



LICENCE 3 DE SCIENCES, MENTION INFORMATIQUE

RÉSEAUX ET PROTOCOLES

# MODÈLE DE DOCUMENT POUR UN RAPPORT D'ÉTUDE

Présenté par  
Julien MONTAVONT  
montavont@unistra.fr

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Concepts et utilisation</b>	<b>5</b>
2.1	En-tete IPv4 . . . . .	5
<b>3</b>	<b>Suite de protocole</b>	<b>8</b>
3.1	ARP . . . . .	8
3.1.1	Exemple . . . . .	8
3.1.2	Entête . . . . .	8
3.1.3	Fonctionnement . . . . .	8
3.1.4	ACD . . . . .	9
3.2	ICMP . . . . .	9
3.2.1	Message de type 3: Unreachable Destination . . . . .	9
3.2.2	Message de type 4: . . . . .	10
3.2.3	Message de type 11: Time Exceeded . . . . .	10
3.2.4	Message de type 5: . . . . .	10
3.2.5	Message de type 8 et 0: . . . . .	11
3.3	IGMP . . . . .	11
3.4	DHCP . . . . .	11
<b>4</b>	<b>Passage de IPv4 à IPv6</b>	<b>13</b>
4.1	Raison du passage de IPv4 à IPv6 . . . . .	13
4.1.1	Problème posé par IPv4 . . . . .	13
4.1.2	Solutions . . . . .	13
4.2	Différence entre IPv4 et IPv6 . . . . .	13
<b>5</b>	<b>Conclusion</b>	<b>13</b>
	<b>Références</b>	<b>13</b>

## 1 Introduction

A la fin des années 1960 il y eu une forte demande de la part des universités et centres de recherche aux Etats-Unis pour la création d'un réseau permettant d'une part, d'interconnecter les différentes structures afin de partager les informations et d'autre part, de faire des expérimentations sur les réseaux. C'est dans le but de faire ce réseau que l'ARPA a créé l'ARPANET qui est l'ancêtre d'internet. En effet, c'est à partir des principes de ce réseau qu'a été créé internet. L'ARPA (Advanced Research Project Agency ) est une agence de recherche créée par le département américain de la défense en 1957 afin de développer de nouvelles technologies à usage militaire. L'ARPA a mis en place en 1969 le premier réseau mondial qui a été appelé ARPANET ( Advanced Research Project Agency NETwork ). Ce réseau a été conçu de façon a ne pas dépendre d'un centre névralgique qui aurait pu être détruit en cas d'attaque nucléaire. Ce réseau a été constitué comme une toile reliant plusieurs serveurs. Chaque serveur est un noeud et peut stocker, traiter ou servir de relais. Ainsi, il existe plusieurs chemins pour accéder à un noeud et lorsqu'un noeud est hors service, il est toujours possible de rejoindre le noeud destinataire en passant par un autre chemin. Avant ARPANET, la communication réseau était également basée sur la communication par circuit électrique dont les informations étaient envoyées en continue dans un seul morceau. Ce réseau est le premier qui fonctionne sur la base de paquets. Le principe de communication par paquet est de découper l'information à transmettre en de plus petits paquets qui peuvent chacun prendre un chemin différent pour arriver à desti-

nation. Ce réseau se développa et il compta 23 noeuds en 1971 et en 1977 il en compta 111. Afin d'uniformiser ce réseau, Vint Cerf et Bob Kahn ont introduit la première version du protocole TCP. Historiquement, les protocoles IP constituaient la partie du protocole TCP qui s'occupe de la transmission en mode sans connexion. La transmission en mode sans connexion est une transmission de donnée dans laquelle chaque paquet contient l'adresse de destination. Ceci permet une transmission du paquet sans que les deux hôtes soient obligés d'établir une connexion auparavant. Cette version est ce qu'on aurait pu nommer l'IPV1 et elle est documentée dans la RFC 675. Cette version fut modifiée et publiée en 1977. Elle correspond à la deuxième version de TCP (IPV2). Initialement, le protocole TCP avait deux fonctions. ( as a host level end to end protocol, and to serve as an internet packaging and routing protocol. These two things should be provided in a layered and modular way. I suggest that a new distinct internetwork protocol is needed, and that TCP be used strictly as a host level end to end protocol.") Premièrement, il devait permettre une transmission fiable d'informations entre deux hôtes. Il devait également servir en tant que protocole de routage et de packaging. Cependant, pour être cohérent avec le modèle en couche, qui différencie la fiabilité (couche transport) et le routage (couche réseau), il fut décidé en 1978 de diviser le protocole TCP. Le protocole TCP ne s'occupe maintenant plus que de la partie transport. La partie réseau a été prise en charge par les protocoles IP. C'est finalement le 1er janvier 1983 que l'ARPANET adopte les protocoles TCP/IP et donc l'IPV4.

La technique de classe d'adressage IP ( classful network ) est une méthode utilisée de 1981 à 1993 pour allouer des adresses IPV4. Il a été défini en 1981 qu'une adresse IP est divisée en deux parties: une partie qui sert à identifier le réseau et une partie qui sert à identifier une interface sur ce réseau. Dans cette méthode une adresse IP est divisée en 5 plages d'adresses IP et sont appelées classes. Ils sont organisés comme dans le tableau ci-dessous :

Classe	Bits de départ	Début	Fin	Masque de sous-réseau par défaut
Classe A	0	0.0.0.0	127.255.255.255	255.0.0.0
Classe B	10	128.0.0.0	191.255.255.255	255.255.0.0
Classe C	110	192.0.0.0	223.255.255.255	255.255.255.0
Classe D	1110	224.0.0.0	239.255.255.255	Non défini
Classe E	1111	240.0.0.0	255.255.255.255	Non défini

Chaque classe a un certain nombre d'octets servant à identifier le réseau. Une adresse IP de classe A a un identificateur de réseau sur 1 seul octet. Une adresse IP de classe B sur 2 octets et une de classe C sur 3 octets. Les adresses IP de classe D et E correspondent à des adresses IP particulières. Les réseaux des différentes classes utilisent un certain nombre d'octets pour identifier le réseau. Ils ont donc un nombre différent d'octets restants qu'ils peuvent donner à des interfaces. Pour déterminer à quelle classe appartient une adresse IP il suffit de regarder les premiers bits de l'adresse. Afin d'avoir un niveau supplémentaire, grâce auquel on gagne en flexibilité et en efficacité dans l'attribution d'adresse à l'intérieur d'une classe, on a introduit le concept de sous-réseau. Celui-ci introduit un nouveau numéro entre le numéro de réseau et le numéro d'hôte. Grâce aux sous-réseaux on peut par exemple diviser une adresse de classe B en 256 sous-réseaux pouvant chacun avoir 256 interfaces connectées. On utilise un masque de sous-réseau pour obtenir la partie réseau de l'adresse IP. Le masque de sous-réseau est obtenu en mettant tous les bits de la partie réseau à 1 et tous les bits de la partie interface à 0. Lorsque deux adresses IP appartiennent au même sous-réseau, elles ont en commun les bits identifiants ce sous-réseau. Pour déterminer si 2 interfaces appartiennent au même sous-réseau, on les compare donc d'abord au masque de sous-réseau puis on les compare entre elles. Cependant, ce système d'adressage a un grand inconvénient. En effet, il n'existe que 4 classes différentes et donc 4 types de réseaux de taille différentes. Cela conduit souvent à de grand gaspillage d'adresse. Par exemple, lorsqu'une entreprise souhaite une adresse IP. Si celle-ci possède 2000 interfaces, une adresse de classe C (2 hôtes possibles) ne sera pas suffisante. Une adresse de classe B sera par contre largement trop grande (2 hôtes possibles). C'est à cause de ce problème de gaspillage et du manque d'adresses IP que l'on est passé au Classless InterDomain Routing (CIDR).

## 2 Concepts et utilisation

### 2.1 En-tête IPv4

Un packet IPv4 est précédé par un en-tête ayant une longueur minimale de 20 octets (dans les cas aucune option supplémentaire a été spécifiée). La figure suivante montre le contenu de l'en-tête d'un packet IPv4.

TODO FIGURE

Comme on peut voir dans la figure ci dessus, un en-tête IPv4 est composé par 13 champs. En réalité nous verrons plus loin que cet en-tête peut, quand il est nécessaire, contenir un champ additionnel qui servira à spécifier quelque option qui n'est pas présente dans les 13 champs ci dessus.

Commençons par voir plus en détail les 13 champs d'un en-tête IPv4 standard:

**Version** Cette champ occupe les premiers 4 bits de l'en-tête IPv4. Il est utilisé pour déterminer le type de protocole utilisé par la couche réseau (couche 3). Dans le cas de IPv4 cet champs contiendra toujours la valeur 4, qui justement identifie le protocole IPv4.

La position de ce champ dans l'en-tête n'est pas casuelle. En effet pour connaître la positions des autres champs de l'en-tête il faut d'abord savoir quel est le protocole utilisé et donc le type de en-tête. En pratique dans la plus part des cas cet champ n'est pas très utile, car le protocole à utiliser pour la couche 3 est souvent spécifié dans l'en-tête du protocole de la couche liaison.

**IHL** Le champ IHL spécifie la taille de l'en-tête IPv4, en fait IHL est l'acronyme de Internet Header Length. Bien entendu, en disant cela on souligne un concept important à propos du protocole IPv4: la taille de l'en-tête n'est pas fixe.

La taille de l'en-tête est exprimée en blocs de 32 bits. Étant donné une taille de 4 bits pour le champ IHL, la longueur maximale d'un en-tête IPv4 est de 15 blocs de 32 bits,

qui correspond a 60 octets. Comme l'en-tete IPv4 a une taille minimale de 20 octets (160 bits), le champ IHL ne peut pas contenir une valeur inferieure a 5.

**Type of Service** Le champ Type of Service, mieux connu avec l'acronyme ToS, est utilise pour specifier la qualite de service souhaite pour l'envoi d'un packet IPv4. Cet champ occupe un octet de l'en-tete et il se compose en trois parties. Une premiere partie de 3 bits permet d'indiquer la precedence avec la quelle le packet doit etre traite, les 3 bits apres sont utilise pour specifier certaines caracteristiques du service, notamment: le temp, le debit et la fiabilite. Enfin l'emploi des 2 derniers bits n'a pas ete specifie et leur usage a ete laisse libre pour des implementationes futures.

En realite l'histoire de cet champs est bien plus longue et complexe que ca, car en pratique la facon d'utiliser cet champs a ete modifie plusieurs fois au cours des annees.<sup>1</sup> Cette manque de stabilite a par fois cause une certaine confusion dans les implementations.<sup>2</sup>

Aujourd'hui les 8 bits du champ ToS sont utilise par le mecanisme DiffServ (Differentiated Services). Cet systeme utilise les premieres 6 bits du champ ToS (DSCP - Differentiated Services Code Point) pour marquer chaque paquet comme appartenant a un niveau de priorite et une classe de service. Chaque classe determine le type de traitement que on souhaite demander pour le paquet aux routers au long du chemin (PHB - Per-Hop behaviour), toutefois le service offert par chaque router est fortement lie a sa configuration.<sup>3</sup> Les dernieres 2 bits du champ ToS sont utilise pour l'extension ECN (*Explicit Congestion Notification*). Cette extension, proposee par RFC2481 et introduite deux annees apres avec le RFC3168, ajoute un systeme de controle de la congestion du trafic reseau. Dans le cas d'une saturation de la reseau cet champ est utilise pour notifier cet probleme et demander a le dispositif emetteur une reduction du rythme au quel les packets sont envoye, avec l'objectif de reduire l'attente et la perte de packets.

**Total length** Comme le suggere le nom, ce champs est utilise pour indiquer la taille totale du packet IPv4: en tete plus donnees. Le champ *Total length* est defini sur 16 bits, ceci permet de indiquer une valeur compris entre 0 et 65,535 octets. Comme l'en-tete est compris dans la longueur totale d'un packet cette valeur ne serait jamais inferieur a 20 (taille minimale d'un en-tete IPv4 en octets). RFC 791 impose a toutes les dispositifs d'une reseau IPv4 la capacite de recevoir des packets jusqu'a une taille de 576 octets, cette prerogative permet de eviter une excessive fragmentation.

**Identification** Cet en-tete (sur 16 bits) permet d'identifier les fragments appartenent au meme packet.

**Flags** Les 3 bits du champs Flags sont utilise pour gerer la fragmentation d'un packet. Un de ces bit est employe pour indiquer si le paquet peut etre fragmente ou non. Cet bit, appelle DF (*Don't Fragment*), doit etre pris en consideration par les routers dans le chemin pour decider si un paquet trop grand pour etre transmis peut etre retransmis sous forme de fragments plus petits ou rejete. Un autre bit, appelle MF (*More Fragments*), indique si le paquet est suivi par d'autres fragments. Le bit MF est mis a 0 dans le dernier fragment ou dans des paquet qui n'ont pas ete fragmente.

---

<sup>1</sup>L'utilisation des 8 bits du champ ToS a ete redefinie par cinq standard differents (plus divers standards experimentals). Les documents presentant ces standard sont mentionne dans le chapitre "Historical Definitions for the IPv4 TOS Octet" du RFC 3168

<sup>2</sup>Comme le souligne le RFC 3260 "At least one implementor has expressed confusion about the relationship of the DSField, as defined in RFC 2474, to the use of the TOS bits, as described in RFC 1349"

<sup>3</sup>"The DiffServ standard does not specify a precise definition of "low," "medium," and "high" drop probability. Not all devices recognize the DiffServ (DS2 and DS1) settings; and even when these settings are recognized, they do not necessarily trigger the same PHB forwarding action at each network node. Each node implements its own response based on how it is configured." - Implementing Quality of Service Policies with DSCP <http://www.cisco.com/c/en/us/support/docs/quality-of-service-qos/qos-packet-marking/10103-dscpvalues.html>

Un des trois bits de ce champs n'est pas actuellement utilise mais il a ete reserve pour applications futures possibles. <sup>4</sup>

**Fragment Offset** Lorsque un paquet a ete fragmentee cet en-tete est utilisee pour determiner la position (offset) d'un fragment par rapport a les donnees du paquet reassemble. Le decalage de chaque fragment est exprime en blocs de huit octets (ou 64 bits). Le champ Fragment Offset utilise 13 bits de l'en-tete IPv4, cela permet un offset maximale de 65,528 octets.<sup>5</sup> Etant donne que la flag MF (*More Fragments*) doit etre mise a zero lorsque si un paquet n'est pas fragmentee ou si il est le dernier fragment d'un paquet plus grand, l'unique difference entre ce deux types de paquets est le valeur de le champ Fragment Offset que, dans le cas d'un paquet pas fragmentee, est toujours zero.

**Time to Live** Cet champ determine le nombre maximal de fois qu'un paquet peut etre retransmis, il est utilise pour empecher que un paquet puisse etre retransmis a l'infini. Chaque router au long du chemin d'un paquet est tenu a detruire un paquet si le valeur du TTL (*Time to Live*) est zero ou decrementer cet champs par le nombre de secondes que le paquet passe en attente d'etre transmis.

En theorie le TTL indique le nombre de secondes pendant le quels un paquet peut continuer a etre retransmis dans une reseau ,mais , etant chaque router toujours tenu a decrementer cet champ de au moins 1 (meme si le paquet a ete retransmis en moin qu'un seconde) et considerant les performances des routers d'aujourd'hui, en pratique le TTL indique le nombre maximum de router que un paquet peut rencontrer au long de son chemin.

Le space reservee au TTL dans l'en-tete IPv4 est de un octet ce que comporte un TTL maximum de 255.<sup>6</sup>

Quand un paquet a ete detruit ensuite a l'expiration du TTL, le router qui a detruit le paquet peut decider de envoyer un message d'erreur a l'emetteur du paquet detruit. Cet type de message (ICMP Time exceeded) est utilise par utils comme *traceroute* pour decouvrir, approximativement, le chemin d'un packet IP.

**Protocol** Chaque paquet IPv4 specifie le protocole utilise par les donnees transmises: cela est l'objectif de ce champs de 8 bits

**Header Checksum** Cet champ contient une somme de controle et est utilise pour detecter des erreurs dans l'en-tete IPv4. Le valeur de cet champs est recalcule at chaque retransmission<sup>7</sup>: si il ne correspond pas avec celui presente dans l'en-tete le paquet est detruit.

**Adresse source et Adresse destination** Les adresses de chaque paquet IPv4 (soit l'adresse source que l'adresse de destination d'un paquet) sont represente sous forme d'une suite de 32 bits.

L'adresse source de chaque packet represente dans la plus part des cas l'adresse logique<sup>8</sup> de la machine qui a envoye le paquet (a laquelle il faudra eventuellement repondre donc). Dans certains cas specifiques cette adresse ne correspond pas a celui de la machine qui a

---

<sup>4</sup>Cet bit a ete aussi le protagoniste d'un des plus connu poissons d'avril presentee par le IETF. Pour faciliter les taches des systems de filtrage le RFC 3514 propose d'utiliser ce bit pour etiqueter paquets mailveillant, a ce titre tous les paquets etant envoye avec ce bit (renomme "*Evil Bit*") mis a 1 seront mis a la poubelle.

<sup>5</sup>En pratique un tel offset n'est jamais utilisee car, en ajoutant un en-tete minimale de 20 octets, la taille totale du paquet reassemble depaserait la longueur maximale d'un paquet IPv4.

<sup>6</sup>Le RFC 1700 recommande un valeur par defaut de 64.

<sup>7</sup>Cela est necessaire car le TTL est decrementee at chaque retransmission et un changement de l'en-tete comporte un valeur different dans la somme de controle

<sup>8</sup>Il ne faut surtout pas oublier la difference entre un adresse physique, comme par exemple un adresse MAC (qui est lie a l'hardware et est donc unique pour chaque machine), et un adresse logique, comme par exemple un adresse IP (qui peut changer et identifie une machine dans une reseau en particuliere).

envoie le paquet, c'est par exemple ce qui se passe dans une requete *ARP probe* la ou le valeur de l'adresse de la machine source est 0.0.0.0 (ce qui represente un adresse indefini<sup>9</sup>) car elle n'a pas encore determine son adresse IP.

L'adresse de destination d'un paquet IPv4 identifie la machine vers laquelle le paquet doit etre expedie. Comme dans le cas de l'adresse source, aussi l'adresse destination peut contenir des valeur speciaux. En effet certains valeurs puvent etre utilise par exemple pour identifier plusieurs machines (adresses multicast), toutes le machines d'une reseau (adresse de broadcast) ou la machine actuelle (adresse de loopback).

Une description plus detaile des mecanismes lies aux adresses IPv4 est propose dans le chapitre de ce rapport.

**Options** Cet champ n'est pas obligatoire et donc il peut ne pas etre present dans un en-tete IPv4. La presence de cet champ est determine par le valeur du IHL: lorsque cette valeur indique une taille de l'en-tete IPv4 superieure a la taille minimale (20 octets), l'en-tete contiens des options. Etant donne que un en-tete IPv4 peut avoir une taille maximale de 60 octets (le valeur du IHL est egal a 15), le champ Options peut occuper 40 octets au meximum.

Ce champ est ne pour etendre les possibilites de IPv4 en ajoutant par exemple des fonctions de debuggage, de securite, etc .... Aujourd'hui il y a quelque dizaine de options qui ont ete specifie<sup>10</sup> (si on considere aussi les options experimentales) mais peu entre eux sont reelement utilise. Entre les options les plus connues on retrouve les options: SEC, TR, MTUP et MTUR, TS et RR.

## 3 Suite de protocole

### 3.1 ARP

ARP (Address resolution protocol) est un protocole à cheval sur la couche 2 et la couche 3. La fonction principale d'ARP est de faire la conversion entre les adresses de niveau 2 et de niveau 3. Cela est très utilisé étant donné que les hôte connaissent souvent les adresses IP de leur destinataire, mais rarement l'adresse de niveau 2 de ce destinataire ou de la passerelle à contacter pour joindre le destinataire.

#### 3.1.1 Exemple

Dans la partie qui suit nous allons nous placer dans l'exemple suivant. TODO LAN exemple

#### 3.1.2 Entête

#### 3.1.3 Fonctionnement

Prenons l'exemple où A veut envoyer un message à B. A connait l'adresse IP de B. Donc A va préparer son paquet qu'il va envoyer à B, avec son adresse IP en source et l'adresse IP de B en destination. La paquet passe dans la couche liaison, il va être enpaqueté dans une trame de niveau 2. Cette trame aura comme adresse source l'adresse de niveau 2 de A, mais à ce moment il

---

<sup>9</sup>Notex que la signification de l'adresse 0.0.0.0 est lie a la facon dont il est utilise. En general il indique *aucune adresse en particulier*. Dans la plus part des cas cet adresse est utilise pour indiquer un de ces valeurs: l'adresse de la machine courant (c'est l'adresse de loopback), n'importe quel adresse ou reseau (c'est le cas de la route par default dans une table de routage), un adresse indefini ou bien une combinaison des possibilites precedentes (c'est le cas d'une requete *ARP probe* ou bien d'une requete *DHCP Discovery* ou *DHCP Request*, ou en fait l'adresse source 0.0.0.0 indique un adresse indefinie mais aussi l'adresse de la machine actuelle, que del reste n'est pas encore defini...).

<sup>10</sup><http://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml>



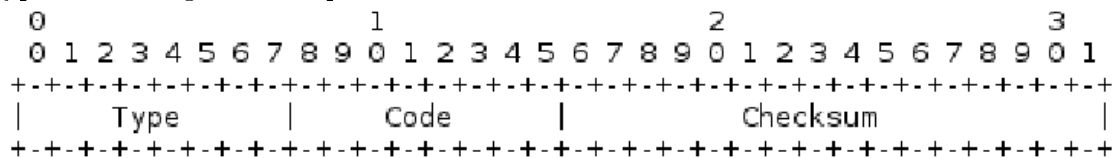
ne peut pas compléter l'adresse destination de la trame: en effet, il ne connaît l'adresse de niveau 2 du destinataire. Le paquet reste bloqué en couche 2 et ne peut pas être envoyé au destinataire. Comment obtenir l'adresse de niveau 2 du destinataire? Le protocole ARP est capable de faire cette translation.

### 3.1.4 ACD

Adresse conflict detection

## 3.2 ICMP

ICMP (Internet Control Message Protocol) est un protocole de niveau 3 faisant partie intégrante du protocole IPv4. Il permet de transmettre des informations de contrôle et d'erreur. Les messages ICMP sont empaquetés dans des paquets IP, ils disposent donc d'un entête de paquet IP. Cet entête est le même que pour tous les autres entêtes de paquet d'IPv4. Deux champs sont intéressants dans le cas d'un paquet ICMP, les champs Protocol et Type of service. Le champ Protocol est mis à la valeur 1 pour dire que le paquet contient un message ICMP, et le champ ToS est mis à 0 //TODO(pourquoi 0?)//. Après le header du paquet IPv4, commence la partie data qui contient le message ICMP. Ce message contient des champs différents en fonction du type de message à passer. Cependant les trois premiers champs sont toujours les mêmes. Les types de messages ICMP qui vont suivre sont décrits dans la RFC 792.<sup>11</sup>

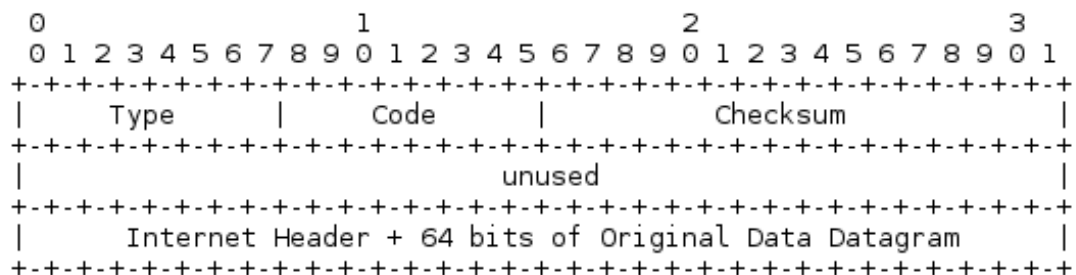


Le premier champ est celui de type. Il permet, premièrement, de donner le type du paquet et de l'information à transmettre, et deuxièmement de préciser la nature des champs qui vont suivre. En effet, comme vu plus haut, les messages contiennent des champs différents selon le type du message ICMP.

Le deuxième champ est le code. Il permet de subdiviser le type en donnant des détails plus précis.

Enfin le troisième champ est la somme de contrôle (checksum)//TODO(plage de contrôle).

Commençons avec les messages qui possèdent l'ensemble de champs le plus simple.



Les messages qui utilisent cette organisation sont les messages de type 3, 4 et 11.

### 3.2.1 Message de type 3: Unreachable Destination

Les messages de type 3 sont émis lorsqu'un paquet n'a pas réussi à joindre la destination (Unreachable destination). Cette erreur peut être due à plusieurs facteurs, et les codes permettent de préciser pourquoi le paquet n'a pas pu rejoindre sa destination.

#### Code 0

<sup>11</sup><https://tools.ietf.org/html/rfc792>

### 3.2.2 Message de type 4:

### 3.2.3 Message de type 11: Time Exceeded

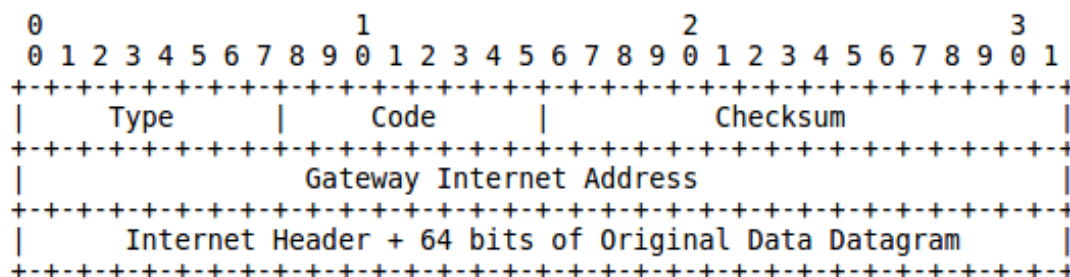
Ces messages sont envoyés lorsque le TTL d'un paquet à atteint 0. Une autre utilisation des ces messages est lorsque que le temps de ré-assemblage des fragments d'un paquet est dépassé. Ces deux cas sont distingé par le code. Ces messages ont pour destinataire l'hôte qui à envoyé le paquet qui à provoqué l'erreur.//TODO(vérifier) Le champ Internet header contient l'entête du paquet qui a été supprimé plus les 64 bits suivant celui-ci. Cela permet à l'émetteur de retrouver quel paquet à été supprimé.

**Code 0:** Le code 0 est utilisé pour indiquer que le TTL du paquet posant problème est arrivé à 0. Lorsque le TTL d'un paquet arrive à 0, celui-ci est supprimer et un message ICMP de type 11 et de code 0 est envoyer par le routeur qui à détecté le problème. Cela permet principalement d'éviter qu'un paquet sans dans une boucle et qu'il soit relayé à l'infini.

**Code 1:** Le code 1 est quant à lui utilisé pour indiquer //TODO

### 3.2.4 Message de type 5:

Les message de type 5 utilisent les entêtes ci-dessous et servent à faire de la redirection. En effet, lorsqu'un routeur détecte que le prochain routeur dans lequel va transiter le paquet se trouve dans le même réseau que l'émetteur de ce paquet, il va envoyer un message ICMP pour avertir cet hôte (et/ou le réseau) qu'il existe un chemin plus court en envoyant directement les paquets vers le prochain routeur. Ce message ICMP va avoir pour effet de modifier la table de routage interne à l'émetteur (et/ou des hôtes connecté au réseau). Concernant le paquet que le premier routeur à reçu, il va le transmettre vers sa destination.



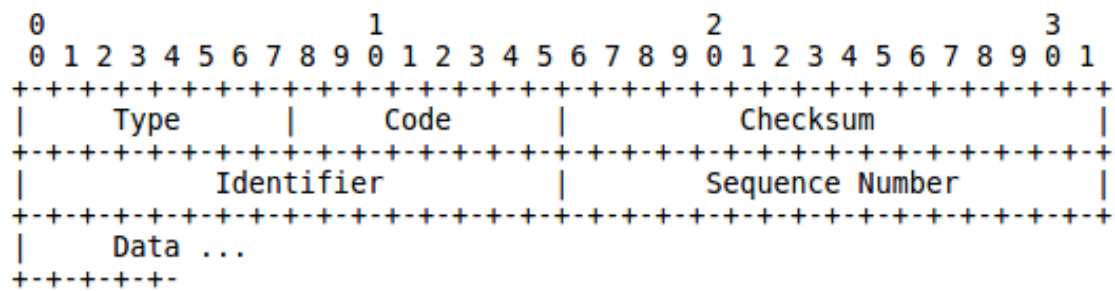
Le champ Gateway Internet Address contient la l'adresse du routeur auquel il faut faire transiter le trafic directement pour avoir un chemin de routage plus court. Le champ Internet Header contient toujours l'entête du message ayant porvoqué l'envoi du message ICMP plus les 64 bits suivant l'entête. Cela permet à (aux) hôte(s) de pouvoir modifier leur table de routage en fonction la destination que cherchait à atteindre le paquet.

**Code 0** Ce code indique que la redirection est adresser à tout le réseau de l'émetteur du paquet.

**Code 1** Ce code indique que la redirection est adresser à l'émetteur du paquet.

**Code 2** Ce code indique que la redirection est adresser à tout le réseau de l'émetteur du paquet et aux services(//TODO(préciser)).

**Code 3** Ce code indique que la redirection est adresser à l'émetteur du paquet(//TODO(préciser)).



### 3.2.5 Message de type 8 et 0:

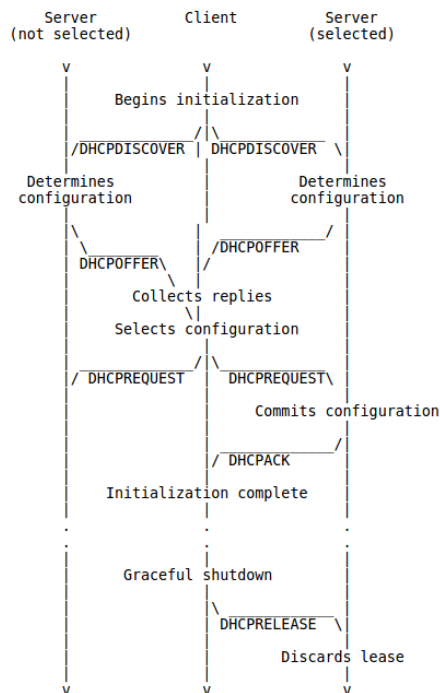
Les messages de type 8 et 0 servent à faire des envoies et des renvoies d'information. Ils utilisent pour cela l'entête ci-dessus. Les messages de type 8 font des envoies d'informations, appelé echo request. Tandis que les messages de type 0 sont envoyés en réponse aux echo request et renvoie les informations reçus de ceux-ci; ils sont appelés echo reply. Etant donnée que les echo reply sont des réponses aux echo request, l'adresse destination des echo reply est l'adresse source des echo request. Ces deux messages peuvent envoyés et reçu aussi bien par un hôte que par un routeur. Ce sont notamment les message envoyés par la commande *ping* qui permet de vérifier si l'on peut communiquer avec un hôte ou un routeur. Les champs Identifier et Sequence Number aide l'émetteur de l'echo request à associer les echos request qu'il à envoyés avec les echos reply qu'il à reçus. //TODO(qui a t-il dans data?)

## 3.3 IGMP

## 3.4 DHCP

<sup>12</sup> Le protocole DHCP (Dynamic Host Configuration Protocol) sert à l'autoconfiguration des interfaces. Plus précisément, il permet d'attribuer une adresse IP à une interface et de lui faire parvenir d'autres information essentielle pour le fonctionnement de l'interface sur le réseau. Voyons comment une interface peut ce configurer auprès d'un serveur DHCP.

<sup>12</sup>RFC 2131: <https://tools.ietf.org/html/rfc2131>



Lorsqu'une interface, qui n'a pas d'adresse IP, souhaite en recevoir une, elle va émettre un message DHCPDISCOVER en broadcast sur son réseau. Des agents DHCP peuvent faire passer ce message DHCP sur un autre réseau si le serveur DHCP (qui distribue les adresses) ne se trouve pas sur le même réseau que l'hôte qui fait la demande.

Étant donné que le message est envoyé en broadcast, tout les hôtes sur le réseau vont recevoir le message, et en particulier le ou les serveurs DHCP qui pourraient s'y trouver. Si cela est le cas, ceux-ci vont répondre avec un DHCPOFFER. Ce message contient entre autre l'adresse IP proposé pour le client souhaitant se configurer, ainsi que le masque de sous-réseau de l'adresse. A ce moment là l'adresse n'est pas encore attribuer et réservé pour l'hôte étant donné qu'il peut refuser l'offre et accepter l'offre d'un autre serveur. Si jamais l'hôte ne reçoit aucun DHCPOFFER, il va ré-émettre un DHCPDISCOVERY. Si il reçoit un ou plusieurs DHCPOFFER, l'hôte va devoir choisir une configuration qui lui est proposé. Une fois ce choix fait, il va informé les serveurs DHCP de son choix à l'aide d'un message DHCPREQUEST émis en broadcast. Ce message va contenir l'identifiant du serveur DHCP retenu ainsi que la configuration souhaité par l'hôte (adresse IP et masque de sous-réseau). Ce message peut être interprété de deux manières différentes selon le serveur:

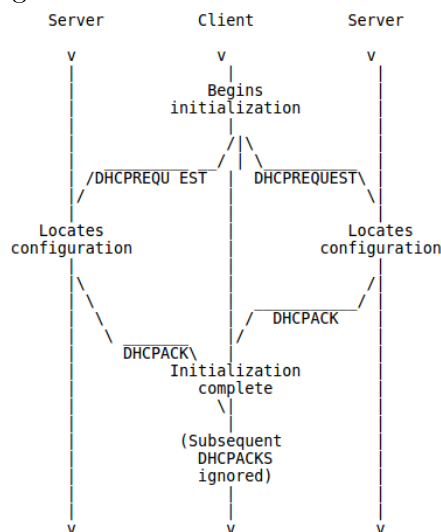
- si ce n'est pas le serveur retenu, il considère le message comme une déclinaison de l'offre.
- si c'est le serveur retenu, il va sortir l'adresse attribué l'hôte de la plage d'adresse libre pour ne plus l'attribuer à un autre hôte. Il va ensuite émettre un message DHCPACK contenant la configuration effective de l'hôte avec notamment: l'adresse IP, le masque de sous-réseau, la durée du bail, l'adresse de la passerelle par défaut et l'adresse du serveur DNS. Si pour quelque raisons le serveur n'est pas capable d'attribuer l'adresse proposé dans l'offre (par exemple si l'adresse à été attribuer entre temps), le serveur émet un DHCPNAK pour avertir l'hôte que l'adresse n'est plus disponible. L'hôte devra alors recommencer la procédure pour obtenir une adresse IP. Enfin si le serveur ne reçoit pas de message DHCPREQUEST, la procédure s'arrêtera à ce moment et l'adresse n'étant pas encore attribuer à l'hôte elle reste disponible pour être attribuer à d'autre hôte. Arrive la dernière étape. Si le client reçoit un message DHCPACK, il peut prendre en compte la configuration (adresse IP, masque de sous-réseau, DNS, passerelle par défaut et durée de bail). Il va effectuer une dernière vérification pour s'assurer que l'adresse qui lui à été attribué est bien unique sur le réseau pour éviter d'avoir deux hôte avec la même adresse. Il va pour cela utilisé le protocole ARP et la méthode de vérification vu plus haut. Si jamais l'adresse est

déjà utilisé par un autre hôte, il va envoyer un message DHCPDECLINE au serveur DHCP pour lui indiquer qu'il n'utilisera pas la configuration proposé par celui-ci, et il va recommencer la procédure pour pouvoir obtenir une nouvelle configuration.

Si jamais l'adresse proposé par le serveur est unique sur le réseau, la configuration est terminée et l'hôte peut utiliser l'adresse (durant la durée du bail de celle-ci). Dernier cas possible, si jamais le l'hôte ne reçoit pas de DHCPACK ou de DHCPNAK, il va rémettre le message DHCPREQUEST pour esperer recevoir une réponse du serveur.

//TODO algo de retransmission //TODO fonctionnement agent relais dhcp //client peut renoncer à son bail //identification des message faisant partit d'un meme echange avec client identifier, server identifier //TODO fonctionnement bail

L'hôte est donc configuré et peut utiliser son adresse. Cependant, il ne peut l'utiliser que durant la durée de son bail. Une fois le bail expiré, l'hôte ne peut plus utiliser son adresse. Lorsque l'hôte a reçu le message DHCPACK du serveur, celui-ci lui a transmis la durée du bail. De cette durée, l'hôte va en extraire deux temps noté T1 et T2. T1 correspond à la moitié de la durée du bail et T2 à 0.875 la durée du bail. Ces temps sont exprimé de manière relatif étant donnée que les horloges du serveur et de l'hôte ne sont pas synchronisées. Une fois que l'hôte a atteint le temps T1, il va chercher à contacter le serveur qui lui a attribué sa configuration avec un message DHCPREQUEST pour étendre la durée de son bail. Ce message est émis de manière unicast. A ce moment l'hôte est entré en état RENEWING. Si l'hôte reçoit un message DHCPACK du serveur lui accordant un prolongement de la durée de son bail, alors il va sommer le temps qu'il avait insérer dans le DHCPREQUEST avec la durée accordé par le serveur et qui se trouve dans le message DHCPACK. L'hôte retourne dans l'état BOUND. Cependant l'hôte n'est pas obligé d'attendre T1 pour pouvoir étendre son bail. Si jamais l'hôte ne reçoit pas de reponse DHCPACK avant l'arrivé de T2, il passe en état REBINDING. A ce moment il va émettre un DHCPREQUEST en broadcast pour espérer pourvoir étendre son bail auprès de n'importe quel serveur DHCP. Pour parer aux eventuels cas de perte de DHCPREQUEST, l'hôte va renvoyer un message une fois la moitié de la durée entre T1 et T2 passé, en état RENEWING; et une fois la moitié de la durée entre T2 et la fin du baille, en état REBINDING(et avec un minimum de temps de 60 secondes). Si malgré tout, la durée du bail venait à expirer, alors l'hôte ne possèderait plus de configuration réseau et ne pourrait plus communiquer avec d'autre hôtes. Il rentre alors en état INIT; il doit alors recommencer la procédure pour obtenir une adresse configuration.



## **4 Passage de IPv4 à IPv6**

### **4.1 Raison du passage de IPv4 à IPv6**

#### **4.1.1 Problème posé par IPv4**

//Pénurie, adressage privé/publique compliqué

#### **4.1.2 Solutions**

//NAT,IPv6

### **4.2 Différence entre IPv4 et IPv6**

## **5 Conclusion**