Dest. network	Next hop	Interf .	metric
10.0.0.0	direct	P1	0
30.0.0.0	direct	P3	0
70.0.0.0	direct	P2	0

R3

Dest. network	Next hop	Interf .	metric
50.0.0.0	direct	P3	0
60.0.0.0	direct	P2	0
70.0.0.0	direct	P1	0

R4

Dest. network	Next hop	Interf .	metric
20.0.0.0	direct	P1	0
40.0.0.0	direct	P2	0

90.0.0.0	direct	P3	0
----------	--------	----	---

R5

Dest. network	Next hop	Interf	metric
30.0.0.0	direct	P3	0
40.0.0.0	direct	P1	0
50.0.0.0	direct	P4	0
60.0.0.0	direct	P2	0

2.3 On considèrera la séquence d'échange de vecteurs de distance suivante:

<i>Instant</i>	<i>Evénement</i>
T ₁	R2, R4 reçoivent VR1 (vecteur de distance de R1)
T ₂	R1, R3, R5 reçoivent VR2
T ₃	R1, R5 reçoivent VR4
T ₄	R2, R4 reçoivent VR1 puis VR5
T ₅	R2, R5 reçoivent VR3
T ₆	R1 reçoit VR2
T ₇	R3, R4 reçoivent VR5

Donnez la table de routage de chaque routeur, obtenue une fois que l'algorithme de routage a convergé.

R2

Network	Next hop	Int.	metric
10.0.0.0	direct	P1	0
30.0.0.0	direct	P3	0
70.0.0.0	direct	P2	0
80.0.0.0	10.0.0.1	P1	1
20.0.0.0	10.0.0.1	P1	1

R4

Network	Next hop	Int.	metric
20.0.0.0	direct	P1	0
40.0.0.0	direct	P2	0
90.0.0.0	direct	P3	0
10.0.0.0	20.0.0.1	P1	1
80.0.0.0	20.0.0.1	P1	1

À T₂, R1, R3, R5 reçoivent VR2 = ([10.0.0.0, 0], [20.0.0.0, 1], [30.0.0.0, 0], ([70.0.0.0, 0], [80.0.0.0, 1])

R1

Network	Next hop	Int.	metric
10.0.0.0	direct	P2	0
20.0.0.0	direct	P3	0
80.0.0.0	direct	P1	0
30.0.0.0	10.0.0.2	P2	1
70.0.0.0	10.0.0.2	P2	1

R3

Network	Next hop	Int.	metric
50.0.0.0	direct	P3	0
60.0.0.0	direct	P2	0
70.0.0.0	direct	P1	0

10.0.0.0	70.0.0.1	P1	1
20.0.0.0	70.0.0.1	P1	2
30.0.0.0	70.0.0.1	P1	1
80.0.0.0	70.0.0.1	P1	2

R5

Network	Next hop	Int.	metric
30.0.0.0	direct	P3	0
40.0.0.0	direct	P1	0
50.0.0.0	direct	P4	0
60.0.0.0	direct	P2	0
10.0.0.0	30.0.0.2	P3	1
70.0.0.0	30.0.0.2	P3	1
20.0.0.0	30.0.0.2	P3	2
80.0.0.0	30.0.0.2	P3	2

À T_3 = R1 et R5 reçoivent VR4 = ([10, 0] ; [20, 1] ; [30, 0] ; [[70, 0] ; [80, 1]])

R1

Network	Next hop	Int.	metric
10.0.0.0	direct	P2	0
20.0.0.0	direct	P3	0
80.0.0.0	direct	P1	0
30.0.0.0	10.0.0.2	P2	1
70.0.0.0	10.0.0.2	P2	1
40.0.0.0	20.0.0.2	P3	1
90.0.0.0	20.0.0.2	P3	1

R5

Network	Next hop	Int.	metric
30.0.0.0	direct	P3	0
40.0.0.0	direct	P1	0
50.0.0.0	direct	P4	0
60.0.0.0	direct	P2	0
10.0.0.0	30.0.0.2	P3	1
70.0.0.0	30.0.0.2	P3	1
20.0.0.0	30.0.0.2	P3	2
80.0.0.0	30.0.0.2	P3	2
80.0.0.0	40.0.0.1	P1	1
90.0.0.0	40.0.0.1	P1	1

La route vers 80.0.0.0 et reçu de R4 (via son interface d'adresse 40.0.0.1) est plus courte que celle enregistrée dans la table de R5. Par conséquent, celle-ci remplace l'ancienne avec un coût de 1 (nombre de sauts pour atteindre le réseau 80.0.0.0).

À T_4 = R2 et R4 reçoivent VR1 puis VR5

VR1 = ([10, 0] ; [20, 0] ; [30, 1] ; [[40, 1] ; [70, 1] ; [80, 0] ; [90, 1]])

VR5 = ([10, 1] ; [20, 1] ; [30, 0] ; [40, 0] ; [50, 0] ; [60, 0] ; [70, 1] ; [80, 2] ; [90, 1])

R2

Network	Next hop	Int.	metric
10.0.0.0	direct	P1	0
30.0.0.0	direct	P3	0
70.0.0.0	direct	P2	0
80.0.0.0	10.0.0.1	P1	1
20.0.0.0	10.0.0.1	P1	1
40.0.0.0	10.0.0.1	P1	2

90	10.0.0.1	P1	2
40	30.0.0.1	P3	1
50	30.0.0.1	P3	1
60	30.0.0.1	P3	1

R4

Network	Next hop	Int.	metric
20.0.0.0	direct	P1	0
40.0.0.0	direct	P2	0
90.0.0.0	direct	P3	0
10.0.0.0	20.0.0.1	P1	1
80.0.0.0	20.0.0.1	P1	1
30.0.0.0	20.0.0.1	P1	2
70.0.0.0	20.0.0.1	P1	2
30.0.0.0	40.0.0.2	P2	1
50.0.0.0	40.0.0.2	P2	1
60.0.0.0	40.0.0.2	P2	1

À T₅ = R2 et R5 reçoivent VR3 = ([10, 1] ; [20, 2] ; [30, 1] ; [50, 0] ; [60, 0] ; [70, 0], [80, 2])

R2

Network	Next hop	Int.	metric
10.0.0.0	direct	P1	0
30.0.0.0	direct	P3	0
70.0.0.0	direct	P2	0
80.0.0.0	10.0.0.1	P1	1
20.0.0.0	10.0.0.1	P1	1
90	10.0.0.1	P1	2
40	30.0.0.1	P3	1
50	30.0.0.1	P3	1
60	30.0.0.1	P3	1

R5

Network	Next hop	Int.	metric
30.0.0.0	direct	P3	0
40.0.0.0	direct	P1	0
50.0.0.0	direct	P4	0
60.0.0.0	direct	P2	0
10.0.0.0	30.0.0.2	P3	1
70.0.0.0	30.0.0.2	P3	1
20.0.0.0	30.0.0.2	P3	2
80.0.0.0	40.0.0.1	P1	1
90.0.0.0	40.0.0.1	P1	1

Pas de modification des tables de R2 et R5 à T

À T₆ = R1 reçoit VR2 = ([10, 0] ; [20, 1] ; [30, 0] ; [40, 1] ; [50, 1] ; [60, 1] ; [70, 0], [80, 1] ; [90, 2])

R1

Network	Next hop	Int.	metric
10.0.0.0	direct	P2	0
20.0.0.0	direct	P3	0
80.0.0.0	direct	P1	0
30.0.0.0	10.0.0.2	P2	1
70.0.0.0	10.0.0.2	P2	1
40.0.0.0	20.0.0.2	P3	1

90.0.0.0	20.0.0.2	P3	1
50.0.0.0	10.0.0.2	P2	2
60.0.0.0	10.0.0.2	P2	2

À T₇ = R3 et R4 reçoivent VR5 = ([10, 1] ; [20, 1] ; [30, 0] ; [40, 0] ; [50, 0] ; [60, 0] ; [70, 1], [80, 2] ; [90, 1])

R3

Network	Next hop	Int.	metric
50.0.0.0	direct	P3	0
60.0.0.0	direct	P2	0
70.0.0.0	direct	P1	0
10.0.0.0	70.0.0.1	P1	1
20.0.0.0	70.0.0.1	P1	2
30.0.0.0	70.0.0.1	P1	1
80.0.0.0	70.0.0.1	P1	2
40.0.0.0	50.0.0.1	P3	1
90.0.0.0	50.0.0.1	P3	2

R4

Network	Next hop	Int.	metric
20.0.0.0	direct	P1	0
40.0.0.0	direct	P2	0
90.0.0.0	direct	P3	0
10.0.0.0	20.0.0.1	P1	1
80.0.0.0	20.0.0.1	P1	1
70.0.0.0	20.0.0.1	P1	2
30.0.0.0	40.0.0.2	P2	1
50.0.0.0	40.0.0.2	P2	1
60.0.0.0	40.0.0.2	P2	1

Pas de changement de la table de R4.

A la fin des 7 étapes, les tables de routage sont les suivantes :

R1

Network	Next hop	Int.	metric
10.0.0.0	direct	P2	0
20.0.0.0	direct	P3	0
30.0.0.0	10.0.0.2	P2	1
40.0.0.0	20.0.0.2	P3	1
50.0.0.0	10.0.0.2	P2	2
60.0.0.0	10.0.0.2	P2	2
70.0.0.0	10.0.0.2	P2	1
80.0.0.0	direct	P1	0
90.0.0.0	20.0.0.2	P3	1

R2

Network	Next hop	Int.	metric
10.0.0.0	direct	P1	0
20.0.0.0	10.0.0.1	P1	1
30.0.0.0	direct	P3	0
40.0.0.0	30.0.0.1	P3	1
50.0.0.0	30.0.0.1	P3	1
60.0.0.0	30.0.0.1	P3	1
70.0.0.0	direct	P2	0

80.0.0.0	10.0.0.1	P1	1
90.0.0.0	10.0.0.1	P1	2

R3

Network	Next hop	Int.	metric
10.0.0.0	70.0.0.1	P1	1
20.0.0.0	70.0.0.1	P1	2
30.0.0.0	70.0.0.1	P1	1
40.0.0.0	50.0.0.1	P3	1
50.0.0.0	direct	P3	0
60.0.0.0	direct	P2	0
70.0.0.0	direct	P1	0
80.0.0.0	70.0.0.1	P1	2
90.0.0.0	50.0.0.1	P3	2

R4

Network	Next hop	Int.	metric
10.0.0.0	20.0.0.1	P1	1
20.0.0.0	direct	P1	0
30.0.0.0	40.0.0.2	P2	1
40.0.0.0	direct	P2	0
50.0.0.0	40.0.0.2	P2	1
60.0.0.0	40.0.0.2	P2	1
70.0.0.0	20.0.0.1	P1	2
80.0.0.0	20.0.0.1	P1	1
90.0.0.0	direct	P3	0

R5

Network	Next hop	Int.	metric
10.0.0.0	30.0.0.2	P3	1
20.0.0.0	30.0.0.2	P3	2
30.0.0.0	direct	P3	0
40.0.0.0	direct	P1	0
50.0.0.0	direct	P4	0
60.0.0.0	direct	P2	0
70.0.0.0	30.0.0.2	P3	1
80.0.0.0	40.0.0.1	P1	1
90.0.0.0	40.0.0.1	P1	1

2.4 Pourquoi et au bout de combien de temps une route est elle considérée comme non valide puis supprimée de la table d'un routeur ?.

Toute les 30 secondes, le daemon RIP (routed) qui s'exécute sur le routeur, transmet sur le port 520 (UDP) sa table de routage vers tous ces routeurs voisins, sous la forme de paquets IP broadcast (255.255.255.255). Si un routeur voisin ne répond pas au bout de 180 sec (3 mns), toutes les routes passant par ce routeur sont invalidées avec un coût mis à 16 (infini). Ces routes sont conservées durant encore 120 sec (2 mns), puis supprimées de la table si aucun paquet RIP n'est reçu de ce routeur.

2.5 Travaux pratiques : RIP

2.5.1 - Visualiser la table de routage sur le serveur et sur le client, avec la commande : **route**

La commande **route**, tout comme **ifconfig** sert à la fois à connaître l'état de la table de routage de l'hôte et à configurer de nouvelles routes statiques au besoin.

Exemple de table de routage sous windows avec la commande système (C:\route print)

Description	Destination réseau	du	Masque réseau	de	Passerelle	Interface	Métrique
Itinéraire par défaut	0.0.0.0		0.0.0.0		10.0.0.1	10.0.0.169	1
Réseau de bouclage	127.0.0.0		255.0.0.0		127.0.0.1	127.0.0.1	1
Réseau local	10.0.0.0		255.0.0.0		10.0.0.169	10.0.0.169	1
Adresse IP locale	10.0.0.169		255.255.255.255		127.0.0.1	127.0.0.1	1
Adresses multicast	224.0.0.0		240.0.0.0		10.0.0.169	10.0.0.169	1
Adresse de diffusion limitée	255.255.255.255		255.255.255.255		10.0.0.169	10.0.0.169	1

Exemple de table de routage sous Linux/UNIX avec la commande système (route)

```
# route -n
Kernel IP routing table
Destination      Gateway          Genmask         Flags Metric Ref    Use Iface
192.168.1.0      0.0.0.0         255.255.255.0  U      0      0      0 eth0
0.0.0.0          192.168.1.1    0.0.0.0        UG     0      0      0 eth0
```

2.5.2 – Sur le serveur ajouter une route vers le client au moyen de la commande ci-dessous, et visualiser la nouvelle table:

route add -net 192.168.119.0 netmask 255.255.255.0 vmnet1

- Lancer l'analyseur de protocole Wireshark sur le serveur. Puis lancez le daemon RIP « **routed** » sur le serveur grâce à la commande ci-dessous. Analyser les paquets RIP transmis par le serveur:

```
[prof@machine] $/etc/init.d/routed start.
```

3. Commutation de paquets IP

Pour accélérer les transferts de flux multimédia (téléphonie, audio et vidéo) entre les stations du réseau de la figure 1, on décide de combiner le routage IP avec **la commutation de circuit virtuel**. Ces nouveaux équipements, appelés commutateurs IP, sont capables d'établir et de gérer des circuits virtuels entre les stations en insérant un nouveau champ appelé **Label** ou **étiquette** devant chaque en-tête des paquets IP émis vers la destination.

On supposera aussi qu'un paquet d'appel (**setup**) est transmis avant les données multimédia pour initialiser le circuit virtuel. Ce paquet d'appel est acquitté par la destination et est routé au moyen des tables de routage RIP précédemment calculées.

3.1 – Citer les différentes méthodes de commutation que vous connaissez.

- commutation de circuit

- commutation de message
- commutation de paquet
- commutation de cellule

3.1 – Qu'est ce que la commutation de circuit virtuel ?

Une communication par commutation de circuit virtuel est fondée sur la négociation et la construction d'un chemin unique d'une machine A à une machine B, lors de l'établissement d'une séquence de dialogue entre ces deux machines. Le chemin ainsi créé perdure jusqu'à la clôture de la séquence de dialogue qu'il sous-tend. Ce chemin physique étant partagé entre plusieurs communications, il est appelé chemin logique ou circuit virtuel, d'où le nom de cette méthode de communication.

3.2 – quels sont les avantages d'utiliser un service de transfert de données par commutation de circuit virtuel, plutôt qu'un service en mode datagramme ?

- le séquençement des données reçues,
- la négociation préalable des ressources de communication (mémoire, bande passante, CPU, fenêtre de contrôle de flux, temporisateurs, ...) nécessaires au bon transfert des données ;
- vérification de la disponibilité des machines communicantes
- la minimisation du délai de traitement (temps de service) des paquets dans chaque nœud.

3.2 On supposera que 2 circuits virtuels CV1 et CV2 sont établis entre A et B. et un 3^{ème} CV (CV3) est établi entre C et B. Les CV sont activées dans l'ordre chronologique suivant : CV1, CV3, puis CV2. Calculer les tables de commutation des labels des commutateurs appartenant aux circuits virtuels en utilisant le format de table ci-dessous.

R1			
Entrée		Sortie	
Label	Interface	Interface	Label

L'initialisation du circuit virtuel CV1 est réalisée par l'envoi d'un premier paquet d'appel (call setup) par la machine A vers la destination B.

- Dans chaque nœud traversé, l'adresse IP de la machine destination B contenu dans ce paquet d'appel est analysée pour identifier la route à suivre.
- A la réception de ce paquet d'appel, les nœuds traversés consultent leur table de routage IP, identifie le prochain nœud vers la destination, alloue un numéro de circuit virtuel (appelé étiquette ou label) à cette échange et mettent à jour leur table de commutation de circuit.
- Les variables suivantes sont enregistrées dans la table de commutation: le n° de port d'entrée, le n° de CV en entrée, le n° de port de sortie. La dernière variable de la table à savoir le « n° de CV de sortie » sera déterminé par le nœud suivant et communiqué au nœud pour assurer son unicité et permettre le bon démultiplexage des paquets associés à chaque dialogue.
- Ce premier paquet d'appel est ainsi routé jusqu'à la destination et donnera lieu à un acquittement.

Initialisation du circuit virtuel CV1 :

1. Comme R1 est la passerelle par défaut de A, A transmet un message d'appel à R1 en précisant l'adresse IP de la machine B destinataire.
2. La table de routage de R1, indique que pour atteindre la machine B d'adresse 60.0.0.3 appartenant au sous-réseau 60.0.0.0, il faut transiter via l'interface d'adresse 10.0.0.2, qui appartient au nœud R2. Pour atteindre le prochain nœud R2, il faut sortir de R1 par son port n° P2.
3. R1 alloue le premier n° de circuit virtuel disponible, à savoir 1, et le communique à la machine A.
4. R1 initialise sa table de commutation avec le vecteur (n° de circuit virtuel en entrée = 1, n° de port d'entrée = P1 ; n° de port de sortie = P2, n° de CV en sortie = à déterminer par le prochain nœud R2).

R1			
Entrée		Sortie	
Label	Interface	Interface	Label
1	P1	P2	?

5. R1 transmet le paquet d'appel au prochain nœud R2. R2 attribue le premier n° de CV disponible à cette connexion et le transmet au nœud R1. R1 finalise l'initialisation de sa table de commutation.

R1			
Entrée		Sortie	
Label	Interface	Interface	Label
1	P1	P2	1

- R2 effectue les mêmes opérations que R1 (étapes 2 à 5). La table de routage de R2, indique que pour atteindre la machine B d'adresse 60.0.0.3, il faut passer par le prochain nœud R5 en sortant par le port P3.
- La table de commutation de R2 est alors initialisée comme suit :

R2			
Entrée		Sortie	
Label	Interface	Interface	Label
1	P3	P3	1

- R5 effectue les mêmes opérations que R1 et R2 (étapes 2 à 5). La table de routage de R5, indique que le sous réseau de B est directement raccordé au nœud via le port P2.
- La table de commutation de R5 est alors initialisée comme suit :

R5			
Entrée		Sortie	
Label	Interface	Interface	Label
1	P3	P2	1

Initialisation du circuit virtuel CV3 :

- La procédure d'initialisation est identique à celle suivie pour le CV1 (étapes 1 à 5), et permet d'identifier les nœuds traversés par le CV3 entre les machines C et B, soit R4 et R5. Les tables de commutation de R4 et R5 sont mise à jour comme suit. Sachant que pour R5, le premier « n° de CV de sortie » disponible est alors 2.

R4			
Entrée		Sortie	
Label	Interface	Interface	Label
1	P3	P2	1

R5			
Entrée		Sortie	
Label	Interface	Interface	Label
1	P3	P2	1
1	P1	P2	2

Initialisation du circuit virtuel CV2 :

- comme pour le CV1, le CV2 traverse les nœuds R1 puis R2 puis R5. Leur table de commutation est initialisée comme suit :

R1

Entrée		Sortie	
Label	Interface	Interface	Label
1	P1	P2	1
2	P1	P2	2

R2

Entrée		Sortie	
Label	Interface	Interface	Label
1	P3	P3	1
2	P3	P3	2

R5

Entrée		Sortie	
Label	Interface	Interface	Label
1	P3	P2	1
1	P1	P2	2
2	P3	P2	3