

# Network Architecture / M1-RID: Fragmentation

Ali Benzerbadj

Ain Temouchent University Belhadj Bouchaïb (ATU-2B)

21 avril 2021

# Plan

## Remarque

- La couche 3 va devoir découper les datagrammes trop gros par rapport à la MTU imposée par la couche 2.
- Il nous faut encore découvrir comment ces fragments vont être identifiés, et comment une machine qui va recevoir ces fragments va pouvoir reconstituer les réassembler.
- Cette information (MTU -Maximum Transmission Unit-) doit être donnée par la couche 2, car c'est elle qui va encapsuler le datagramme IP et qui peut donc dire quelle taille de datagramme elle peut transporter.

## Remarque

- La couche 2 va donc dire à la couche 3 quelle est la taille qu'elle peut accepter comme datagramme, et la couche 3 va faire en sorte de découper les datagrammes trop gros à cette taille. Par exemple, pour le protocole Ethernet, la MTU est de 1500 octets.

## Remarque

- Le datagramme est l'information de couche 3 au moment où la machine qui émet une information reçoit les informations de la couche 4.
- Un paquet est l'élément de couche 3 qui circule sur le réseau. Il peut être un datagramme complet s'il n'a pas été découpé, ou un des morceaux de datagrammes fragmentés.
- Un fragment est comme un paquet, sauf qu'on utilise souvent ce terme quand on parle de fragmentation.

# Fragmentation Datagramme IPv4

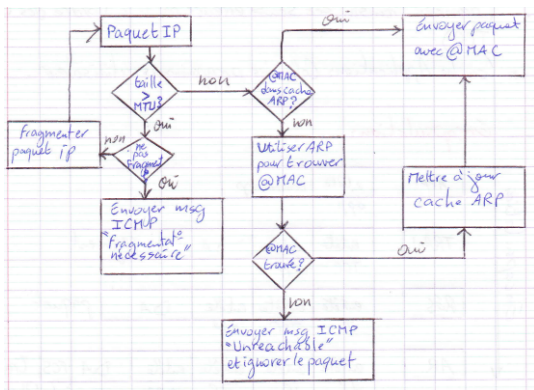


Figure 1 – Transfert d'un paquet IP d'un hôte H1 vers un hôte H2.

## Remarque très importante

- La taille des données d'un paquet IP doit obligatoirement être un multiple de 8.
- Le dernier paquet provenant d'un datagramme fragmenté n'a pas besoin d'avoir une taille de données multiple de 8 car il n'y aura aucun fragment après lui et donc de besoin d'avoir un nombre de données multiple de 8.
- La taille des données d'un paquet IP doit obligatoirement être un multiple de 8, sauf pour le dernier fragment d'un datagramme fragmenté, ou d'un paquet qui n'a pas du tout été fragmenté.

- Supposons qu'une machine souhaite envoyer un datagramme de 1600 octets. Sachant qu'une trame Ethernet pouvant transporter un datagramme de 1500 octets (1518 - 18 octets d'en-tête Ethernet), il va falloir donc fragmenter ce datagramme comme indiqué dans la Figure 2 ci-dessous :



# Fragmentation Datagramme IPv4

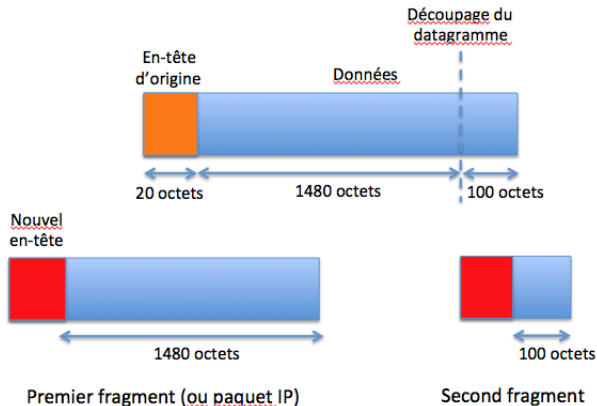


Figure 2 – Fragmentation paquet IPv4

# Exemple de Fragmentation Datagramme IPv4.

## Exemple

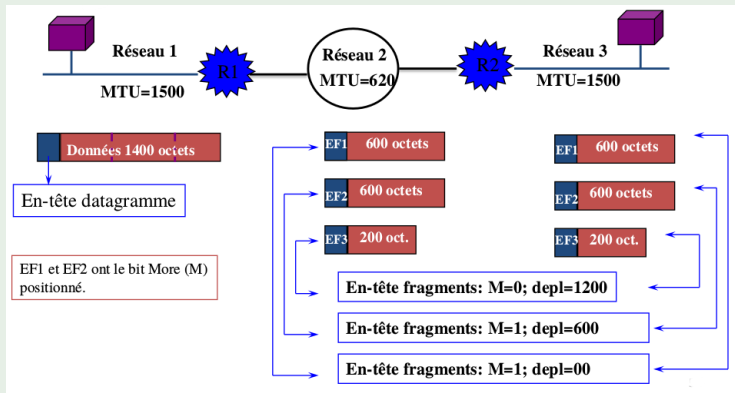
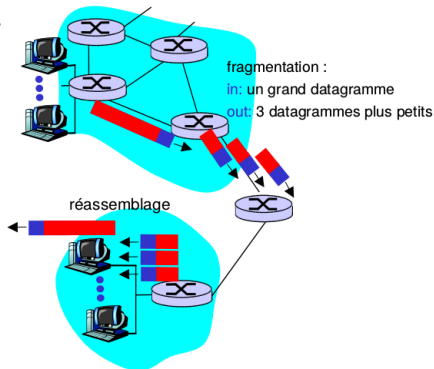


Figure 3 – Exemple de Fragmentation de paquet IPv4.

## Exemple

- les liens réseau ont un MTU (taille max. de transfert) : la plus grande trame possible.
  - types de liens différents, MTUs différents
- les grands datagrammes IP divisés ("fragmentés") dans le réseau
  - un datagramme donne lieu à plusieurs datagrammes
  - "réassemblés" uniquement au niveau de la dest. finale
  - bits de l'entête IP utilisés pour identifier, ordonner les fragments



## Exemple

### Exemple

- datagramme de 4000 octets
- MTU = 1500 octets

	long.	ID	flag frag.	offset	
	=4000	=x	=0	=0	

Un grand datagramme divisé en plusieurs petits datagrammes

	long.	ID	flag frag.	offset	
	=1500	=x	=1	=0	

	long.	ID	flag frag.	offset	
	=1500	=x	=1	=1480	

	long.	ID	flag frag.	offset	
	=1040	=x	=0	=2960	

# Fragmentation IPv4 : Exercice

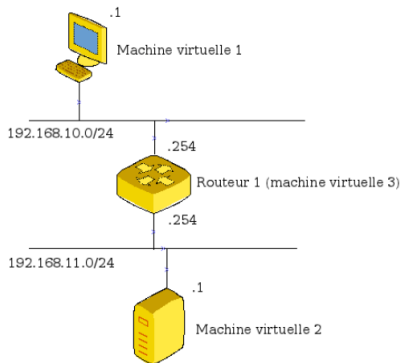


Figure 4 – Topologie du réseau

- La machine virtuelle 1 du réseau de la Figure 5 veut envoyer un datagramme de 5600 octets à la machine virtuelle 2. Les MTU pour les réseaux 192.168.10.0/24 et 192.168.11.0/24 sont identiques et valent 1500 octets (un réseau Ethernet habituel).
- Il faut donc fragmenter.

- ①  $1500 \text{ octets} - 20 \text{ octets} = 1480 \text{ octets}$ . 1480 est un multiple de 8,  $1480 = 8 * 185$  (sinon, on aurait cherché le plus proche entier de 1480 qui est un multiple de 8).
- ②  $5600 \text{ octets} - 20 \text{ octets} = 5580 \text{ octets}$
- ③  $\frac{5580}{1480} = 3.77027027$ . Donc, on aura 3 fragment de 1480 et un fragment de  $5580 - 1480 * 3 = 1140$

- ① Datagramme d'origine :
  - ID : 10254 (par exemple)
  - 0/DF/MF : 0/0/0
  - FO (Fragment Offset) : 0
- ② Premier fragment :
  - ID : 10254 (par exemple)
  - 0/DF/MF : 0/0/1
  - FO (Fragment Offset) : 0
- ③ Second fragment :
  - ID : 10254 (par exemple)
  - 0/DF/MF : 0/0/1
  - FO (Fragment Offset) :  $1480/8=185$



- ④ Troisième fragment :
  - ID : 10254 (par exemple)
  - 0/DF/MF : 0/0/**1**
  - FO (Fragment Offset) :  $(1480 \times 2) / 8 = 370$
- ⑤ Quatrième fragment :
  - ID : 10254 (par exemple)
  - 0/DF/MF : 0/0/**0**
  - FO (Fragment Offset) :  $(1480 \times 3) / 8 = 555$

## Vérification des calculs effectués

- Pour s'assurer que nous n'avons pas fait d'erreur de calcul, nous allons faire une petite vérification.
- Le Fragment Offset du dernier fragment représente toutes les données des fragments précédents. Si on y ajoute le nombre de ses données, on devrait retrouver le nombre de données total. Vérifions cela :
- $(555 \times 8) + 1140 = 4440 + 1140 = 5580$  octets, ce qui correspond bien.
- La valeur du Fragment Offset dépend de la taille des données contenues dans les fragments précédents et non dans celles contenues dans le fragment lui-même.

# Fragmentation IPv4 : Exercice

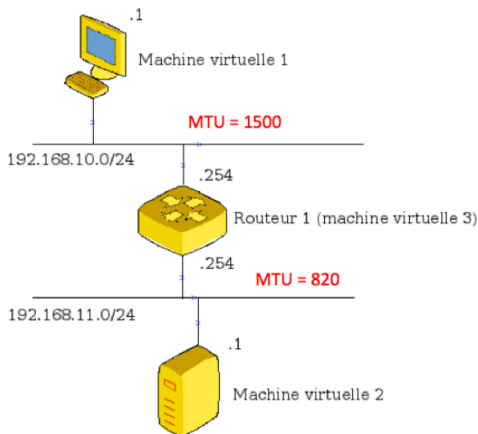


Figure 5 – Topologie du réseau (même réseau que précédemment, mais la MTU sur le second réseau est de 820).

- Le découpage va donc être, dans un premier temps, le même que le précédent.
- Mais une fois arrivés au routeur 1, la table de routage de celui-ci va lui dire de les envoyer sur le second réseau qui a une MTU de 820 Le routeur 1 a deux choix :
  - Soit il ré-assemble les fragments et les découpe une nouvelle fois
  - Soit il découpe directement les fragments reçus en fragments plus petits
  - C'est en fait la seconde solution qui est utilisée en réseau, car aucun matériel réseau n'a le droit de ré-assembler des fragments, sauf la machine destinatrice (et les firewalls parfois pour des raisons de sécurité).
  - Le routeur 1 va donc découper chacun des fragments qu'il a reçu en fonction de la nouvelle MTU qui est égale à 820.

- Le routeur 1 va donc découper chacun des fragments qu'il a reçu en fonction de la nouvelle MTU (820). Les nouveaux fragments vont donc avoir un maximum de 800 octets de données.
  - 1  $820 \text{ octets} - 20 \text{ octets} = 800 \text{ octets}$ . 800 est un multiple de 8,  $800 = 8 * 100$  (sinon, on aurait cherché le plus proche entier de 800 qui est un multiple de 8).
  - 2  $\frac{1480}{800} = 1.85$ . Donc, on aura 1 fragment de 800 et un fragment de  $1480 - 800 = 680$ . Ceci est valable pour les trois fragments de 1480 octets
  - 3  $\frac{1140}{800} = 1.425$ . Donc, on aura 1 fragment de 800 et un fragment de  $1140 - 800 = 340$ . Ceci est valable pour le fragment de 1140 octets

- ① Datagramme d'origine :
  - ID : 10254 (par exemple)
  - 0/DF/MF : 0/0/0
  - FO (Fragment Offset) : 0
- ② Premier fragment :
  - ID : 10254
  - 0/DF/MF : 0/0/1
  - FO (Fragment Offset) : 0
- ③ Second fragment :
  - ID : 10254 (par exemple)
  - 0/DF/MF : 0/0/1
  - FO (Fragment Offset) :  $800/8=100$

④ Troisième fragment :

- ID : 10254 (par exemple)
- 0/DF/MF : 0/0/**1**
- FO (Fragment Offset) :  $(800+680)/8=185$

⑤ Quatrième fragment :

- ID : 10254 (par exemple)
- 0/DF/MF : 0/0/**1**
- FO (Fragment Offset) :  $(2*800+680)/8=285$

⑥ Cinquième fragment :

- ID : 10254 (par exemple)
- 0/DF/MF : 0/0/1
- FO (Fragment Offset) :  $(2*800+680*2)/8=370$

⑦ Sixième fragment :

- ID : 10254 (par exemple)
- 0/DF/MF : 0/0/1
- FO (Fragment Offset) :  $(3*800+680*2)/8=470$



## 8 Septième fragment :

- ID : 10254 (par exemple)
- 0/DF/MF : 0/0/1
- FO (Fragment Offset) :  $(3*800+680*3)/8=555$

## 9 Huitième fragment :

- ID : 10254 (par exemple)
- 0/DF/MF : 0/0/0
- FO (Fragment Offset) :  $(4*800+680*3)/8=655$

## Vérification des calculs effectués

- Pour s'assurer que nous n'avons pas fait d'erreur de calcul, nous allons faire une petite vérification.
- Le Fragment Offset du dernier fragment représente toutes les données des fragments précédents. Si on y ajoute le nombre de ses données, on devrait retrouver le nombre de données total. Vérifions cela :
- $(655 \times 8) + 340 = 5240 + 340 = 5580$  octets, ce qui correspond bien.
- Nous tenons à rappeler que la valeur du Fragment Offset dépend de la taille des données contenues dans les fragments précédents et non dans celles contenues dans le fragment lui-même.

## Rappel

- Le flag More Fragment n'est positionné à 0 que pour le dernier fragment même si plusieurs fragmentations successives ont eu lieu.
- Le réassemblage des fragments n'a lieu qu'à la destination. Tout routeur intermédiaire ne réassemble pas ceux-ci (sauf les firewalls)