RFC : 791
STD 5
Septembre 1981

PROTOCOLE INTERNET

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, California 90291

PRÉFACE

Ce document spécifie le protocole Internet standard du DoD *(Department of Defense)*. Ce document se fonde sur les six éditions précédentes de la spécification ARPA Internet Protocol, et le présent texte en découle fortement. De nombreuses contributions ont été apportées à ce travail à la fois en termes de concepts et en termes de rédaction. Cette édition revoit certains aspects concernant l'adressage, la gestion des erreurs, les codes d'options, ainsi que les aspects sécurité, priorité, compartimentage, et restrictions d'usage définies par le protocole Internet.

Éditeur : Jon Postel
Traduction : V.G. Fremaux / EISTI
Relecture Claude Brière de L'Isle (11/2008)

RFC : 791
Remplace : RFC 760, IEN 128, 123, 111, 80, 54, 44, 41, 28, 26

Table des matières

1 INTRODUCTION

1.1 Motivation

Le Protocole Internet est conçu pour prendre en charge l'intercommunication de systèmes informatiques sur une base de réseau par commutation de paquets. Un tel système est appelé "catenet" [1]. Le rôle du protocole Internet est la transmission de blocs de données, appelés datagrammes, d'une source vers une destination, la source et la destination étant des ordinateurs hôtes identifiés par une adresse de longueur fixe. Le protocole Internet dispose des mécanismes permettant la fragmentation de longs datagrammes et leur réassemblage, lors de leur transmission à travers des réseaux de "dimension" inférieure.

1.2 Cadre

Le protocole Internet est limité aux fonctions nécessaires à l'acheminement d'un paquet de bits (un datagramme Internet) depuis une source vers une destination via un ensemble de réseaux interconnectés. Aucun mécanisme particulier destiné à augmenter la fiabilité des données de "bout en bout" n'y est mis en œuvre, ni mécanisme de contrôle de flux, de séquencement, ni aucun autre service communément fourni par d'autres protocoles "hôte vers hôte". Le protocole Internet capitalisera les services des réseaux qui le supportent pour offrir divers types et qualités de service.

1.3 Interfaces

Ce protocole est appelé par d'autres protocoles "hôte vers hôte" de l'environnement Internet. Ce protocole appelle à son tour un protocole de réseau local pour transporter le datagramme vers le routeur le plus proche ou directement vers l'hôte destinataire.

Par exemple, un module TCP s'appuiera sur le module Internet pour transporter un segment TCP (comprenant un en-tête TCP plus les données utilisateur) considéré lui-même comme le segment de données du datagramme Internet. Le module TCP renseignera les adresses et les autres paramètres de l'en-tête Internet par passage de paramètres lors de l'appel. Le module Internet constituera alors le datagramme Internet et appellera à son tour l'interface réseau local pour transmettre le datagramme.

Dans le cas d'ARPANET, par exemple, le module Internet appellera le module réseau local qui ajoutera l'en-tête 1822 [2] en début de datagramme, constituant ainsi un message ARPANET à transmettre à l'IMP. L'adresse ARPANET sera déduite de l'adresse Internet par l'interface réseau local et sera l'adresse d'un hôte raccordé à l'ARPANET, celui-ci pouvant être un routeur vers d'autres réseaux.

1.4 Fonctionnement

Le protocole Internet met en œuvre deux fonctions de base : l'adressage et la fragmentation. Les modules Internet exploiteront les adresses inscrites dans l'en-tête Internet pour acheminer le datagramme vers sa destination. La sélection d'un chemin partant de la source vers la destination est appelée le **routage**.

Les modules Internet exploitent des champs de l'en-tête Internet pour fragmenter et réassembler les datagrammes Internet lorsque le réseau à traverser n'accepte que des paquets de taille plus réduite.

Le modèle de fonctionnement est qu'il existera un module Internet dans chaque hôte concerné par la communication Internet ainsi que dans chaque routeur situé sur le chemin du datagramme. Ces modules partagent un certain nombre de règles communes pour l'interprétation des champs d'adresse et pour la fragmentation et le réassemblage des datagrammes Internet. De plus, ces modules (surtout dans les routeurs) disposeront de fonctions permettant de prendre des décisions de routage ainsi que d'autres fonctions.

Le protocole Internet considère chaque datagramme Internet comme une entité indépendante et sans relation aucune avec d'autres datagrammes. Il n'y a dans ce concept aucune notion de circuit ou de communication (ni virtuelle ni d'aucun ordre).

Le protocole Internet utilise quatre mécanismes clefs pour procurer le service promis : Type de Service, Durée de Vie, Options, et Somme de contrôle d'en-tête.

Le **Type de Service** indique la qualité du service désiré. Le type de service est un ensemble générique de paramètres qui caractérisent les choix de service disponibles sur le réseau qui prend en charge la communication Internet. Cette indication de type de service sera utilisée par les routeurs pour sélectionner les paramètres de transmission réels d'un réseau particulier, le réseau à utiliser sur le segment suivant, ou pour spécifier le routeur suivant lors du routage d'un datagramme.

La **Durée de Vie** indique une limite haute pour la durée de vie d'un datagramme dans le réseau. Elle est réglée par

l'émetteur du datagramme et décrémentée par les éléments actifs situés sur le chemin que parcourt le datagramme. Si cette durée de vie atteint la valeur zéro avant que le datagramme Internet n'atteigne sa destination, ce dernier sera détruit. Cette durée de vie peut être vue comme une temporisation d'autodestruction du datagramme.

Les **Options** fournissent les fonctions de contrôle utiles voire nécessaires dans certaines situations particulières mais secondaires pour la fonction essentielle de la communication. Les options permettent par exemple l'horodatage, le codage de sécurité, et des commandes de routage spéciales.

La **somme de contrôle d'en-tête** permet de vérifier que les informations utilisées pour le traitement d'un datagramme Internet ont été correctement transmises. Les données peuvent contenir des erreurs. Si la somme de contrôle échoue, le datagramme Internet est immédiatement rejeté par l'entité qui détecte l'erreur.

Le protocole Internet ne prend absolument pas en charge le contrôle d'intégrité des données transportées. Il n'intègre aucun mécanisme d'acquittement ni "de bout en bout" ni "segment par segment". Aucun contrôle d'erreur n'est effectué sur les données, il y a seulement une somme de contrôle de l'en-tête. Il n'y a aucun mécanisme de retransmission de paquet. Il n'y a aucun mécanisme de contrôle de flux.

Les erreurs détectées pourront être signalées via le protocole de message de commande Internet (ICMP, *Internet Control Message Protocol*) [3] mis en œuvre dans le module de protocole Internet.

2 VUE D'ENSEMBLE

2.1 Relations avec les autres protocoles

Le diagramme suivant montre la position du protocole Internet dans la hiérarchie des protocoles :

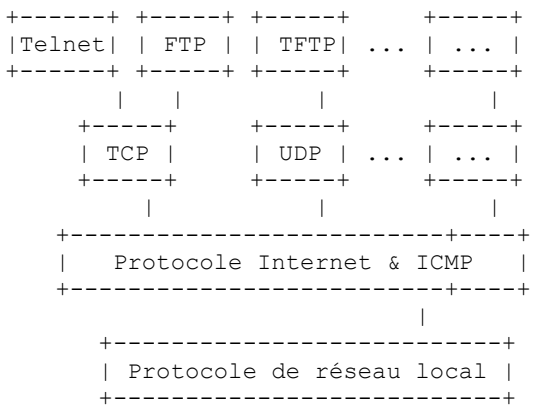


Figure 1 : Relations entre les protocoles

Le protocole Internet s'interface d'un côté avec les protocoles d'hôte vers hôte de niveau supérieur et de l'autre côté avec le protocole de réseau local. Dans ce contexte, un "réseau local" peut être un petit réseau d'entreprise comme un réseau beaucoup plus étendu comme ARPAnet.

2.2 Modèle de fonctionnement

Le modèle de fonctionnement de la transmission d'un datagramme d'un programme d'application vers un autre est illustré par le scénario suivant :

Nous supposons ici que la transmission traverse un routeur intermédiaire.

Le programme d"application émettrice prépare ses données et appelle son module Internet local pour envoyer ces données sous forme de datagramme et lui passe l'adresse de destination et les autres paramètres comme arguments de l'appel.

Le module Internet prépare un en-tête de datagramme et lui ajoute les données. Le module Internet détermine une adresse réseau locale correspondant à cette adresse Internet, dans notre cas, il s'agit de l'adresse d'un routeur. Il envoie ensuite ce datagramme ainsi que l'adresse réseau locale à l'interface réseau local.

L'interface réseau local crée un en-tête de réseau local, et ajoute à son tour le datagramme, puis envoie le résultat via le réseau local.

Le datagramme arrive sur un routeur hôte encapsulé dans son en-tête de réseau local, l'interface de réseau local

désensule cet en-tête, et transfère le datagramme vers le module Internet du routeur. Le module Internet routeur détermine en fonction de l'adresse Internet que le datagramme doit être transmis à un autre hôte sur un second réseau. Le module Internet détermine une nouvelle adresse de réseau local visant à ce moment l'ordinateur cible. Il appelle l'interface de réseau local traitant ce segment pour y envoyer le datagramme.

Cette interface de réseau local crée un nouvel en-tête de réseau local et y attache le datagramme puis transmet le tout à l'hôte de destination.

Arrivé à destination, le datagramme est extrait de son en-tête de réseau local par l'interface de réseau local destinataire, puis est transmis au module Internet destinataire.

Le module Internet détermine à quel programme applicatif de l'hôte est destiné le datagramme. Il passe alors les données au programme applicatif en réponse, accompagné de l'adresse de la source et des autres paramètres autitre de l'appel.

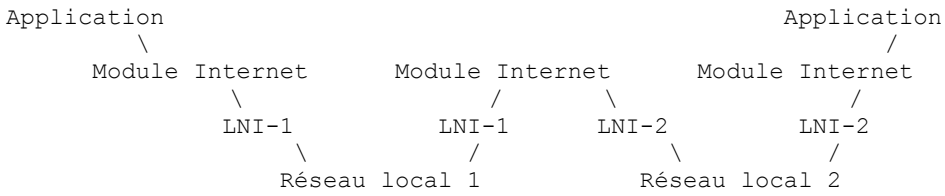


Figure 2 : Chemin de transmission

2.3 Description fonctionnelle

La fonction ou rôle du Protocole Internet est de déplacer les datagrammes à travers un ensemble de réseaux interconnectés. Ceci est réalisé en transférant les datagrammes d'un module Internet à l'autre jusqu'à atteindre la destination. Les modules Internet sont des programmes exécutés dans les hôtes et les routeurs du réseau Internet. Les datagrammes sont acheminés d'un module Internet à l'autre à travers les réseaux individuels sur la base de l'interprétation d'une adresse Internet. De ce fait, un des plus importants mécanismes du protocole Internet est la gestion de cette adresse Internet.

Lors de l'acheminement d'un datagramme d'un module Internet vers un autre, les datagrammes peuvent avoir éventuellement à traverser une section de réseau qui admet une taille maximale de paquet inférieure à celle du datagramme. Pour surmonter ce problème, un mécanisme de fragmentation est fourni par le protocole Internet.

Adressage

Une distinction doit être faite entre *noms*, *adresses*, et *chemins* [4]. Un nom indique ce que nous cherchons. Une adresse indique où cela se trouve. Un chemin indique comment y aboutir. Le protocole Internet s'occupe essentiellement des adresses. C'est à des protocoles de niveau plus élevé (par exemple, d'hôte vers hôte ou d'application) que revient la tâche de transposer les noms en adresses. Le module Internet transpose l'adresse Internet en une adresse de réseau local. La tâche qui consiste à transcrire l'adresse de réseau local en termes de chemin revient au procédures de niveau inférieur (par exemple, de réseau local ou de routeur).

Les adresses ont une longueur fixe de 4 octets (32 bits). Une adresse commence toujours par un numéro de réseau, suivi d'une adresse locale (appelée le champ "reste") codant l'adresse de l'hôte sur ce réseau. Il existe trois formats ou classes d'adresses Internet : pour la classe A, le bit de poids fort vaut zéro, les 7 bits suivants désignent le réseau, les derniers 24 bits désignent l'adresse locale de la machine ; pour la classe B, les deux bits de poids fort valent 1 et 0, les 14 bits suivants désignent le réseau et les 16 derniers bits l'adresse locale de machine ; pour la classe C, les trois bits de poids fort forment le schème 110, les 21 bits suivants forment l'adresse réseau et les 8 derniers bits l'adresse locale.

La transcription d'adresse Internet en adresse de réseau local doit être soumise à quelques précautions ; un hôte physique unique doit être capable d'agir comme si il était plusieurs hôtes distincts utilisant des adresses Internet distinctes. Certains hôtes peuvent disposer de plusieurs interfaces physiques (multi-homing).

De ce fait, il faudra pouvoir considérer le cas d'un hôte à plusieurs interfaces physiques, chacune abritant plusieurs adresses Internet distinctes.

Des exemples de répartition d'adresses peuvent être trouvés dans "Transpositions d'adresse" [5].

Fragmentation

La fragmentation du datagramme Internet devient nécessaire dès lors que pour atteindre sa destination, un datagramme de grande taille arrive sur une portion de réseau qui n'accepte la transmission que de paquets plus courts.

Un datagramme Internet peut être marqué "Ne pas fragmenter". Un tel datagramme Internet ne doit jamais être

fragmenté quelques soient les circonstances. Si un datagramme Internet marqué Ne pas fragmenter ne peut être acheminé jusqu'à sa destination sans être fragmenté, alors il devra être rejeté.

La fragmentation, la transmission et le réassemblage à travers un réseau local hors de vue d'un module de protocole Internet est appelée fragmentation Intranet et peut être utilisée [6].

Les procédures de fragmentation et réassemblage Internet doivent pouvoir "casser" un datagramme Internet en un nombre arbitraire de "fragments" qui pourront être réassemblés ultérieurement. Le récepteur des fragments utilise le champ d'identification pour s'assurer que des fragments de datagrammes différents ne puissent être mélangés. Le champ "Décalage de fragment" indique au récepteur la position du fragment reçu dans le datagramme original. Les champs "Décalage de fragment" et "Longueur Totale" déterminent la portion du datagramme original que représente le fragment. Le fanion "Fragments à venir" indique (lors de sa remise à zéro) au récepteur qu'il s'agit du dernier fragment. Ces champs véhiculent suffisamment d'information pour réassembler les datagrammes.

Le champ d'identification sert à distinguer les fragments d'un datagramme de ceux d'un autre datagramme. Le module Internet émetteur d'un datagramme Internet initialise le champ d'identification à une valeur qui doit être unique pour cette paire source-destination et pour ce protocole pendant toute la durée de transmission de ce datagramme. Le module Internet terminant l'émission d'un datagramme met le bit "Fragment à venir" et le champ "Décalage de fragment" à zéro.

Pour fragmenter un long datagramme, un module de protocole Internet (par exemple, dans un routeur), crée deux nouveaux datagrammes et copie le contenu des champs d'en-tête Internet du long datagramme dans les deux nouveaux en-têtes. Les données du long datagramme sont divisées en deux portions, la première d'une taille multiple de 8 octets (64 bits) (la taille de la seconde portion n'est donc pas nécessairement un multiple de 8 octets). Nous appellerons le nombre de blocs de 8 octets dans la première portion NBF (ou Nombre de Blocs du Fragment). La première portion de données est placée dans le premier des deux nouveaux datagrammes, et le champ "Longueur Totale" est renseigné avec la taille de ce datagramme. Le fanion "Fragment à venir" est basculé à 1. La seconde portion de données est placée dans le second des deux nouveaux datagrammes, et le champ "longueur totale" est renseigné avec la taille du second datagramme. Le fanion "Fragment à venir" est placé à la même valeur que celui du datagramme original. Le champ "Décalage de fragment" du second datagramme constitué est renseigné avec la valeur du même champ du datagramme original plus NFB.

Cette procédure peut être généralisée à une fragmentation en n fragments, plutôt que les deux décrits ci-dessus.

Pour réassembler les fragments d'un datagramme Internet, un module Internet (par exemple dans un hôte de destination) recombine les datagrammes dont les valeurs des quatre champs suivants sont identiques : identification, source, destination, et protocole. La recombinaison est réalisée en replaçant la portion de donnée contenue dans chaque fragment à la position relative indiquée par le champ "Décalage de fragment" lu dans l'en-tête Internet de ce fragment. Le premier fragment aura le champ Décalage de fragment à zéro, et le dernier fragment aura le bit "Fragment à venir" remis à zéro.

2.4 Routeurs

Les routeurs mettent en œuvre le protocole Internet pour transférer les datagrammes entre réseaux différents. Les routeurs mettront de plus en œuvre le protocole de routeur à routeur (GGP, *Gateway to Gateway Protocol*) [7] leur servant à coordonner le routage et à s'échanger d'autres informations de gestion du réseau.

Dans un routeur, les protocoles de niveau supérieur n'ont pas à être mis en œuvre et les fonctions GGP sont ajoutées an module IP.

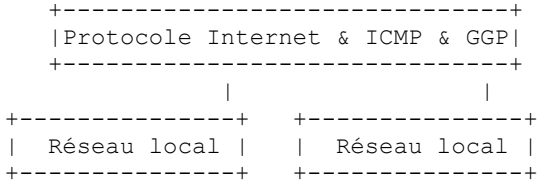
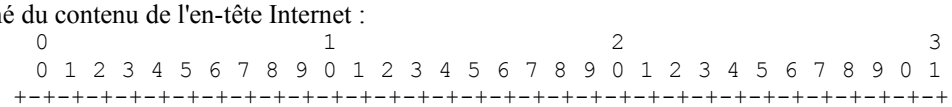


Figure 3. : Protocoles dans les routeurs

3 SPÉCIFICATION

3.1 Format d'en-tête Internet

Voici un résumé du contenu de l'en-tête Internet :



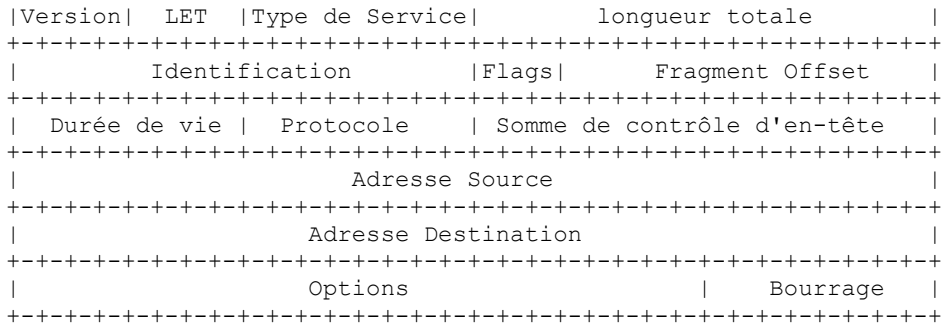


Figure 4 : Exemple d'en-tête de Datagramme Internet

Notez que chaque marque indique une position de bit.

Version : 4 bits

Le champ Version indique le format de l'en-tête Internet. Ce document décrit le format de la version 4 du protocole.

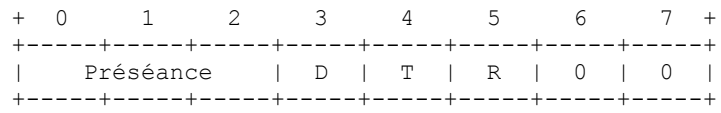
Longueur d'en-tête : 4 bits

Le champ Longueur d'En-Tête (LET) code la longueur de l'en-tête Internet, l'unité étant le mots de 32 bits, et de ce fait, marque le début des données. Notez que pour être valide ce champ ne peut prendre une valeur en dessous de 5.

Type de Service : 8 bits

Le Type de Service donne une indication sur la qualité de service souhaitée, qui reste cependant un paramètre "abstrait". Ce paramètre est utilisé pour "guider" le choix des paramètres des services réels lorsqu'un datagramme transite dans un réseau particulier. Certains réseaux offrent un mécanisme de priorité, traitant préférentiellement un tel trafic par rapport à un trafic moins prioritaire (en général en acceptant seulement de véhiculer des paquets d'un niveau de priorité au dessus d'un certain seuil lors d'une surcharge momentanée). Principalement, le choix offert est une négociation entre les trois contraintes suivantes : faible retard, faible taux d'erreur, et haut débit.

Bits 0-2 :	Préséance.	
Bit 3 :	0 = Retard standard,	1 = Retard faible.
Bits 4 :	0 = Débit standard,	1 = Haut débit.
Bits 5 :	0 = Taux d'erreur standard	1 = Taux d'erreur faible.
Bit 6-7 :	Réservé.	



Préséance

111 – Contrôle réseau
110 – Contrôle inter-réseaux
101 - CRITIC/ECP
100 – Subrogation éclair
011 - Éclair
010 - Immédiate
001 - Priorité
000 - Routine

L'utilisation des indications en termes de retard (D, *delay)*, débit (T, *throughput*), et fiabililité (R, *reliability*) peut augmenter le "coût" (d'un certain point de vue) du service. Dans la plupart des réseaux, de meilleures performances pour l'un de ces paramètres s'obtiennent au prix d'une dégradation des performances pour un autre. À moins d'une situation exceptionnelle, il sera préférable de ne pas activer plus de deux optimisations sur les trois.

Le "Type de Service" sert à préciser le traitement effectué sur le datagramme pendant sa transmission à travers Internet. Des exemples d'association de ce code aux améliorations de service proposées par des réseaux existants comme AUTODIN II, ARPANET, SATNET, et PRNET sont données dans la RFC 795 "Transpositions de service" [8].

La désignation de préséance "Contrôle réseau" est destinée à être utilisée comme une priorité à l'intérieur d'un seul réseau. L'utilisation et le contrôle réels de cette désigantion dépend de chaque résesau. La désignation "Contrôle inter-réseau" est destinée à n'être utilisée que par les générateurs de contrôle de routeurs. Si l'utilisation réelle de ces désignations de préséance a une signification pour un réseau particulier, il est de la responsabilité de ce dernier de contrôler l'accès et l'utilisation de ces désignations de préséance.

Longueur Totale : 16 bits

Le champ "Longueur Totale" est la longueur du datagramme, mesurée en octets, y compris l'en-tête Internet et les données. Ce champ permet une longueur de datagramme d'au plus 65 535 octets. Une telle longueur rendrait de toutes façons les datagrammes impossibles à gérer pour la plus grande partie des hôtes et des réseaux. Tous les hôtes devront être prêts à accepter des datagrammes jusqu'à une longueur de 576 octets (qu'ils arrivent en entier ou en fragments). Il est recommandé que les hôtes n'envoient des datagrammes de plus de 576 octets que dans la mesure où ils sont sûrs que la destination est capable de les accepter.

Le nombre 576 a été choisi pour permettre à un bloc de données de taille raisonnable d'être transmis dans un datagramme, tenant compte des données à ajouter pour constituer les informations d'en-tête. Par exemple, cette taille permet la transmission d'un bloc de 512 octets, plus 64 octets d'en-tête dans un datagramme unique. (*NdT : je rappelle ici que la taille de 512 octets correspond à un secteur sur la plupart des supports de stockage)* La taille maximale d'un en-tête Internet étant de 60 octets, et sa taille typique étant de 20 octets, ce nombre permet de conserver une bonne marge pour les en-têtes de protocoles de plus haut niveau.

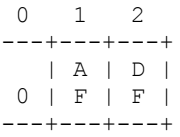
Identification : 16 bits

Une valeur d'identification allouée par l'émetteur pour identifier les fragments d'un même datagramme.

Fanions *(flags)* **:** 3 bits

Divers fanions de contrôle.

Bit 0 : réservé, doit être laissé à zéro
Bit 1 : (AF) 0 = Fragmentation possible, 1 = Non fractionnable.
Bit 2 : (DF) 0 = Dernier fragment, 1 = Autres fragments.



Décalage de fragment : 13 bits

Ce champ indique la position du fragment dans le datagramme complet. Le décalage du fragment est mesuré en blocs de 8 octets (64 bits). Le décalage du premier fragment vaut zéro.

Durée de vie : 8 bits

Ce champ indique la durée maximum pendant laquelle un datagramme est autorisé à rester dans le système Internet. Si ce champ prend la valeur zéro, le datagramme doit être détruit. Ce champ est modifié pendant le traitement de l'en-tête Internet. La durée de vie est mesurée en secondes. Chaque module Internet doit retirer au moins une unité de temps à ce champ, même si le traitement complet du datagramme par le module est effectué en moins d'une seconde. De ce fait, cette durée de vie doit être interprétée comme la limite supérieure du temps pendant lequel un datagramme peut exister. Ce mécanisme est motivé par la nécessité de détruire les datagrammes qui n'ont pu être acheminés, en limitant la durée de vie même du datagramme.

Protocole : 8 bits

Ce champ indique le protocole de prochain niveau utilisé dans la section données du datagramme Internet. La liste des valeurs des divers protocoles figure dans la RFC " Numéros alloués" [9].

Somme de contrôle d'en-tête : 16 bits

Une somme de contrôle calculée sur l'en-tête uniquement. Comme certains champs de l'en-tête sont modifiés (par exemple, durée de vie) pendant leur transit à travers le réseau, cette somme de contrôle doit être recalculée et vérifiée en chaque point du réseau où l'en-tête est traité.

L'algorithme de Somme de contrôle est le suivant :

Le champ Somme de contrôle est le complément à un sur 16 bits de la somme des compléments à un de tous les mots de 16 bits de l'en-tête. Pour les besoins du calcul de la somme de contrôle, la valeur du champ Somme de contrôle est zéro.

C'est une somme de contrôle facile à calculer et l'expérience indique qu'elle est adéquate. Il se peut que cet algorithme soit plus tard remplacé par un calcul de type CRC, suivant l'expérience qu'on en aura.

Adresse source : 32 bits

L'adresse Internet de la source. Voir au paragraphe 3.2.

Adresse destination : 32 bits

L'adresse Internet du destinataire. Voir au paragraphe 3.2.

Options : variable

Les datagrammes peuvent contenir ou non des options. Celles-ci doivent être mises en œuvre par tous les modules IP (hôtes et routeurs). Le caractère "optionnel" concerne leur transmission, et non leur mise en œuvre.

Dans certains environnements, l'option de sécurité peut être obligatoire dans tous les datagrammes.

Le champ Option est de longueur variable. Un datagramme peut comporter zéro ou plus options. Voici les deux formats possibles d'une option :

Cas 1 : Un type d'option codé sur un seul octet.

Cas 2 : Un octet codant le type d'option, un octet donnant la longueur de l'option, les octets de données de l'option réelle.

L'octet Longueur d'option compte les octets de type d'option, l'octet Longueur d'option ainsi que les octets de donnée de l'option.

L'octet de type d'option est composé de trois champs :
1 bit fanion de recopie,
2 bits classe d'option,
5 bits numéro d'option.

Le fanion de recopie marque le fait que l'option est recopiée dans tous les segments d'un datagramme fragmenté.
0 = non recopiée
1 = recopiée

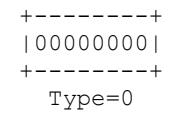
Les classes d'option sont :
0 = contrôle
1 = réservé pour usage futur
2 = débogage et mesure
3 = réservé pour usage futur

Les options suivantes sont actuellement définies :

Classe	Numéro	Longueur	Description
0	0	-	Fin de liste d'option. Sur un seul octet, pas d'octet de longueur.
0	1	-	Pas d'opération. Sur un seul octet, pas d'octet de longueur.
0	2	11	Sécurité. Transporte les informations de sécurité, compartiment, Groupe utilisateur (TCC), et Codes de Restriction d'usage compatibles DOD *(application militaire)*.
0	3	variable	Routage lâche de source. Utilisé pour acheminer le datagramme selon des informations données par la source.
0	9	variable	Routage strict de source. Utilisé pour acheminer le datagramme selon des informations données par la source.
0	7	variable	Traceur. Utilisé pour mémoriser le chemin pris par un datagramme Internet.
0	8	4	ID de flux. Transporte l'identifiant du flux.
2	4	variable	Horodatage Internet.

Définition des options spécifiques

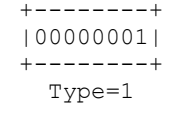
Fin de liste d'option



Cette option indique la fin de la liste d'options qui ne coïncide pas nécessairement avec la fin de l'en-tête Internet, selon la définition de la longueur de celui-ci. Cette option est utilisable une fois à la fin du bloc d'options, et non pas après chaque option, et peut n'être utilisée que dans le cas où la fin de liste d'options ne peut coïncider avec la fin de l'en-tête Internet. (*NdT : Rappel, un en-tête IP comporte toujours un multiple de 4 octets*).

Cet octet peut être recopié, introduit ou supprimé lors d'opérations de fragmentation, ou pour toute autre raison.

Pas d'opération

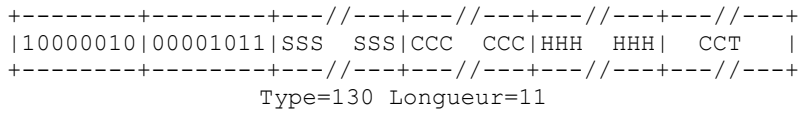


Cette option peut être utilisée entre deux options significatives, par exemple, pour aligner le début de l'option suivante sur le début d'un mot de 32 bits.

Peut être recopié, introduit, ou supprimé lors d'opérations de fragmentation, ou pour toute autre raison.

Sécurité

Cette option permet à un hôte d'envoyer des informations de sécurité, compartimentation, restrictions d'usage, et TCC (groupe fermé d'utilisateurs). Le format de cette option est le suivant :



Sécurité (Champ S) : 16 bits
Définit un niveau de sécurité parmi 16 (dont 8 sont réservés pour usage futur).

00000000 00000000 - Non classé
11110001 00110101 - Confidentiel
01111000 10011010 - EFTO
10111100 01001101 - MMMM
01011110 00100110 - PROG
10101111 00010011 - Restreint
11010111 10001000 - Secret
01101011 11000101 - Top Secret
00110101 11100010 - (Réservé pour usage futur)
10011010 11110001 - (Réservé pour usage futur)
01001101 01111000 - (Réservé pour usage futur)
00100100 10111101 - (Réservé pour usage futur)
00010011 01011110 - (Réservé pour usage futur)
10001001 10101111 - (Réservé pour usage futur)
11000100 11010110 - (Réservé pour usage futur)
11100010 01101011 - (Réservé pour usage futur)

Compartiments (Champ C): 16 bits

Une valeur nulle de ce champ indique que l'information n'est pas compartimentée. Les autres valeurs admissibles sont attribuées par la "Defense Intelligence Agency" américaine.

Restrictions d'usage (Champ H) : 16 bits

Les valeurs pour marquer la prise de contrôle et la libération sont des digraphes alphanumériques définis dans le "Defense Intelligence Agency Manual" DIAM 65-19, "Standard Security Markings".

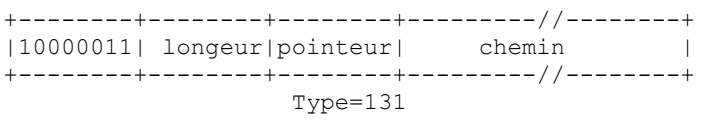
Code de Contrôle de Transmission (Champ TCC) : 24 bits

Procure un moyen de différencier le trafic et de définir des groupes contingentés d'abonnés partageant un même centre

d'intérêt. Les valeurs de TCC sont des trigraphes, et sont attribués par le HQ DCA Code 530.

Cette option est à recopier impérativement lors d'une fragmentation. Elle doit apparaître au plus une fois dans un datagramme.

Routage lâche et enregistrement du chemin



L'option de routage lâche et d'enregistrement de chemin (LSRR, *loose source and record route*) permet à la source d'un datagramme Internet de transmettre les informations d'acheminement aux routeurs qui acheminent le datagramme à sa destination, et d'enregistrer les indications de chemin parcouru.

Cette option débute par le code de type d'option. Le second octet est la longueur d'option qui inclut le code de type d'option et l'octet de longueur, l'octet pointeur, et les 3 octets de longueur de données du chemin. Le troisième octet est le pointeur sur les données d'acheminement qui indique la prochaine adresse de source à traiter. Le pointeur se rapporte à cette option, et la plus petite valeur légale pour le pointeur est 4.

Les données d'acheminement sont composées d'une série d'adresses Internet. Chaque adresse étant codée sur 32 bits, et donc 4 octets. Si la valeur du pointeur est plus grande que la longueur d'option, le chemin de source est vide (et le chemin enregistré plein) et le routage doit prendre comme référence le champ d'adresse destinataire.

Si l'adresse contenue dans le champ Adresse de destination a été atteinte et si le pointeur n'est pas supérieur à la longueur, l'adresse suivante dans l'acheminement de source remplace l'adresse dans le champ Adresse de destination, l'adresse de chemin enregistrée remplace l'adresse de source utilisée, et le pointeur est augmenté de quatre unités.

L'adresse de chemin enregistrée est la propre adresse Internet du module Internet telle qu'elle est connue dans l'environnement dans lequel le datagramme est transmis.

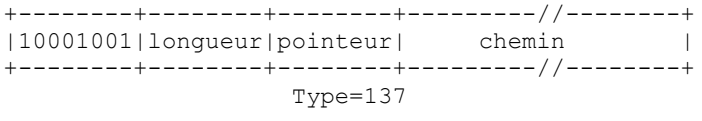
Cette procédure de remplacement de l'adresse de source par le chemin enregistré (bien que le chemin soit inscrit dans l'ordre inverse de ce qui serait nécessaire pour répondre au datagramme en utilisant le chemin inverse) permet de conserver à cette option (ainsi qu'à l'adresse IP en général) une longueur constante tout au long du "voyage" du datagramme à travers Internet.

Cette option spécifie une route de source "lâche" en ce sens qu'un routeur ou un hôte IP est autorisé à choisir n'importe quel nombre d'autres routeurs intermédiaires pour atteindre la prochaine adresse sur le chemin.

Doit impérativement être reporté lors d'une fragmentation. Ne peut apparaître qu'une seule fois dans un datagramme.

Note : Il faut comprendre le champ "chemin" comme une liste des adresses Internet de chaque module intermédiaire entre la source et le destinataire, constituant un chemin "préférentiel" tel que le connaît l'émetteur du datagramme. Au fur et à mesure que le datagramme progresse dans le réseau, chaque adresse est effectivement remplacée par celle du module réellement traversé par le datagramme. Le routage est dit "lâche" car le chemin suivi effectivement par le datagramme n'est pas obligatoirement celui qui est préconisé par la liste initiale fournie par la source.

Routage strict et enregistrement de chemin



L'option Routage strict et enregistrement du chemin (SSRR) permet à la source d'un datagramme Internet de transmettre des informations de routage à destination des routeurs qui acheminent le datagramme vers la destination, et d'enregistrer les indications de chemin parcouru.

Cette option débute avec le code de type de l'option. Le second octet est la longueur d'option qui inclut le code de type d'option et l'octet de longueur, l'octet pointeur, et les trois octets de longueur de données de chemin. Le troisième octet est le pointeur sur les données d'acheminement qui indiquent l'octet qui débute la prochaine adresse de source à traiter. Le pointeur se rapporte à cette option, et la plus petite valeur légale pour le pointeur est 4.

Les données de chemin se composent d'une série d'adresses Internet. Chaque adresse est de 32 bits, et donc 4 octets. Si la valeur du pointeur est plus grande que la longueur d'option, le chemin de source est vide (et le chemin enregistré plein) et le routage doit prendre comme référence le champ Adresse de destination.

Si l'adresse contenue dans le champ Adresse de destination a été atteinte et le pointeur n'est pas supérieur à la longueur,

l'adresse suivante dans la route de source remplace l'adresse contenue dans Adresse de destination, et l'adresse de chemin enregistré remplace l'adresse de source qui vient d'être utilisée. Le pointeur est augmenté de quatre unités.

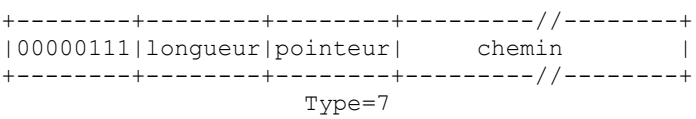
L'adresse de chemin enregistrée est la propre adresse Internet du module Internet telle qu'elle est connue de l'environnement dans lequel ce datagramme est transmis.

Cette procédure qui consiste à remplacer le chemin de source par le chemin enregistré (bien que le chemin soit inscrit dans l'ordre inverse que ce qui serait nécessaire pour répondre au datagramme en utilisant le chemin inverse) permet de conserver à cette option (ainsi qu'à l'adresse IP en général) une longueur constante tout au long du "voyage" du datagramme à travers Internet.

Cette option spécifie un acheminement "strict" de source en ce sens qu'un routeur ou un hôte IP doit envoyer le datagramme directement à la prochaine adresse dans le chemin de source seulement à travers le réseau directement connecté indiqué dans la prochaine adresse pour atteindre le prochain routeur ou hôte spécifié dans le chemin.

Doit impérativement être recopié lors d'une fragmentation. Doit apparaître au plus une fois dans un datagramme.

Traceur



L'option Traceur permet d'enregistrer le chemin parcouru par un datagramme Internet.

Cette option débute par le code de type d'option. Le second octet est la longueur de cette option qui inclut le code de type d'option et l'octet de longueur, l'octet pointeur, et les trois octets de longueur des données de chemin. Le troisième octet est le pointeur sur les données de chemin qui indiquent l'octet qui débute la prochaine zone où doit être mémorisée une adresse de chemin. Le pointeur se rapporte à cette option, et sa plus petite valeur légale est 4.

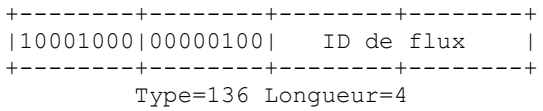
Un chemin enregistré se compose d'une série d'adresses Internet. Chaque adresse étant codée sur 32 bits, et donc 4 octets. Si la valeur du pointeur est supérieure à la longueur d'option, la zone de données de chemin enregistré est pleine. L'hôte émetteur du datagramme devra composer cette option en prévoyant une zone de données de chemin suffisamment longue pour contenir toute l'adresse prévue. La taille de l'option ne doit effectivement plus changer lors de l'enregistrement effectif du chemin. Le chemin, au départ du datagramme est initialisé avec des zéros par l'émetteur.

Lorsqu'un module Internet achemine un datagramme, il vérifie la présence de l'option traceur. Si c'est le cas, il insère sa propre adresse Internet telle qu'elle est connue de l'environnement dans lequel le datagramme va être transmis dans le chemin enregistré à l'octet indiqué par le pointeur, puis incrémente le pointeur de quatre unités.

Si la zone de données de chemin est déjà entièrement remplie (le pointeur excède la longueur de l'option), le datagramme est retransmis sans insérer l'adresse dans le chemin enregistré. S'il reste de la place dans la zone, mais pas assez pour insérer une adresse complète, alors cela indique une erreur et le datagramme doit être détruit. Dans ces deux cas, un message d'erreur ICMP peut être envoyé à l'hôte de source [3].

Ne doit pas être recopié lors d'une fragmentation, mais apparaître seulement dans le premier fragment. Ne peut apparaître qu'une fois au plus dans un datagramme.

Identifiant de flux

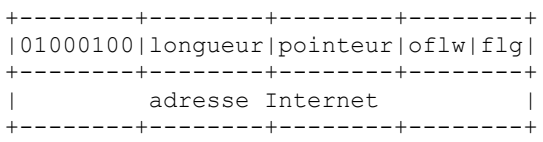


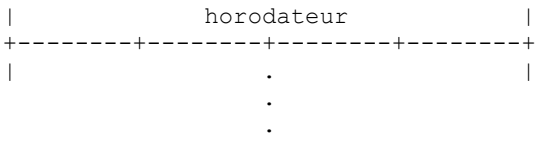
(Corrigé selon Errata du 03/01/2007)

Cette option permet la transmission de l'identifiant de flux SATNET de 16 bits à travers des réseaux qui ne prennent pas en charge la notion de flux.

Doit être recopié lors de fragmentation. Ne peut apparaître au plus qu'une fois dans un datagramme.

Horodateur Internet





Longueur d'option compte le nombre d'octets de l'option y compris le type, la longueur, le pointeur, et les octets de dépassement de capacité/fanions (longueur maximale 40).

Le Pointeur est le nombre d'octets depuis le début de cette option jusqu'à la fin de l'horodatage plus un (c'est-à-dire qu'il pointe sur l'octet qui commence l'espace pour le prochain horodatage). La plus petite valeur légale est 5. La zone Horodatage est pleine lorsque la valeur du pointeur dépasse la longueur de l'option.

Le champ de dépassement de capacité (oflw) [4 bits] compte le nombre de modules IP qui n'ont pas pu enregistrer d'horodatage faute de place dans la zone.

Les valeurs des fanions (flg) [4 bits] sont :

0 – Horodatages seuls, mémorisés sous forme de mots consécutifs de 32 bits.
1 – Chaque horodatage est précédé de l'adresse Internet de l'entité qui l'a enregistré.
3 – Les champs d'adresse Internet sont pré-spécifiés. Un module IP n'enregistre son horodatage que si il fait correspondre sa propre adresse Internet à l'adresse Internet spécifiée suivante.

L'horodatage compte sur 32 bits le temps écoulé depuis 0 heure UTC en millisecondes, et est justifié à droite. Si cette valeur n'est pas disponible en millisecondes ou ne peut être calculée à partir de la référence 0 heure UTC, alors la valeur disponible sera marquée dans l'étiquette et le bit de poids fort du champ Horodatage sera marqué à un pour prévenir de l'utilisation d'un format non standard.

L'émetteur du datagramme doit composer cette option en prévoyant une zone de données d'horodatage suffisamment longue pour contenir toutes les informations d'horodatage attendues. La taille de l'option ne doit effectivement plus changer lors de l'enregistrement effectif des horodatages. Le contenu initial de la zone de données d'horodatage doit être à zéro ou à des paires d'adresse Internet à zéro.

Si la zone des données d'horodatage est déjà pleine (le pointeur dépasse la longueur), le datagramme est retransmis sans insérer l'horodatage, mais le compteur de dépassement de capacité est incrémenté de un.

S'il reste de la place dans la zone, mais insuffisamment pour enregistrer un horodatage complet, ou si le champ de dépassement de capacité lui-même est au maximum de sa valeur, le datagramme d'origine est considéré en erreur et sera détruit. Dans ces deux cas, un message d'erreur ICMP problème de paramètre peut être retourné à l'hôte de source [3].

L'option Horodatage ne doit pas être recopiée lors d'une fragmentation. Elle est transportée dans le premier fragment. Elle ne doit apparaître qu'une fois dans un datagramme.

Bourrage : variable

Le bourrage d'en-tête Internet est utilisé pour assurer qu'il se termine sur une limite de 4 octets. Le bourrage se fait par des bits à zéro.

3.2 Discussion

La mise en œuvre d'un protocole doit répondre au principe de robustesse. Chaque mise en œuvre doit s'attendre à interopérer avec une autre mise en œuvre programmée par quelqu'un d'autre. Bien que la fonction de cette spécification soit de décrire explicitement ce protocole, il reste néanmoins la possibilité de voir apparaître des interprétations divergentes. On adopte comme principe général qu'une mise en œuvre doit être stricte quant à ce qu'elle émet, et libérale par rapport à ce qu'elle reçoit. C'est à dire qu'elle doit faire attention à émettre des datagrammes conformes et correctement constitués, mais doit accepter tout datagramme qu'elle est en mesure d'interpréter (par exemple, exempt d'erreurs d'ordre technique et tant que sa signification reste déchiffrable).

Les services de base d'Internet s'appuient sur le concept de datagramme qui prévoit une possibilité de fragmentation par les routeurs, avec une fonction de réassemblage exécutée par le module Internet de l'hôte de destination. Bien sûr, la fragmentation et le réassemblage des datagrammes, localement à un segment de réseau, ou suite à un accord particulier entre les routeurs d'un même réseau, sont permis, dans la mesure où cette technique est totalement transparente pour les protocoles Internet et pour les protocoles de niveau supérieur. Ce type de fragmentation-réassemblage transparent est appelé "dépendant du réseau" (ou encore Intranet) et ne sera plus évoqué dans la suite.

Les adresses Internet distinguent les sources et les destinations au niveau de l'hôte et comportent aussi un champ "protocole". Il est supposé ici que chaque protocole disposera de toutes les fonctions de multiplexage nécessaires à l'intérieur même de l'hôte.

Adressage

Pour conserver toute la souplesse d'allocation d'adresse à des réseaux et pouvoir prendre en compte un grand nombre de réseaux de petite taille ou de taille moyenne, la structure des champs d'adresse est codée de sorte à désigner un petit nombre de réseaux accueillant un très grand nombre d'hôtes, un nombre modéré de réseaux accueillant un nombre modéré d'hôtes, et un grand nombre de réseaux accueillant un nombre restreint d'hôtes. De plus, un codage spécial permet de prévoir un mode d'adressage étendu futur.

Formats d'adresse :

Bits de poids forts	**Format**	**Classe**
0	7 bits réseau, 24 bits hôte	A
10	14 bits réseau, 16 bits hôte	B
110	21 bits réseau, 8 bits hôte	C
111	basculement en mode adressage étendu	

Une valeur zéro dans le champ réseau signifie "ce réseau". Ceci n'est utilisé que dans certains messages ICMP. Le mode d'adressage étendu est à ce jour non défini. Ces deux interprétations sont réservées pour un usage futur.

Les valeurs allouées actuellement pour les adresses de réseau sont données dans le document "Numéros alloués" [9].

L'adresse locale, allouée par le réseau local, doit permettre à un seul hôte "physique" d'agir comme plusieurs hôtes Internet distincts. Ceci veut dire qu'il doit y avoir une transposition entre les adresses Internet d'hôte et les interfaces réseau/hôte permettant à plusieurs adresses Internet d'être accessibles par la même interface. Un hôte doit à l'inverse pouvoir disposer de plusieurs interfaces physiques au réseau et traiter les datagrammes y parvenant comme s'ils avaient été adressés à un hôte unique.

Les transpositions d'adresses Internet en adresses ARPANET, SATNET, PRNET, ou d'autre réseaux sont définies dans le document "Transpositions d'adresse" [5].

Fragmentation et Réassemblage.

Le champ Identification (ID) permet, en combinaison avec les adresses de source et de destination et les champs de protocole, d'identifier les fragments appartenant au même datagramme en vue du réassemblage.

Le bit du fanion Fragment à venir (DF) est mis à un si le datagramme ne porte pas le dernier fragment du datagramme original. Le champ Décalage de fragment identifie la position relative du fragment transporté, par rapport au début du datagramme original non fragmenté. Les fragments sont mesurés par blocs de 8 octets. La stratégie de fragmentation est ainsi faite qu'un datagramme non fragmenté porte tous les champs de contrôle de fragmentation à zéro (DF = 0, décalage de fragment = 0). Si un datagramme Internet est fragmenté, alors le découpage de la portion de données devra être fait par blocs de multiples de 8 octets excepté le dernier fragment.

Le format choisi pour Décalage de fragment permet la numérotation de 2**13 = 8192 positions de blocs de 8 octets chacun pour un total de 65 536 octets. Notez que ceci est cohérent avec le format du champ Longueur totale (bien sûr, l'en-tête est compté pour le calcul de la longueur totale, et pas pour la position relative des segments).

Lors d'une fragmentation, certaines options sont recopiées dans chaque en-tête de fragment, d'autres ne sont transmises qu'une fois dans l'en-tête du premier segment.

Tout module Internet doit être capable de traiter un datagramme d'au moins 68 octets sans fragmentation supplémentaire. Ceci est dû au fait qu'un en-tête Internet comprend au plus 60 octets, et le fragment minimal fait 8 octets.

Tout destinataire Internet doit être capable de recevoir un datagramme d'au moins 576 octets soit d'un seul morceau soit en plusieurs fragments à réassembler.

Les champs qui peuvent être affectés par la fragmentation sont :

(1) le champ Options
(2) le fanion Fragment à venir
(3) le champ Décalage de fragment
(4) le champ Longueur d'en-tête Internet
(5) le champ Longueur totale

(6) la somme de contrôle d'en-tête

Si le bit fanion anti-fragmentation (DF) est mis à un, alors toute fragmentation du datagramme Internet est rigoureusement INTERDITE, bien que le datagramme puisse être éliminé. Ceci peut être utilisé pour interdire la fragmentation dans le cas où les modules récepteurs ne disposent pas de ressources mémoires suffisantes pour réassembler correctement les fragments.

Un exemple d'utilisation de cette fonctionnalité est lorsque l'on veut diminuer la charge en ligne d'un petit hôte. Un petit hôte peut travailler sous un système d'exploitation minimum (bootstrap) acceptant un datagramme en entrée, l'enregistrant en mémoire, puis l'exécutant.

Les procédures de fragmentation et de réassemblage sont bien mieux décrites par des exemples. La procédure suivante est un exemple de mise en œuvre de fragmentation.

Dans les pseudo-programmes suivants, les conventions ci-après sont utilisées : "≤" signifie "inférieur ou égal", "#" signifie "différent de", "=" signifie "égal à", "<-" signifie "est réglé à". De plus, "x à y" inclut x et exclut y ; par exemple, "4 à 7" comprend 4, 5, et 6 (mais pas 7).

Exemple de procédure de fragmentation

Le datagramme de la plus grande taille pouvant être transmise à la section de réseau suivante est appelé unité de transmission maximale (UTM).

Si la longueur totale est inférieure ou égale à la taille de l'UTM alors le datagramme doit être directement transmis à l'étape suivante du traitement du datagramme ; autrement, le datagramme est coupé en deux fragments, le premier de taille égale à la taille de l'UTM, et le second avec ce qui reste. Le premier fragment est soumis à l'étape suivante, tandis que le deuxième est "réentré" dans la présente procédure, au cas où sa taille dépasserait encore la taille de l'UTM.

Notation:

FO - Décalage de fragment *(Fragment Offset)*
LET - Longueur d'en-tête
AF - Fanion anti-fragmentation
DF - Fanion Fragment à suivre
LT - Longueur totale
OFO - Ancien décalage de fragment *(Old Fragment Offset)*
OIHL - Ancienne longueur d'en-tête Internet *(Old Internet Header Length)*
ODF - Ancien fragment à suivre
OLT - Ancienne longueur totale
NBF - Nombre de blocs de fragments
UTM - Unité de transmission maximum

Procédure:

```
SI LT ≤ UTM ALORS
        Soumettre le datagramme à l'étape suivante
AUTREMENT SI AF = 1 ALORS détruire le datagramme
AUTREMENT
// Pour produire le premier fragment :
(1)     Copier l'en-tête Internet d'origine ;
(2)     OIHL <- LET; OLT <- LT; OFO <- FO; ODF <- DF ;
(3)     NBF <- (UTM-LET*4)/8 ;
(4)     Attacher les NBF*8 premiers octets de donnée ;
(5)     Corriger l'en-tête :
        DF <- 1;  TL <- (LET*4)+(NBF*8) ;
        Recalculer la somme de contrôle ;
(6)     Soumettre le fragment à l'étape suivante du traitement du datagramme ;

// pour produire le deuxième fragment  :
(7)     Copier sélectivement l'en-tête internet (seulement certaines options, cf. définition des options) ;
(8)     ajouter le reste des données ;
(9)     Corriger l'en-tête :
        LET <- (((OIHL*4)-(longueur des options non copiées))+3)/4 ;
        LT <- OLT - NBF*8 - (OLET-LET)*4) ;
        FO <- OFO + NBF ;  DF <- ODF;  Recalculer la somme de contrôle ;
(10)    Soumettre ce fragment au test de fragmentation ; TERMINÉ.
```

Dans la procédure ci-dessus, tous les fragments (sauf le dernier) ont la taille maximale qu'admet le réseau en sortie. Une autre mise en œuvre pourrait produire des fragments d'une taille inférieure à la taille maximale. Par exemple, une solution consisterait à diviser récursivement un datagramme en deux (en respectant la règle des blocs de 8 octets) tant que les datagrammes restent supérieurs à la taille de l'UTM.

Exemple de procédure de réassemblage

Pour chaque datagramme, l'identifiant de mémoire tampon est constitué en enchaînant les adresses de source, de destination, le champ protocole, et d'identification. Si c'est un datagramme complet (c'est à dire que ses champs Décalage de fragment et Fragment à venir sont tous deux à zéro), alors toutes les ressources de réassemblage pour cet identifiant de mémoire tampon sont libérées et le datagramme est transmis à l'étape de traitement suivante.

Si aucun autre fragment avec cet identifiant de mémoire tampon n'est actuellement en cours, alors les ressources de réassemblage sont allouées. Les ressources de réassemblage consistent en une mémoire tampon de données, une autre pour l'en-tête, un tableau des bits des blocs de fragments, un champ de longueur totale des données, et un temporisateur. Les données du fragment sont copiées dans la mémoire tampon de données à leur position relative indiquée par le décalage de fragment et l'indication de longueur, et les bits sont installés dans le tableau des bit des blocs de fragments correspondant aux blocs de fragments reçus.

S'il s'agit du premier fragment (celui dont Décalage de fragment est à zéro) son en-tête est placé dans la mémoire tampon d'en-tête. S'il s'agit du dernier fragment (celui dont le champ Fragment à suivre vaut zéro) la longueur des données totale est calculée. Si ce fragment, qu'il soit le dernier ou non, termine le datagramme (ce qui est vérifié en contrôlant les bits mis à un dans le tableau des blocs de fragments), le datagramme est alors envoyé à l'étape de traitement suivante ; sinon, le temporisateur est réglé à la valeur maximum du champ durée de vie notifiée dans ce fragment ou du temporisateur selon la plus grande des deux ; le sous-programme de réassemblage rend alors la main.

Si le temporisateur arrive à expiration, toutes les ressources consommées pour cet identifiant de mémoire tampon sont libérées. Le réglage initial du temporisateur est la limite inférieure du temps d'attente de réassemblage. Ce choix se justifie du fait que le temps effectif de réassemblage peut augmenter si le champ durée de vie du fragment reçu est supérieur à la valeur courante de temporisation, mais pas diminuer si il est inférieur. La valeur maximale que ce temporisateur peut prendre est la durée de vie maximum (approximativement 4,25 minutes). La valeur du réglage de temporisation initiale recommandée aujourd'hui est d'environ 15 secondes. Cette valeur sera susceptible de changement à l'usage. Notez que le choix de la valeur de paramètre est lié à la capacité de mémoire tampon disponible ainsi qu'à la vitesse de transmission du support ; c'est-à-dire que débit de données * valeur du temporisateur = taille de la mémoire tampon (par exemple, 10 kbit/s * 15 s = 150 kbit).

Notation :

FO - Décalage de fragment
IHL - Longueur d'en-tête
MF - Fanion Fragment à suivre
TTL - Durée de vie
NFB - Nombre de blocs de fragments
TL - Longueur totale
TDL - Longueur totale des données
BUFID - Identifiant de mémoire tampon
RCVBT - Tableau des bits des blocs reçus
TLB - Limite inférieure du temporisateur

Procédure:

```
(2) SI FO =  0 ET MF =  0
    (3)     ALORS SI mémoire tampon avec BUFID est allouée
    (4)     ALORS purger toutes les ressources de réassemblage pour ce BUFID ;
    (5)     Soumettre le datagramme à l'étape suivante ; FAIT.
    (6)     AUTREMENT SI aucune mémoire tampon avec BUFID n'est allouée
    (7)     ALORS allouer les ressources de réassemblage avec BUFID ; TIMER <- TLB ; TDL <- 0;
    (8)     mettre les données du fragment dans la mémoire tampon de données
            avec BUFID de l'octet FO*8 à l'octet (TL-(IHL*4))+FO*8 ;
    (9)     mettre les bits RCVBT de FO to FO+((TL-(IHL*4)+7)/8) ;
    (10)    SI MF =  0 ALORS TDL <- TL-(IHL*4)+(FO*8)
    (11)    SI FO =  0 ALORS mettre l'en-tête dans la mémoire tampon d'en-tête
```

```
(12)    SI TDL # 0
(13)    ET tous les bits RCVBT de 0 à (TDL+7)/8 sont mis à un
(14)    ALORS TL <- TDL+(IHL*4)
(15)    Soumettre le datagramme à l'étape suivante ;
(16)    Libérer toutes les ressources de réassemblage pour ce BUFID ; FAIT.
(17)    TIMER <- MAX(TIMER,TTL) ;
(18)    abandonner jusqu'au prochain fragment ou l'expiration du temporisateur :
(19)    expiration du temporisateur : purger toutes les ressources de réassemblage avec ce BUFID ; FAIT.
```

Dans le cas où deux fragments ou plus contiennent les mêmes données, soit intégralement, soit partiellement, cette procédure utilisera la dernière copie des données reçues dans la mémoire tampon et dans le datagramme livré.

Identification

Le choix d'un identifiant de datagramme est motivé par la nécessité de pouvoir distinguer de façon unique les fragments appartenant à un datagramme particulier. Le module de protocole qui rassemble les fragments juge que des fragments appartiennent à un même datagramme si ils ont une source, une destination, un protocole, et un identifiant identiques. De ce fait, l'émetteur doit choisir un identifiant univoque pour telle paire de source et destinataire, et pour tel protocole durant toute la durée pendant laquelle le datagramme (ou de tous ses fragments) pourrait être actif dans l'Internet.

Il semble que le module Internet émetteur doive garder en mémoire un tableau des identifiants, avec une entrée pour chaque destination avec laquelle il a communiqué dans la période de durée de vie maximale d'un paquet pour l'Internet.

Cependant, comme le champ Identifiant autorise 65 536 valeurs différentes, certains hôtes peuvent être capables d'utiliser simplement des identifiants univoques indépendamment de la destination.

Il est approprié pour certains protocoles de niveau supérieur de choisir l'identifiant. Par exemple, les modules de protocole TCP peuvent retransmettre un segment TCP identique, et la probabilité d'une réception correcte sera augmentée si la retransmission porte le même identifiant que la transmission originale dans la mesure où les fragments de l'une ou l'autre transmission peuvent servir à reconstruire le segment TCP correct.

Type de Service

Le type de service (TOS) sélectionne la qualité de service Internet délivrée. Le type de service est spécifié selon les paramètres abstraits de préséance, retard, débit, et fiabilité. Ces paramètres abstraits doivent être transposés dans les paramètres de service réels des réseaux particuliers que traverse le datagramme.

Préséance. Une mesure objective de l'importance de ce datagramme.
Délai. Une livraison rapide est importante pour les datagrammes qui ont cette indication.
Débit. Un débit de données élevé est important pour les datagrammes qui portent cette indication.
Fiabilité. Un effort particulier doit être fait pour assurer la livraison de ces datagrammes.

Par exemple, ARPANET a un bit de priorité, et un choix entre des messages "standard" (type 0) et des messages "non contrôlés" (type 3), (le choix entre des messages à paquet unique ou à paquets multiples peut aussi être considéré comme un paramètre de service). Les messages non contrôlés ont tendance à être livrés avec une fiabilité moindre et subissent moins de retard. Supposons qu'un datagramme Internet doive transiter par ARPANET. Soit son type de service défini par :

```
Préséance :   5
Délai :       0
Débit :       1
Fiabilité :   1
```

Dans cet exemple, la transposition de ces paramètres dans les paramètres de service disponibles pour l'ARPANET ferait mettre à un le bit de priorité d'ARPANET car la préséance Internet est dans la moitié supérieure de sa gamme, ferait choisir message standard car les exigences de débit et de fiabilité sont indiquées mais pas le délai. Plus de détails sont donnés sur les transpositions de service dans le document "Transpositions de service" [8].

Durée de vie

La durée de vie est réglée par l'émetteur du datagramme à la durée maximum pendant laquelle le datagramme sera admis dans le système Internet. Si le datagramme est plus longtemps dans le système Internet que la durée de vie, alors ce datagramme doit être détruit.

Ce champ doit être décrémenté en chaque point du réseau où l'en-tête Internet est traité pour représenter le temps passé à traiter le datagramme. Même si aucune information locale n'est disponible sur le temps réellement passé, le champ doit être décrémenté de 1. Le temps est mesuré en unités de secondes (c'est-à-dire que la valeur 1 signifie une seconde).

Et donc, la durée de vie maximale est de 255 secondes soit 4,25 minutes. Comme chaque module Internet qui traite un datagramme doit décrémenter le TTL d'au moins un même si le traitement du datagramme a demandé moins de temps, le TTL ne doit être interprété que comme la durée théorique maximale pendant laquelle le datagramme peut exister. L'intention de ce mécanisme est d'éliminer automatiquement les datagrammes qui n'ont pu trouver leur destinataire, et de fixer une limite à la durée de vie maximale d'un datagramme.

Certains protocoles de fiabilité de connexion de niveau supérieur se fondent sur l'hypothèse que les vieux duplicata de datagrammes n'arriveront plus après qu'un certain temps se soit écoulé. Le mécanisme du TTL est un moyen pour que de tels protocoles aient l'assurance que leur hypothèse est satisfaite.

Options

Les options sont facultatives dans chaque datagramme, mais exigées dans les mises en œuvre. En d'autres termes, la présence ou l'absence d'une option dans le datagramme reste un choix de l'émetteur, mais tout module Internet doit être capable d'analyser chaque option. Il peut y avoir plusieurs options dans le champ Options.

Les options peuvent ne pas se terminer sur une limite qui soit un multiple de 32 bits. L'en-tête Internet doit être complété par des octets remplis de zéros. Le premier de ces octets nuls sera interprété comme l'option Fin de liste d'options, les octets suivants comme bourrage d'en-tête Internet.

Tout module Internet doit être capable d'agir sur toute option. L'option Sécurité doit être utilisée en cas de transmission de trafic à diffusion compartimentée, restreinte ou confidentielle.

Somme de contrôle

La somme de contrôle d'en-tête Internet est recalculée si l'en-tête Internet a subi une modification. Par exemple, une réduction de la durée de vie, des ajouts ou modifications d'options Internet, ou suite à une fragmentation. Cette somme de contrôle Internet est destinée à protèger l'en-tête contre les erreurs de transmission.

Il existe certaines applications pour lesquelles quelques erreurs de données binaires restent acceptables alors qu'un retard de retransmission ne l'est pas. Si le protocole Internet mettait en application la notion de contrôle de transmission sur les données, de telles applications ne pourraient plus être prises en charge.

Erreurs

Toutes les erreurs en rapport avec le protocole Internet peuvent être reportées à l'aide de messages ICMP [3].

3.3 Interfaces

La description fonctionnelle des interfaces d'utilisateur avec IP est, au mieux, une fiction, dans la mesure où chaque système d'exploitation proposera ses propres facilités. Par conséquent, nous nous devons d'avertir le lecteur que des mises en œuvre distinctes d'IP pourront présenter des interfaces d'utilisateur différentes. Cependant, tous les modules IP doivent fournir un ensemble minimum de services pour garantir que toutes les mises en œuvre d'IP peuvent prendre en charge la même hiérarchie de protocoles. Cette section spécifie les interfaces fonctionnelles requises pour toutes les mises en œuvre d'IP.

Le protocole Internet a d'un côté une interface avec le réseau local et de l'autre côté une interface soit avec un protocole de niveau supérieur soit un programme applicatif. Dans ce qui suit, le protocole de niveau supérieur ou le programme applicatif (où même un logiciel de routeur) sera appelé "l'utilisateur" dans la mesure où c'est lui qui "utilise" le module Internet. Comme le protocole Internet est fondé sur le datagramme, la mémoire ou les états maintenus entre deux transmissions de datagrammes sont réduits au minimum, et chaque appel de l'utilisateur au module de protocole Internet lui fournit toutes les informations nécessaires pour qu'IP effectue le service demandé.

Exemple d'interface de niveau supérieur

Les deux exemples d'appel suivants satisfont les exigences de communication entre l'utilisateur et le module de protocole Internet ("=>" signifie "retour"):

SEND (src, dst, prot, TOS, TTL, BufPTR, len, Id, DF, opt => result)

où :
src = adresse de source
dst = adresse de destination
prot = protocole
TOS = type de service

TTL = durée de vie
BufPTR = pointeur sur mémoire tampon
len = longueur de mémoire tampon
Id = Identifiant
DF = Ne pas fragmenter
opt = données d'option
result = réponse
OK = datagramme bien envoyé
Error = erreur dans les arguments ou erreur du réseau local

Notez que la préséance est incluse dans le TOS et les données de sécurité/compartiment sont passées comme option.

RECV (BufPTR, prot, => result, src, dst, TOS, len, opt)

dans laquelle :

BufPTR = pointeur sur mémoire tampon
prot = protocole
result = réponse
OK = datagramme bien reçu
Error = erreur dans les arguments
len = longueur de la mémoire tampon
src = adresse de source
dst = adresse de destination
TOS = type de service
opt = donnée d'option

Lorsque l'utilisateur envoie un datagramme, il exécute l'appel SEND en fournissant tous les arguments. Le module Internet, à réception de cet appel, vérifie les arguments, prépare et envoie le message. Si les arguments sont corrects et le datagramme accepté par le module réseau local, alors l'appel retourne une indication de succès. Dans le cas où soit les arguments sont mauvais, soit le datagramme est refusé par le réseau local, l'appel retourne une indication de non réussite. Sur un retour de non réusssite, un rapport raisonnable doit être fait sur la cause du problème, mais les détails d'un tel rapport sont à la discrétion de la mise en œuvre.

Lorsqu'un datagramme arrive au module de protocole Internet du réseau local, deux cas se présentent : soit un appel RECV émis par l'utilisateur est en attente, soit il n'y en a pas. Dans le premier cas, il est répondu à l'appel en attente à l'aide des données contenues dans le datagramme entrant. Dans le second cas, l'utilisateur est averti de la présence d'un datagramme lui étant destiné. Si l'utilisateur visé n'existe pas, un message d'erreur ICMP doit être renvoyé à l'émetteur, et les données détruites.

La notification à l'utilisateur pourra être faite via une pseudo-interruption ou tout mécanisme similaire approprié en fonction de l'environnement du système d'exploitation particulier de la mise en œuvre.

L'appel RECV d'un utilisateur peut donc être immédiatement satisfait par un datagramme en cours ou l'appel peut rester en instance jusqu'à ce qu'un datagramme arrive.

L'adresse de source est incluse dans l'appel SEND au cas où l'hôte émetteur disposerait de plusieurs adresses (raccordements physiques ou adresses logiques multiples). Le module Internet doit vérifier que l'adresse de source donnée est une adresse valide pour cet hôte.

Une mise en œuvre peut aussi permettre ou exiger un appel au module Internet pour indiquer son intérêt ou se réserver l'usage exclusif d'une classe de datagrammes (par exemple, tous ceux dont le champ protocole a une certaine valeur).

La présente section caractérise fonctionnellement une interface utilisateur/IP. La notation utilisée est similaire à celle de la plupart des procédures d'invocation de fonctions dans les langages de haut niveau, mais cette utilisation n'est pas destinée à exclure les appels de service de type trap (par exemple, SVC, UUO, EMT) et autres formes de communications d'intertraitement.

APPENDICE A : Exemples & Scénarios

Exemple 1 :

Voici l'exemple d'un datagramme transmettant le minimum de données possible :

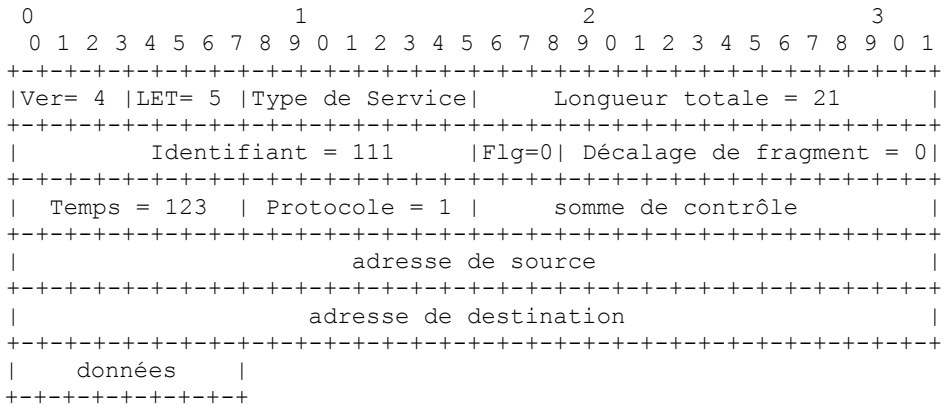


Figure 5 : Exemple de Datagramme Internet

Notez que chaque marque vaut pour une position binaire.

Cet exemple donne le datagramme de la version 4 du protocole Internet ; l'en-tête Internet est formé de 5 mots de 32 bits, et la longueur totale du datagramme est de 21 octets. Ce datagramme est un datagramme complet (pas un fragment).

Exemple 2 :

Dans cet exemple, nous exposons d'abord un datagramme Internet de taille modérée (452 octets de données), puis deux fragments Internet qui pourraient résulter de la fragmentation de ce datagramme si la taille maximale de transmission admise était 280 octets.

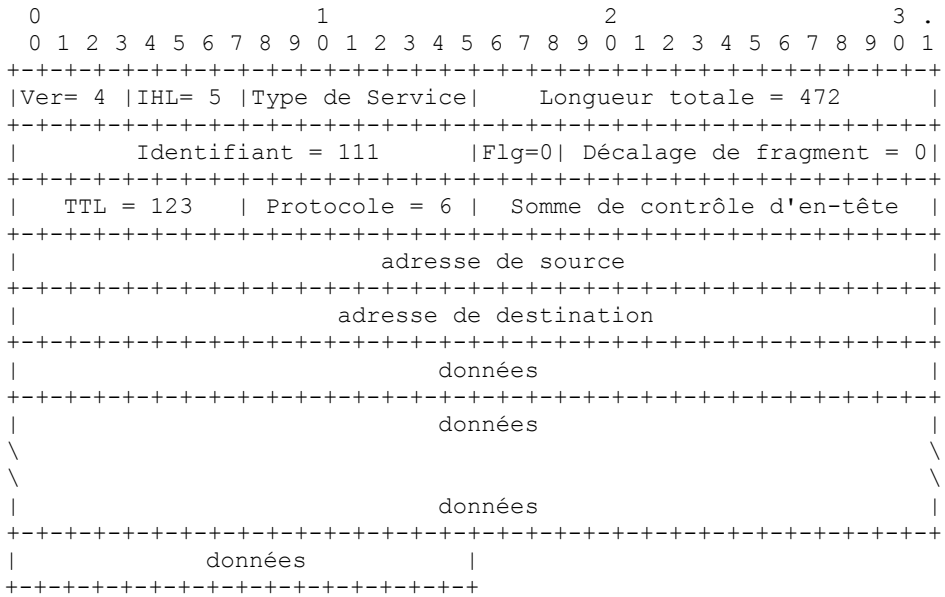
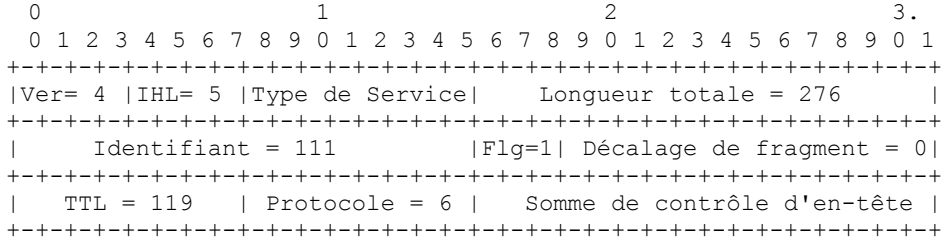


Figure 6 : Exemple de datagramme Internet

Et maintenant le premier fragment obtenu en coupant les données précédentes après le 256ème octet de données.



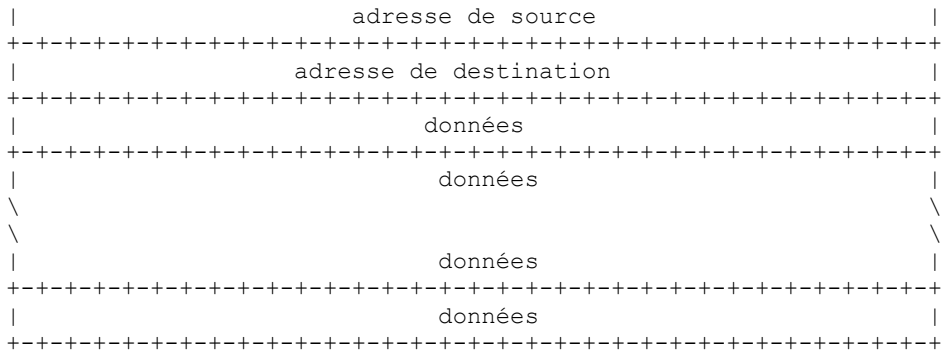


Figure 7 : Exemple de Fragment Internet

Et le second fragment.

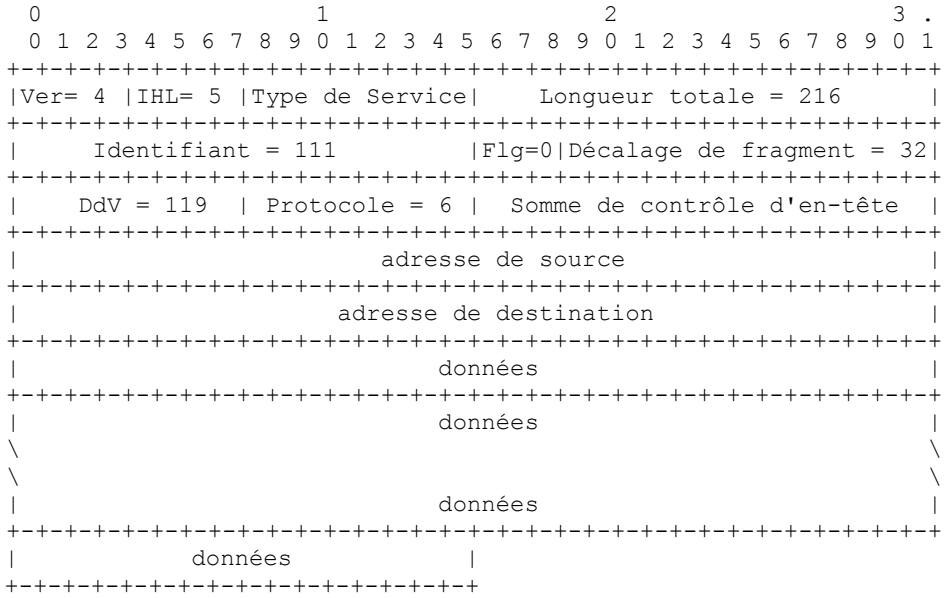
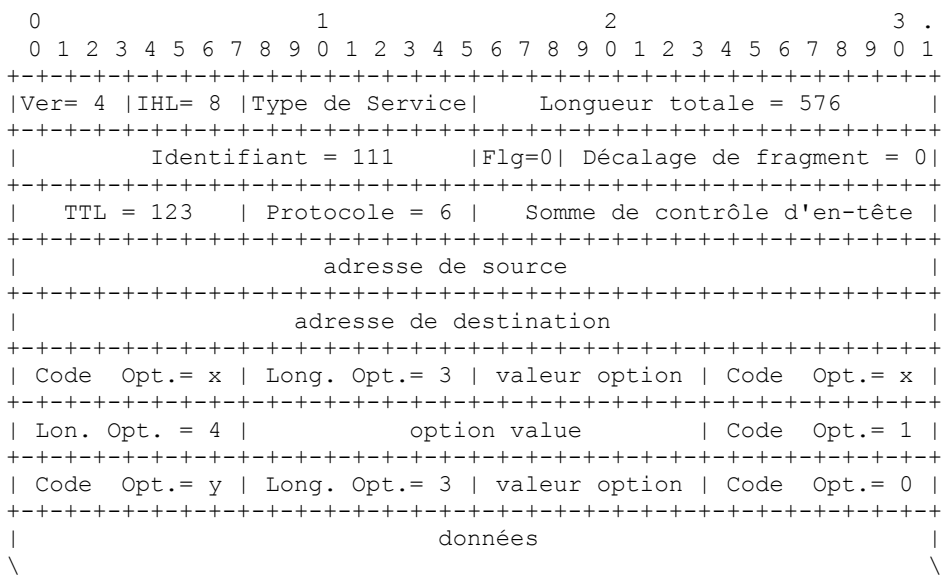


Figure 8 : Exemple de dernier fragment Internet

Exemple 3 :

Voici un exemple de datagramme contenant des options :



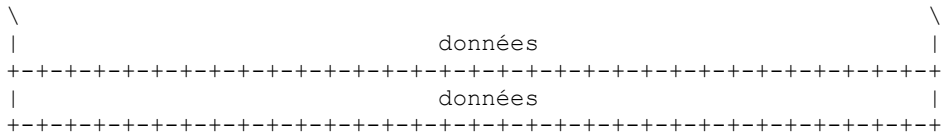


Figure 9 : Exemple Internet Datagramme

APPENDICE B : Ordre de transmission des données

L'ordre de transmission de l'en-tête et des données décrites dans ce document se résout au niveau de l'octet. Chaque fois qu'un datagramme fait apparaître un groupe d'octets, l'ordre de transmission de ces octets est l'ordre normal dans lequel on les lit en français. Par exemple, dans le schéma ci-dessous les octets sont transmis dans l'ordre de leur numérotation.

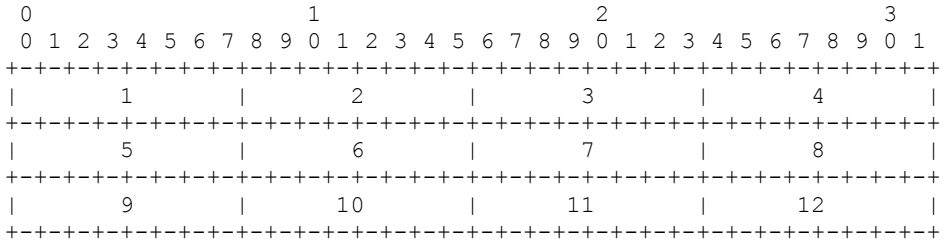


Figure 10 : Ordre de transmission des octets

Chaque fois qu'un octet représente une quantité numérique, le bit le plus à gauche dans le schéma ci-dessous est celui de plus fort poids. Ici, le bit noté 0 est le bit de plus fort poids. L'exemple suivant montre le codage de la valeur 170 (décimale).

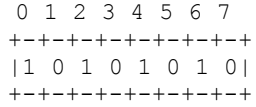


Figure 11 : Conventions sur la signification des bits

De même, chaque fois qu'un champ multi-octets représente une quantité numérique, le bit le plus à gauche de tout le champ est celui de plus fort poids. Lorsqu'un champ multi-octets est transmis, l'octet de plus fort poids est transmis en premier.

GLOSSAIRE

1822
BBN Report 1822, "*A Specification of the Interconnection of a Host and an IMP*". La spécification de l'interface entre un hôte et ARPANET.

Adresse Internet
L'adresse sur 4 octets (32 bits) d'une source ou d'une destination composée d'un champ Réseau et d'un champ Adresse locale.

Adresse Locale
L'adresse d'un hôte dans un réseau. La transpositionr réelle d'une adresse Internet locale en adresse physique d'hôte est assez libre, permettant des affectations non bijectives.

DF *(Don't fragment)*
Le bit Ne pas fragmenter porté dans le champ Fanions.

MF *(More Fragments)*
Fanion Fragment à suivre, porté dans le champ Fanions de l'en-tête Internet.

Fanions *(flags)*
Champ d'en-tête Internet portant divers fanions de contrôle.

Bourrage *(Padding)*
Le champ Bourrage de l'en-tête Internet est utilisé pour s'assurer que les données commencent sur le début d'un mot de 32 bits. Le bourrage est fait de zéros.

Datagramme Internet
L'unité de données échangée entre une paire de modules Internet (incluant l'en-tête Internet).

Destination
L'adresse de destination, un champ d'en-tête Internet.

TTL *(Time-to-live)*
Durée de vie

Durée de Vie
Champ d'en-tête Internet qui donne la durée de vie maximale pendant laquelle un datagramme peut exister dans le réseau.

En-tête
Informations de contrôle au début d'un message, d'un segment, d'un datagramme, d'un paquet ou bloc de données.

Fragment Internet
Une portion des données d'un datagramme Internet associée à un en-tête.

Décalage de fragment *(Fragment Offset)*
Ce champ d'en-tête Internet indique la position d'un fragment dans le datagramme non fragmenté.

GGP *(Gateway to Gateway Protocol)*
Protocole de routeur à routeur, le protocole utilisé principalement entre les routeurs pour contrôler l'acheminement et les autres fonctions de routeur.

ICMP *(Internet Control Message Protocol)*
Protocole de message de commande Internet, mis en œuvre dans le module Internet, ICMP est utilisé depuis les routeurs vers les hôtes et entre hôtes pour le rapport des fautes et faire des suggestions d'acheminement.

Identification
Un champ d'en-tête Internet portant la valeur d'identification allouée par l'émetteur et destinée à aider au réassemblage des fragments d'un datagramme.

IHL *(Internet Header Length)*

Champ d'en-tête Internet Longueur d'en-tête Internet qui est la longueur de l'en-tête en mots de 32 bits.

IMP *(Interface Message Processor)*
Processeur de message d'interface, l'élément de commutation de paquet du réseau ARPANET.

Leader ARPANET
Les informations de contrôle sur un message ARPANET à l'interface hôte-IMP.

Longueur totale
Le champ d'en-tête Internet Longueur totale donne la longueur totale du datagramme en octets, y compris données et en-tête.

Message ARPANET
L'unité de transmission entre un hôte et un IMP dans ARPANET. La taille maximum est d'environ 1012 octets (8096 bits).

Module
Une mise en œuvre, en général logicielle, d'un protocole ou d'une autre procédure.

NFB *(Number of Fragment Blocks)*
Le nombre de blocs de fragment dans la portion données d'un fragment Internet. C'est à dire, la longueur d'une portion de données mesurée en "mots" de 8 octets.

Octet
Huit bits.

Options
Le champ Options de l'en-tête Internet peut comporter plusieurs options, chaque option pouvant être longue de plusieurs octets.

Paquet ARPANET
L'unité de transmission utilisée en interne dans l'ARPANET entre les IMP. La taille maximum est d'environ 126 octets (1008 bits).

Protocole
Dans ce document, l'identifiant du protocole de niveau immédiatement supérieur *(à qui sera délivré le datagramme)* ; champ d'en-tête Internet.

Reste
La partie adresse locale d'une adresse Internet *(donnant l'adresse locale de la machine)*.

Segment TCP
L'unité de données échangée par les modules TCP (avec un en-tête TCP).

Source
L'adresse de source, champ d'en-tête Internet.

TCP *(Transmission Control Protocol)*
Protocole de commande de transmission : Un protocole d'hôte à hôte pour une communication fiable dans des environnemets internet.

TFTP *(Trivial File Transfer Protocol)*
Protocole trivial de transfert de fichier : Un protocole simple de transfert de fichiers fondé sur UDP.

TOS *(Type-Of-Service)*
Type de Service

Type de Service
Champ d'en-tête Internet qui indique le type (ou la qualité) du service pour ce datagramme Internet.

UDP *(User Datagramme Protocol)*
Protocole de datagramme d'utilisateur : Un protocole de niveau utilisateur pour des applications orientées transaction.

Utilisateur
L'utilisateur du protocole Internet. Celui-ci peut être un module de protocole de niveau supérieur, un programme d'application, ou un programme de routeur.

Version
Le champ Version indique le format de l'en-tête Internet.

RÉFÉRENCES

[1] Cerf, V., "Catenet Model for Internetworking", Information Processing Techniques Office, Defense Advanced Research Projects Agency, IEN 48, juillet 1978.
[2] Bolt Beranek and Newman, "Specification for the Interconnection of un Host and an IMP", BBN Technical Report 1822, révisé mai 1978.
[3] Postel, J., "Protocole de message de commande Internet – Spécification du programme DARPA Internet", RFC 792, USC/Information Sciences Institute, septembre 1981.
[4] Shoch, J., "Inter-Network Naming, Addressing, and Routing", COMPCON, IEEE Computer Society, Fall 1978.
[5] Postel, J., "Transpositions d'adresse", RFC 796, USC/Information Sciences Institute, septembre 1981.
[6] Shoch, J., "Packet Fragmentation in Inter-Network Protocols," Computer Networks, v. 3, n. 1, février 1979.
[7] Strazisar, V., "How to Build a Routeur", IEN 109, Bolt Beranek and Newman, août 1979.
[8] Postel, J., "Transpositions de service", RFC 795, USC/Information Sciences Institute, septembre 1981.
[9] Postel, J., "Numéros alloués", RFC 790, USC/Information Sciences Institute, septembre 1981. *(Obsolète, voir www.iana.org)*