



LICENCE 3 DE SCIENCES, MENTION INFORMATIQUE

RÉSEAUX ET PROTOCOLES

# MODÈLE DE DOCUMENT POUR UN RAPPORT D'ÉTUDE

Présenté par  
Julien MONTAVONT  
montavont@unistra.fr

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Concepts et utilisation</b>	<b>6</b>
<b>3</b>	<b>Fonctionnement et Adressage de IPV4</b>	<b>6</b>
3.1	Notion de netid et hostid . . . . .	6
3.2	Masque de réseau . . . . .	7
3.2.1	Format du masque . . . . .	7
3.2.2	Calcul de l'adresse réseau et numéro de l'hôte . . . . .	7
3.2.3	Notation CIDR . . . . .	8
3.2.4	Adresses non utilisées . . . . .	8
3.3	Type d'adresse IP (Schéma dans pics sur comment c'est utilisé dans un ensemble de réseau ) . . . . .	8
3.4	Les classes d'adresses . . . . .	9
3.5	En-tete IPv4 . . . . .	9
<b>4</b>	<b>Suite de protocole</b>	<b>13</b>
4.1	ARP . . . . .	13
4.1.1	Cache ARP . . . . .	13
4.1.2	Entête . . . . .	13
4.1.3	Fonctionnement . . . . .	13
4.1.4	ACD . . . . .	14
4.2	ICMP . . . . .	15
4.2.1	Message de type 3 : Unreachable Destination . . . . .	16
4.2.2	Message de type 4 : . . . . .	16
4.2.3	Message de type 11 : Time Exceeded . . . . .	16
4.2.4	Message de type 5 : . . . . .	16
4.2.5	Message de type 8 et 0 : . . . . .	17
4.3	IGMP . . . . .	17
4.4	DHCP . . . . .	17
<b>5</b>	<b>Passage de IPv4 à IPv6</b>	<b>20</b>
5.1	Raison du passage de IPv4 à IPv6 . . . . .	20
5.1.1	Problème posé par IPv4 . . . . .	20
5.1.2	Solutions . . . . .	21
5.2	Différence entre IPv4 et IPv6 . . . . .	21
<b>6</b>	<b>Conclusion</b>	<b>21</b>
	<b>Références</b>	<b>23</b>

# 1 Introduction

Une forte demande de la part des universités et centres de recherche aux Etats-Unis a donné lieu à la création et la mise en œuvre d'un nouveau concept de réseau permettant d'interconnecter les différentes structures de façon efficace afin de partager les informations, et d'autre part, de faire des expérimentations sur les réseaux.

En effet jusqu'alors les réseaux informatiques utilisent les mêmes principes des réseaux téléphoniques : la commutation de circuits, ce qui n'était pas très efficace en terme de ressources et matériel<sup>1</sup>.

Pendant les années 1960 le concept de réseau à paquets commutés a été inventé et mis en pratique d'abord dans le réseau du NPL (UK National Physical Laboratory) et puis dans l'agence américaine ARPA (*Advanced Research Project Agency*)<sup>2</sup>.

Dans un réseau à commutation de paquets (*Packet Switching*) la connexion entre deux machines n'est pas continue mais elle est coupée en plusieurs paquets. L'abandon d'une connexion continue a permis de se passer de la réservation d'un lien (circuit) dédiée ce qui comporte la possibilité de envoyer et recevoir en même temps paquet vers différents destinataires (un peu comme une boîte à lettre).

Le réseau créé au sein de l'agence gouvernementale ARPA et baptisée ARPANET, est une des premières réseaux à fonctionner 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 destination. Avant ARPANET, la communication réseau était basée sur la communication par circuit électrique dont les informations étaient envoyées en continue dans un seul morceau. Dans ce sens ARPANET a posé la base à partir de laquelle l'internet a été créé.

L'ARPA (aujourd'hui DARPA : *Defence 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. ARPANET, le réseau mis en place par ARPA, a été constitué comme une toile reliant plusieurs serveurs. Chaque serveur est un nœud et peut stocker, traiter ou servir de relais. Ainsi, il existe plusieurs chemins pour accéder à un nœud et lorsqu'un nœud est hors service, il est toujours possible de rejoindre le nœud destinataire en passant par un autre chemin : une des caractéristiques plus intéressantes de ARPANET a été une certaine robustesse, ARPANET ne dépendait pas d'un centre névralgique qui aurait pu être détruit en cas d'attaque<sup>3</sup>

Ce réseau se développa et il compta 23 nœuds 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 données 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 : premièrement, il devait permettre une transmission fiable d'informations entre deux hôtes en plus 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

---

1. Dans un réseau à commutation de circuits chaque périphérique pouvait utiliser un seul lien à la fois (circuit) : lors d'une transmission un lien était réservé pour toute la durée de la communication et rendait donc la périphérique inutilisable pour des transmissions vers un différent destinataire (c'est le même principe des réseaux téléphoniques) [http://www.tcpipguide.com/free/t\\_CircuitSwitchingandPacketSwitchingNetworks.htm](http://www.tcpipguide.com/free/t_CircuitSwitchingandPacketSwitchingNetworks.htm) .

2. [http://www.livinginternet.com/i/iw\\_packet\\_inv.htm](http://www.livinginternet.com/i/iw_packet_inv.htm)

3. Il faut par contre noter que l'hypothèse qui affirme que ARPANET ait été construit dans le but de créer un réseau résistant aux attaques nucléaires a été démythifiée par le *Internet Society* : <http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet> .

de diviser le protocole TCP<sup>4</sup>. 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 :

---

4. *"We are screwing up in our design of internet protocols by violating the principle of layering. Specifically we are trying to use TCP to do two things : serve 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."*  
- IEN 2 (Comments on Internet Protocol and TCP)

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
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 (28 hôtes possibles) ne sera pas suffisante. Une adresse de classe B sera par contre largement trop grande (2<sup>16</sup> 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)[1].

## 2 Concepts et utilisation

## 3 Fonctionnement et Adressage de IPV4

Une adresse ip est un nombre de 32 bits codé sur 4 octets dont 1 octet égal à 8 bits séparés par un point. Souvent cette adresse est écrite avec des valeurs décimales. Il est aussi possible de la saisir sous forme binaire quand c'est indispensable. Chaque nombre étant compris entre 0 et 255. ou en binaire entre 00000000 et 11111111. Pour un réseau informatique, cette adresse ip est un identifiant unique attribué à chaque interface avec le réseau ip et associé à une machine. Cette adresse est unicast utiliser comme adresse source ou de destination. Il faudrait aussi préciser que le protocole IPV4 permet aussi l'utilisation des adresses multicast et broadcast en dehors des adresses unicast.

Exemple : adresse à valeur décimale : 212.217.0.1 => correspond sous sa forme binaire à : 11010100.11011001.00000000.00000001

### 3.1 Notion de netid et hostid

Une adresse ipv4 est la composition de deux parties distinctes : une première partie de l'adresse qui identifie le réseau communément appelé netid, elle est la partie gauche de l'adresse et désigne le réseau auquel appartient les ordinateurs (hôte). Une seconde partie de l'adresse qui identifie le numéro de l'hôte appelé host-ID, elle est la partie droite de l'adresse et désigne l'ensemble des hôtes du réseau.

<-----4 octets-----> Net ID Host ID Identifiant réseau

Identifiant de l'hôte

Pour connaître la limite entre netID et hostID, il faudrait connaître d'abord le masque du réseau que nous introduirons la suite.

**Remarque :** on distingue alors deux situations qui peuvent être soit des échanges directes ou indirectes. Les différents matériels communiquent entre eux dans la mesure qu'ils soient tous sur le même réseau IP (netid). Et peuvent être reliés physiquement. Et en deuxième cas l'échange indirect ou les matériels ne sont pas sur le même réseau, passage obligatoire à travers un routeur pour réaliser une communication externe.

## 3.2 Masque de réseau

Une adresse masque est sous forme de 32 bits, elle est utilisée pour diviser une adresse IP en sous-réseaux et spécifier les hôtes disponibles du réseau. Dans un masque, deux bits sont toujours affectés automatiquement. Par exemple, dans 255.255.255.0, "0" est l'adresse de réseau assignée. Dans 255.255.255.255, "255" est l'adresse de diffusion attribuée. Le 0 et 255 sont toujours assignés et ne peuvent pas être utilisés.

### 3.2.1 Format du masque

Il est composée de 32 bits alors même taille qu'une adresse IPV4, dans le masque ont place les bits à 1 de manière contigus et les bits à 0 à droite.

*Exemples :*

11111111.00000000.00000000.00000000 = 255.0.0.0  
 11111111.11111111.11111111.00000000 = 255.255.255.0  
 11110000.00000000.00000000.00000000 = 240.0.0.0

*Exemple invalide :*

11111111.01111111.00000000.00000000

### 3.2.2 Calcul de l'adresse réseau et numéro de l'hôte

Pour le calcul d'une adresse réseau on effectue un ET logique bit à bit entre le masque de réseau et l'adresse IP. Alors on détermine :

Pour la partie réseau (netid) : on effectue l'opération suivante : net-id <== adresse IP ET (bit à bit) Masque Exemple : 192.168.52.0 <== 192.168.52.85 & 255.255.255.0

Pour la partie hôte (hostid) : on effectue l'opération suivante : host-id <== adresse IP ET (bit à bit) Masque Exemple : 0.0.0.85 <== 192.168.52.85 & 0.0.0.255

**Remarque :** De ce fait on distingue deux adresses particulières parmi tout ceux possible, qui ne doivent jamais être attribués à des machines :

les bits Host-ID sont à 0 : adresse attribuée qu'à un réseau. Exemple : 192.168.10.0 / 255.255.255.0 = 192.168.10.00000000

les bits Host-ID sont à 1 : c'est une adresse de diffusion (broadcast), Exemple : 172.27.255.255 / 255.255.0.0 = 172.27.11111111.11111111

Donc nous pouvons en déduire que parmi tout les adresses assignable, ces derniers sont des adresses interdites.

### 3.2.3 Notation CIDR

Dans un premier temps nous avons vu que pour connaître l'adresse d'un réseau il faut forcément passé par le masque, une seconde forme existe et est connue sous le nom de notation CIDR (classless inter-domain routing) RFC 1519 ou l'adressage sans classe, ce qui veut dire qu'ici on ne tient plus compte de l'adressage par classe ; Donc aucun masque n'est fixé par rapport à une classe. Elle s'écrit avec le numéro du réseau suivi d'un slash et le nombre de bits à 1 (en partant de la gauche) en binaire du masque sous-réseau. De nos jours cette notation est la plus utiliser car les différentes classes utiliser sont devenues obsolètes. Exemple : 186.52.0.0/16 **Remarque :**

cette notation CIDR ne permet pas la construction des masques réseau à trous, alors que c'était possible dans la construction de base de IPV4 mais rarement utilisés car fastidieux la gestion. IPV6 intègre dès sa conception l'écriture et l'agrégation maximale des routes introduites par CIDR.

### 3.2.4 Adresses non utilisées

il existe des adresses non utilisable comme adresse IP pour une machine :

- les adresses réseaux : qui correspond aux adresses qui ont tous les bits de leur partie hostid à zéro(0) ;
- les adresses de diffusion (broadcast) : qui correspond aux adresses qui ont tous les bits de leur partie hostid à un(1)
- 0.0.0.0 : utilise par différentes services (table de routage, DHCP) et possède souvent une signification particulières.
- 127.X.X.X : désigne l'ordinateur lui-même ou dite adresse de bouclage (lookback), 127.0.0.1 pour le localhost
- > à 223.255.255.255 : pour le multicast et la recherche.

## 3.3 Type d'adresse IP (Schéma dans pics sur comment c'est utilisé dans un ensemble de réseau )

On distingue deux (2) types d'adresse IP qui sont les adresses IP publiques & privées : Les adresses IP privées Les adresses IP privées sont représentés par toutes les adresses IP de classe A, B et C qui sont utilisable dans un réseau local (par exemple le LAN) alors ce qui correspond au réseau de votre entreprise ou celle de votre réseau domestique. D'autre part, les adresses IP privées ne sont pas utilisable sur internet (car elles ne peuvent pas être routées sur internet), les machines qui les utilisent ne peuvent être atteint qu'à partir de votre réseau local. Les classes A, B et C ont chacune une correspondance de plage d'adresses IP privées à l'intérieur de la plage globale qui a été définie par la RFC 1918. Mais l'utilisation de celui-ci pour inter-connecter des réseau géante (entreprise) avec des espaces adressage qui se chevauche peut causer des problèmes. Une adresse IP privées est librement paramétrée par l'administrateur du réseau local. Les adresses

privées de la classe A : 10.0.0.0 à 10.255.255.255

Les adresses privées de la classe B : 172.16.0.0 à 172.31.255.255



Les adresses privées de la classe C : 192.168.1.0 à 192.168.255.255

Alors on vient de voir que les adresses IP privées sont utilisable uniquement sur des réseaux locaux, tandis qu'il y a des adresses IP qui ne sont utilisées uniquement que sur internet donc nous pouvons en déduire que c'est les adresses IP publiques non utilisable dans un réseau local. Les routeurs (par exemple : votre box) ont une adresse IP publique du côté d'internet, ce qui permet de rendre votre box visible sur internet (elle répondra certainement au ping). De plus, au moment de vos connexion sur un site web vous utilisez l'adresse publique du serveur web. De ce fait une adresse IP publique est unique dans le monde, ce qui n'est pas le cas dans le systèmes d'adressage des adresses IP privées qui doivent être unique seulement dans un même réseau local mais pas au niveau planétaire étant donné que ces adresses ne peuvent pas être routées sur internet. Une adresse IP publique est soit achetée ou fournie par la FAI. Les IP publiques représentent toutes les adresses IP des classes A, B et C qui ne font pas partie de la plage d'adresses privées de ces classes ou des exceptions de la classe A (voir Adresse non utilisé ci-dessus).

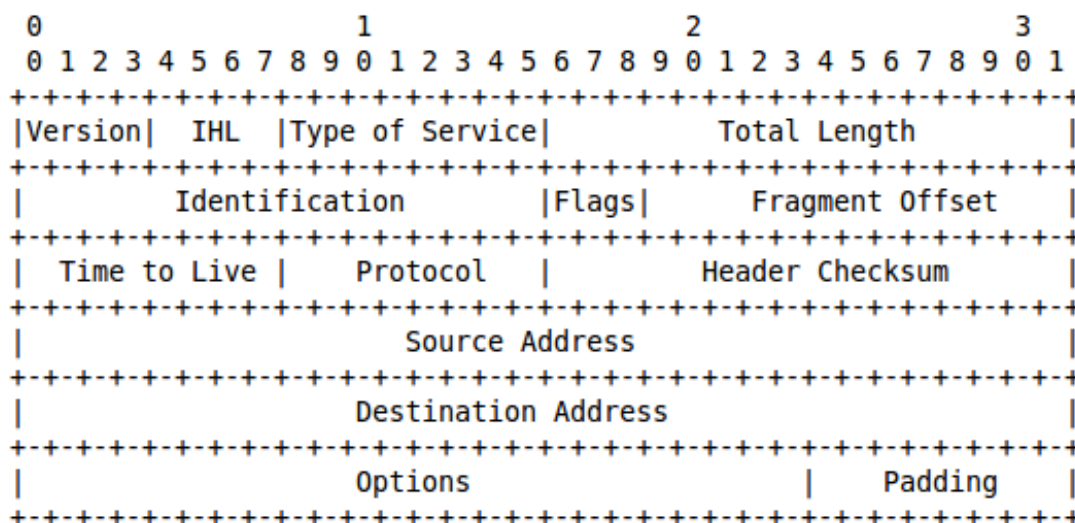
### 3.4 Les classes d'adresses

Au début de la création de IPV4, maintes groupes d'adresses ont été définis pour faciliter le routage (ou cheminement) des paquets. Structurée en 5 classes (A,B,C,D,E) selon la valeur du première octet.

De ce fait on remarque une distribution de l'espace d'adressage selon laquelle la classe A possède 50% l'espace et soit 25% pour la classe B, 12,5% classe C et 6,25% pour D & E. on peut en-déduire une mauvaise répartition de cette espace d'adressage.

### 3.5 En-tete 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.



Comme on peut voir dans la figure ci dessus, un en-tête IPv4 est composé par 13 champs, plus le padding. 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 le 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 (chouche 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 à 60 octets. Comme l'en-tête IPv4 a une taille minimale de 20 octets (160 bits), le champ IHL ne peut pas contenir une valeur inférieure à 5.

**Type of Service** Le champ Type of Service, mieux connu avec l'acronyme ToS, est utilisé pour spécifier la qualité de service souhaitée pour l'envoi d'un paquet IPv4. Cet champ occupe un octet de l'en-tête et il se compose en trois parties. Une première partie de 3 bits permet d'indiquer la priorité avec laquelle le paquet doit être traité, les 3 bits après sont utilisés pour spécifier certaines caractéristiques du service, notamment : le temps, le débit et la fiabilité. Enfin l'emploi des 2 derniers bits n'a pas été spécifié et leur usage a été laissé libre pour des implémentations futures.

En réalité l'histoire de ces champs est bien plus longue et complexe que ça, car en pratique la façon d'utiliser ces champs a été modifiée plusieurs fois au cours des années.<sup>5</sup> Cette manque de stabilité a par fois causé une certaine confusion dans les implémentations.<sup>6</sup>

Aujourd'hui les 8 bits du champ ToS sont utilisés par le mécanisme DiffServ (Differentiated Services). Cet système utilise les premières 6 bits du champ ToS (DSCP - Differentiated Services Code Point) pour marquer chaque paquet comme appartenant à un niveau de priorité et une classe de service. Chaque classe détermine le type de traitement que l'on souhaite demander pour le paquet aux routeurs au long du chemin (PHB - Per-Hop behaviour), toutefois le service offert par chaque routeur est fortement lié à sa configuration.<sup>7</sup> Les derniers 2 bits du champ ToS sont utilisés pour l'extension ECN (*Explicit Congestion Notification*). Cette extension, proposée par RFC2481 et introduite deux ans après avec le RFC3168, ajoute un système de contrôle de la congestion du trafic réseau. Dans le cas d'une saturation du réseau ce champ est utilisé pour notifier ce problème et demander au dispositif émetteur une réduction du rythme auquel les paquets sont envoyés, avec l'objectif de réduire l'attente et la perte de paquets.

**Total length** Comme le suggère le nom, ce champ est utilisé pour indiquer la taille totale du paquet IPv4 : en tête plus données. Le champ *Total length* est défini sur 16 bits, ceci

---

5. L'utilisation des 8 bits du champ ToS a été redéfinie par cinq standards différents (plus divers standards expérimentaux). Les documents présentant ces standards sont mentionnés dans le chapitre "Historical Definitions for the IPv4 TOS Octet" du RFC 3168

6. 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"

7. "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>

permet de indiquer une valeur comprise entre 0 et 65,535 octets. Comme l'en-tête est compris dans la longueur totale d'un paquet, cette valeur ne serait jamais inférieure à 20 (taille minimale d'un en-tête IPv4 en octets). RFC 791 impose à toutes les dispositifs d'un réseau IPv4 la capacité de recevoir des paquets jusqu'à une taille de 576 octets, cette prerogative permet de éviter une excessive fragmentation.

**Identification** Cet en-tête (sur 16 bits) permet d'identifier les fragments appartenant au même paquet.

**Flags** Les 3 bits du champ Flags sont utilisés pour gérer la fragmentation d'un paquet. Un de ces bits est employé pour indiquer si le paquet peut être fragmenté ou non. Ce bit, appelé DF (*Don't Fragment*), doit être pris en considération par les routeurs dans le chemin pour décider si un paquet trop grand pour être transmis peut être retransmis sous forme de fragments plus petits ou rejeté. Un autre bit, appelé MF (*More Fragments*), indique si le paquet est suivi par d'autres fragments. Le bit MF est mis à 0 dans le dernier fragment ou dans des paquets qui n'ont pas été fragmentés.

Un des trois bits de ce champ n'est pas actuellement utilisé mais il a été réservé pour applications futures possibles.<sup>8</sup>

**Fragment Offset** Lorsque un paquet a été fragmenté, cet en-tête est utilisé pour déterminer la position (offset) d'un fragment par rapport aux données du paquet réassemblé. Le décalage de chaque fragment est exprimé en blocs de huit octets (ou 64 bits). Le champ Fragment Offset utilise 13 bits de l'en-tête IPv4, cela permet un offset maximal de 65,528 octets.<sup>9</sup> Étant donné que le flag MF (*More Fragments*) doit être mis à zéro lorsque si un paquet n'est pas fragmenté ou si il est le dernier fragment d'un paquet plus grand, l'unique différence entre ces deux types de paquets est la valeur de ce champ Fragment Offset que, dans le cas d'un paquet pas fragmenté, est toujours zéro.

**Time to Live** Ce champ détermine le nombre maximal de fois qu'un paquet peut être retransmis, il est utilisé pour empêcher qu'un paquet puisse être retransmis à l'infini. Chaque routeur au long du chemin d'un paquet est tenu de détruire un paquet si la valeur du TTL (*Time to Live*) est zéro ou de decrementer ce champ par le nombre de secondes que le paquet passe en attente d'être transmis.

En théorie le TTL indique le nombre de secondes pendant lesquelles un paquet peut continuer à être retransmis dans un réseau, mais, étant chaque routeur toujours tenu de decrementer ce champ de au moins 1 (même si le paquet a été retransmis en moins d'une seconde) et considérant les performances des routeurs d'aujourd'hui, en pratique le TTL indique le nombre maximum de routeurs que un paquet peut rencontrer au long de son chemin.

Le espace réservé au TTL dans l'en-tête IPv4 est d'un octet ce qui comporte un TTL maximum de 255.<sup>10</sup>

Quand un paquet a été détruit suite à l'expiration du TTL, le routeur qui a détruit le paquet peut décider d'envoyer un message d'erreur à l'émetteur du paquet détruit. Ce type de message (ICMP Time exceeded) est utilisé par exemple comme *traceroute* pour découvrir, approximativement, le chemin d'un paquet IP.

**Protocol** Chaque paquet IPv4 spécifie le protocole utilisé par les données transmises : cela est l'objectif de ce champ de 8 bits

---

8. Ce bit a été aussi le protagoniste d'un des plus connus poissons d'avril présentée par le IETF. Pour faciliter les tâches des systèmes de filtrage le RFC 3514 propose d'utiliser ce bit pour étiqueter paquets malveillants, à ce titre tous les paquets étant envoyés avec ce bit (renommé "*Evil Bit*") mis à 1 seront mis à la poubelle.

9. En pratique un tel offset n'est jamais utilisé car, en ajoutant un en-tête minimal de 20 octets, la taille totale du paquet réassemblé dépasserait la longueur maximale d'un paquet IPv4.

10. Le RFC 1700 recommande une valeur par défaut de 64.

**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>11</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>12</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 envoye 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>13</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 des fonctions extra. Aujord'hui il y a quelque dizaine de options qui ont ete specifie<sup>14</sup> (si on considere aussi les options experimentales) mais peu entre eux sont reelement utilise. Entre les options les plus connues on retrouve par exemple des ajoutes utiles a l'administration et au debougage d'une reseau, comme *Record route* qui permet de enregistrer les adresses des routeurs dans le chemin d'un paquet IP, et *Timestamp* qui permet de savoir le temp passe entre chaque hop du chemin.

Entre les options il en y a deux qui ont une fonction speciale : EOL (*End Of Option List*) and NOP (*No Operation*) : l'option EOL est utilise pour indiquer la fin de la chaine d'options, NOP est une option sans aucun effet, elle est utilise comme replissage pour aligner les options quand elles ne sont pas aligne sur 4 octets<sup>15</sup>.

**Padding** Le padding ne contiens que des zeros et est utilise quand la fin de l'en-tete n'est pas aligne sur 32 bits (4 octets). Bien entendu, cet champ est optionel : en effet il pest

---

11. Cela est necessaire car le TTL est decrementee at chaque retransmission et un changement de l'en-tete comporte une valeur different dans la somme de controle

12. 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).

13. 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...).

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

15. On rappelle que le champ IHL indique la longueur de l'en-tete IPv4 en blocues de 32 bits (4 octets)

etre necessaire seulement lorsque l'en-tete IPv4 termine avec une option lequel limite n'est pas aligne sur 32 bits. Un en-tete n'a besoin d'aucun Padding standard IPv4 de 20 octets (donc sans aucune option) termine comme etant aligne sur 4 octets (20 est un multiple de 4).

## 4 Suite de protocole

### 4.1 ARP

ARP (Address resolution protocol) est un protocole à cheval entre la couche 2 et la couche 3 permettant de faire la conversion entre les adresses de niveau 2 et de niveau 3. Cela est très utilisé étant donné que les hôtes 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.

#### 4.1.1 Cache ARP

Le cache ARP ou table ARP est une table stockée en local par un hôte et qui regroupe les associations entre adresse IP et adresse MAC. Ces associations peuvent être soit de type statique, donc écrites "en dur" par l'administrateur, ou dynamique, donc issues d'échange de trame ARP, et qui possèdent, en plus des associations statiques, une durée de validité, étant donné qu'une interface peut changer d'adresse IP et qu'il vaut maintenir la table à jour. Cette table peut donc être représentée comme une suite d'entrées, contenant chacune une adresse IP, une adresse MAC et éventuellement une durée de vie.

#### 4.1.2 Entête

//TODO

#### 4.1.3 Fonctionnement

Prenons l'exemple où A veut envoyer un message à B. A connaît 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. Le paquet passe dans la couche liaison, il va être encapsulé dans une trame de niveau 2. Cette trame aura comme adresse source l'adresse de niveau 2 de A, mais à ce moment il 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.

Pour faire cela l'hôte A va tout d'abord regarder dans sa table ARP si il n'a pas une entrée pour l'adresse IP à laquelle il souhaite envoyer son paquet. Si il trouve une correspondance, il va utiliser l'adresse MAC stockée en correspondance avec l'adresse IP recherchée. Si il ne trouve pas de correspondance il va émettre une trame ARPREQUEST pour tenter de contacter le possesseur de l'adresse IP qu'il souhaite contacter. Il va mettre dans cette trame (en plus de l'entête de niveau 2) un entête ARP. Celui-ci contient plusieurs informations : //TODO quels info mettre ? type de trame ARP : request ou reply Les informations importantes pour permettre la résolution d'adresse sont les adresses IP source et destination, ainsi que les adresses MAC source et destination. L'hôte A va donc mettre son adresse IP en IP source (entête ARP) et son adresse MAC dans l'adresse physique source (entête ARP). Dans le champ d'adresse IP destination (entête ARP) l'hôte A va mettre l'adresse IP de l'hôte qu'il veut contacter. Étant qu'il cherche à avoir l'adresse physique de l'hôte B, il ne peut pas indiquer d'adresse MAC dans le champ d'adresse physique destination. Ce champ est donc rempli avec la valeur 0. Sachant que l'hôte A n'a pas l'adresse physique du destinataire, il va envoyer sa trame en broadcast

pour espérer atteindre l'hôte B sans connaître son adresse MAC. Une fois que l'hôte B reçoit la requête ARP, il va analyser son entête et remplir sa table ARP avec l'adresse IP de l'hôte A et l'adresse physique de A. Cela permet de créer une correspondance entre l'adresse IP et MAC de A, pour non seulement pouvoir répondre à sa requête ARP, et pour pouvoir contacter A dans le futur sans avoir besoin de refaire la demande ARP. Ensuite l'hôte B va répondre avec une trame ARPREQUEST. L'entête est similaire aux ARPREQUEST, seul le champ indiquant s'il s'agit d'une trame ARPREQUEST ou ARPREQUEST change. Dans cette trame, l'hôte B va placer son adresse IP dans le champ adresse IP source et son adresse physique dans le champ d'adresse physique source. Il va aussi mettre l'adresse IP de A dans le champ adresse IP destination et l'adresse MAC de A dans le champ d'adresse physique destination. Il va ensuite envoyer cette trame en unicast à A. Lorsque A reçoit la trame ARPREQUEST, il va à son tour mettre dans sa table ARP, la correspondance entre l'adresse IP source et l'adresse MAC source de la trame (soit les adresses de B). Une fois cette association mise en place, le paquet IP que voulait envoyer A au début et mis en pause le temps que le protocole ARP fasse l'association, est enfin envoyé étant donné que A à maintenant l'adresse physique de B.

//TODO ARP REQUEST en broadcast

#### 4.1.4 ACD

Le protocole ACD (Adresse conflict detection) permet, comme son nom l'indique, de détecter les conflits d'adresse, qui sont l'utilisation de la même adresse IP par deux ou plusieurs hôtes en même temps. Pour ce faire il utilise le protocole ARP avec une succession d'étape permettant de garantir l'utilisation de manière unique d'une adresse IP.

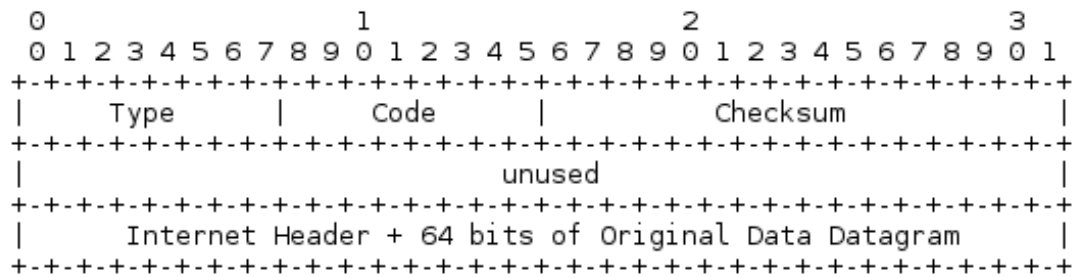
Pour commencer, ACD intervient au moment où une interface reçoit une adresse IP (soit par DHCP, soit par une configuration manuelle,...). Il faut à ce moment vérifier si l'adresse proposée n'est pas déjà utilisée par un autre hôte sur le réseau. L'hôte va alors émettre une requête ARP en broadcast en remplissant l'entête ARP avec son adresse MAC dans le champ d'adresse physique source et 0.0.0.0 dans l'adresse IP source (car il n'a pas encore d'adresse IP attribuée, et pour éviter de corrompre les tables ARP des autres hôtes). Le champ adresse IP destinataire est complété avec l'adresse que l'on souhaite acquérir. On ne peut pas remplir le champ d'adresse physique destinataire étant donné qu'on ne sait pas si il y a des hôtes avec cette adresse déjà configurée. Une requête ARP contenant 0.0.0.0 comme adresse IP source est appelée ARP probe car elle sert à "sonder" si un autre hôte utilise déjà l'adresse que l'on passe dans l'adresse IP destinataire.

Après avoir attendu un temps pouvant aller jusqu'à 1 seconde, l'hôte va envoyer un nombre d'ARP probe compris entre 1 et 3, et tous espacés d'un intervalle compris entre 1 et 2 secondes. Si dans un délai de 2 secondes après l'émission de l'ARP probe l'hôte reçoit un paquet ARP request ou reply avec comme adresse IP source l'adresse qu'il souhaite acquérir, alors cela veut dire qu'un autre hôte est entrain d'utiliser cette adresse. L'ARP reply peut être la réponse à l'ARP probe émis et l'ARP request peut simplement être une demande ARP faite par l'hôte qui utilise déjà l'adresse que l'on souhaite acquérir. En plus de surveiller ces deux types de messages, l'hôte doit vérifier les messages ARP probe qu'il reçoit. En effet, il se peut que deux hôtes décident de configurer leur interfaces avec la même adresse au même moment. Sachant qu'aucune de ces deux interfaces n'a encore d'adresse IP attribuée, aucune ne va répondre à l'ARP probe du deuxième hôte. Cela va conduire à l'attribution de la même adresse pour plusieurs interfaces. Pour éviter ce problème l'hôte doit surveiller les ARP probe qui passent à son interface. Si il en reçoit un avec comme adresse destination la même adresse qu'il souhaite acquérir mais avec une adresse physique différente de la sienne, cela veut qu'un autre hôte souhaite utiliser la même adresse que lui. Dans ce cas l'adresse ne peut pas être utilisée de manière sûre.

Si après les 2 secondes d'attente du dernier ARP probe, l'hôte n'a pas reçu d'ARP probe, ou d'ARP request/reply indiquant un conflit d'adresse, alors l'hôte peut considérer que l'adresse qu'il souhaite utiliser est unique dans le réseau, et qu'il peut l'utiliser de manière sûre.

La dernière étape est d'annoncer que l'hôte utilise l'adresse qu'il vient d'acquérir. Pour ce





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

#### 4.2.1 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

#### 4.2.2 Message de type 4 :

### 4.2.3 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.

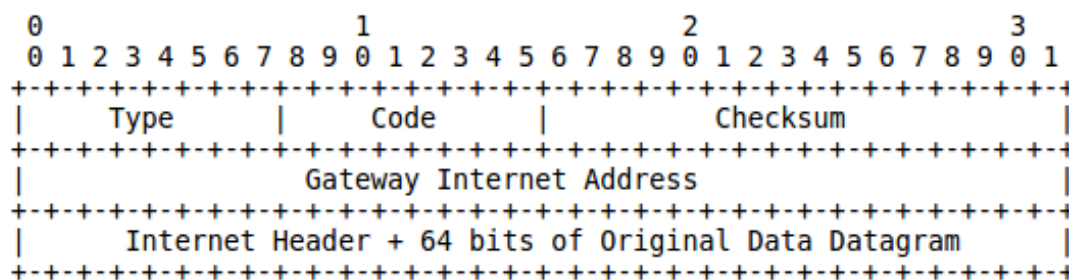
**Code 0 :** Le code 0 est utilisé pour indiquer que le TTL du paquet posant problème est arrivé à 0. Lorsque le TTL d'un paquet arrive à 0, celui-ci est supprimé et un message ICMP de type 11 et de code 0 est envoyé par le routeur qui a détecté le problème. Cela permet principalement d'éviter qu'un 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

#### 4.2.4 Message de type 5 :

Les messages 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 connectés au réseau). Concernant le paquet que le premier routeur a reçu, il va le transmettre vers sa destination.





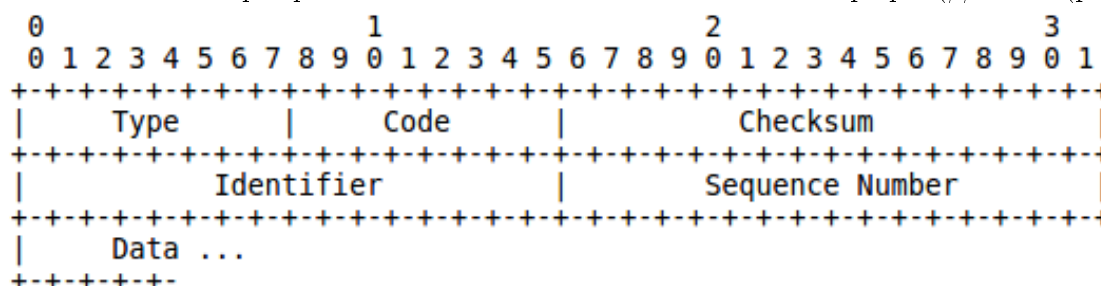
Le champ Gateway Internet Address contient la l'adresse du routeur auquel il faut faire transiter le trafic directement pour avoir un chemin de routage plus court. Le champ Internet Header contient toujours l'entête du message ayant porvoqué l'envoi du message ICMP plus les 64 bits suivant l'entête. 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)).



#### 4.2.5 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 Identififier 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 ?)

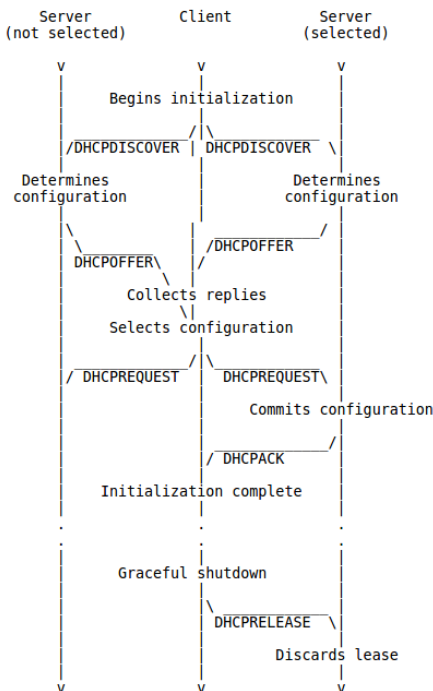
### 4.3 IGMP

### 4.4 DHCP

<sup>17</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

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

faire parvenir d'autres informations essentielles pour le fonctionnement de l'interface sur le réseau. Voyons comment une interface peut se configurer auprès d'un serveur DHCP.



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. L'hôte va utiliser comme adresse IP 0.0.0.0.

Étant donné que le message est envoyé en broadcast, tous 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 autres l'adresse IP proposée pour le client souhaitant se configurer, ainsi que le masque de sous-réseau de l'adresse. À ce moment-là, l'adresse n'est pas encore attribuée et réservée 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 DHCPDISCOVER. Si il reçoit un ou plusieurs DHCPOFFER, l'hôte va devoir choisir une configuration qui lui est proposée. Une fois ce choix fait, il va informer 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 souhaitée 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 attribuée à 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 raison le serveur n'est pas capable d'attribuer l'adresse proposée dans l'offre (par exemple si l'adresse a été attribuée temporairement), 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 attribuée à l'hôte, elle reste disponible pour être attribuée à d'autres hôtes. 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

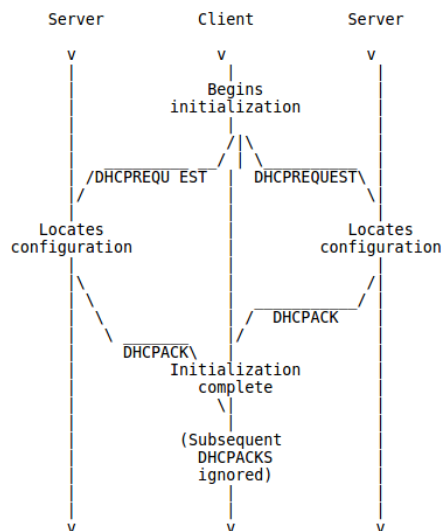
durée de bail). Il va effectuer une dernière vérification pour s'assurer que l'adresse qui lui à été attribué est bien unique sur le réseau pour éviter d'avoir deux hôte avec la même adresse. Il va pour cela utilisé le protocole ARP et la méthode de vérification vu plus haut. Si jamais l'adresse est 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 terminé 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 réémettre 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 à reçu le message DHCPACK du serveur, celui-ci lui a transsmis 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 à atteint le temps T1, il va chercher à contacter le serveur qui lui à 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.

Cependant,dans ce cas comme dans d'autre, l'hôte peut ré-utiliser une configuration précédemment utilisée. Cela permet de raccourcir la négociation entre l'hôte et le serveur DHCP. L'hôte va directement commencé la négociation en faisant un DHCPREQUEST en broadcast et contenant la configuration qu'il souhaite ré-utiliser. Le serveur concerné par l'attribution antérieur de la configuration va donc accepter la demande de l'hôte à l'aide d'un DHCPACK ou la refuser, si la demande n'est pas correct ou si l'adresse est utilisé par un autre hôte, à l'aide d'un DHCPNAK. Cette négociation se fait de manière similaire qu'un négociation complète, elle a juste été raccourci en enlevant quelque étape non indispensable.



## 5 Passage de IPv4 à IPv6

### 5.1 Raison du passage de IPv4 à IPv6

#### 5.1.1 Problème posé par IPv4

//Pénurie, adressage privé/public compliqué

#### PROBLÈMES DE IPV4

**ÉPUISEMENT DES ADRESSES** Lorsque IPV4 a été développé dans les années 70-début des années 80, personnes n'aurait imaginé qu'il y aurait un jour autant d'interfaces qui se connectent à Internet. On pensait qu'une adresse sur 32 bits serait suffisante. De plus, les plages d'adresses étaient distribuées généreusement au début. Cela veut dire que l'on attribuait des adresses permettant un nombre d'interfaces beaucoup plus grand que nécessaire. Cependant, avec la croissance du nombre d'utilisateurs, la plage d'adresses IPV4 disponible a diminué progressivement. C'est en février 2011 que la réserve de bloc libres d'adresses publics IPV4 de l'IANA (Internet Assigned Numbers Authority) est arrivée à épuisement. Afin de résoudre ce problème, plusieurs techniques ont été proposées. La première a été le changement de technique d'adressage. On est passé de la technique de classe d'adressage IP à la technique Classless Inter-Domain Routing. Ceci a permis une meilleure efficacité dans la distribution des adresses IP grâce à la création de réseau de tailles intermédiaire. En effet, avant on ne disposait que de réseau de 3 tailles différentes. Les politiques d'assignement d'adresses ont également été rendu plus stricte afin de mieux tenir compte des besoin réels des demandeurs d'adresses IP. Il a aussi été décidé d'utiliser des blocs autrefois réservé comme 14.0.0.0. Sur base de volontarisme, des blocs autrefois attribués généreusement ou alors des IP non utilisées ont été récupérées. Finalement, il a été remarqué qu'il n'était pas nécessaire que chaque interface a son adresse IP public et le protocole NAT a été développé afin de regrouper plusieurs interface sous une même adresse IP. Ce protocole est de plus en plus utilisé dans IPV4 depuis la fin des années 90.

**Fonctionnement du NAT dynamique (Network Address Translation)** Le NAT est une technique utilisée au niveau du routeur. Le principe du NAT est que le routeur fait correspondre à une adresse IP une autre adresse IP. En général cette technique est utilisée pour avoir une même adresse IP pour tout un réseau comme un intranet ou encore un réseau domestique. Dans ce réseau, toutes les interfaces - même le routeur - auront une adresse privée. Le routeur dispose en plus de cela de une ou plusieurs adresses publics avec lesquelles il est connecté à internet. Une adresse privée est une adresse qui est utilisée à l'intérieur d'un réseau local. Les adresses privées peuvent être choisies parmi les suivantes : 10.0.0.0/8, 172.16.0.0/12 ou 192.168.0.0/16. Lorsqu'une interface envoie un paquet vers l'extérieur du réseau, le routeur effectue plusieurs

changements. Il traduit d'abord l'adresse privée en adresse public et la met dans l'en-tête du paquet. Puis il change tous les checksums qui tiennent compte de l'adresse IP. Enfin, il garde en mémoire dans une table la correspondance entre adresse privée/adresse public comme ci-dessous. <tableau adresse public / privée > Cela n'est cependant pas suffisant. En effet, lorsqu'un paquet arrivera de l'extérieur du réseau, et si tous les interfaces utilisent la même adresse public sans distinction supplémentaire, le routeur ne saura pas à quelle interface envoyer le paquet. Une solution à ce problème existe pour les protocoles utilisant les ports comme TCP et UDP. Le routeur ajoute une information supplémentaire dans la table qui est le port source d'où vient le paquet. Les ports, qui sont implémentés dans la couche transport (couche 4), sont des sortes de "portes" qui permettent de communiquer avec un système d'exploitation. Le numéro de port est un numéro choisit aléatoirement entre 1024 et 65535. Pour illustrer le fonctionnement du NAT imaginons qu'une interface A dont l'ip est 192.168.0.1 veut envoyer un paquet à l'interface B d'ip 217.70.184.38. Le port source est le port 10277 et le port destination est le port 80. La table NAT ressemblera à ceci :

<TABLE NAT complète >

La box internet enverra le paquet :

L'interface B répondra en envoyant le paquet :

Lorsque la box reçoit ce paquet, elle voit que le port de destination est le port 10277. Elle cherche ensuite le port correspondant dans sa table NAT. Lorsqu'elle le trouve elle effectue les changements nécessaire sur le paquet et transmet le paquet à l'interface A. Mais même si cette solution fonctionne la plupart du temps, la probabilité est faible que 2 interfaces envoient des paquet sur les même port. C'est pour éviter cela que la box change le port source lorsqu'elle reçoit un paquet de l'interface A. Ainsi on s'assure que aucun port n'est utilisé plusieurs fois. Enfin, pour éviter de saturer les ports utilisés, un compteur est associé à chaque paire adresse public/adresse privée. Lorsqu'il n'y a pas de trafic entre une adresse privée et l'extérieur durant une durée fixée, le port qui lui est associée peut être réutilisé pour une autre adresse privée.

Le NAT dynamique apporte cependant un grand problème. Lorsqu'une interface extérieur veut se connecter à une interface dans le réseau, elle ne dispose d'aucune autre information que l'adresse IP public. Si elle envoie alors un paquet à cette adresse, le routeur qui le réceptionnera ne saura pas quoi faire avec et le paquet sera perdu. On a réussi à pallier à ce problème grâce au port forwarding.

PORT FORWARDING

NAT STATIQUE

### 5.1.2 Solutions

//NAT,IPv6

## 5.2 Différence entre IPv4 et IPv6

## 6 Conclusion



## Références

- [1] Réseaux informatiques et systèmes embarqués. <https://mathinfo.unistra.fr/offre-de-formation/master-mention-informatique/master-rise/>, section Master 2 RISE.