

Transmission de l'information & protocoles Internet

Philippe Latu

philippe.latu(at)linux-france.org

<http://www.linux-france.org/prj/inetdoc/>

Historique des versions		
\$Revision: 1619 \$	\$Date: 2011-04-05 00:08:00 +0200 (mar. 05 avril 2011) \$	PL
Année universitaire 2004-2005		
Résumé		
Lors de la conception de la modélisation OSI, tout a été prévu pour utiliser plusieurs protocoles différents aux niveaux réseau et transport. Aujourd'hui, la généralisation de la technologie Ethernet dans les réseaux locaux et l'utilisation systématique de l'Internet ont balayé cette ouverture aux protocoles multiples. Cet article illustre le fonctionnement de la transmission de l'information sur l'Internet à partir de ses protocoles emblématiques IP, ICMP, TCP et UDP.		

Table des matières

1. Copyright et Licence	1
1.1. Méta-information	2
2. Les couches réseau & transport de l'Internet	2
2.1. Les protocoles étudiés	2
2.2. Les unités de données	2
3. Le protocole IP de la couche réseau	3
4. Les protocoles TCP & UDP de la couche transport	5
4.1. Le protocole TCP	5
4.2. Le fonctionnement du protocole TCP	7
4.2.1. La segmentation des données	7
4.2.2. L'établissement de la connexion	7
4.2.3. Contrôle de flux	7
4.2.4. fenêtrage	7
4.3. Le protocole UDP	8
5. En guise de conclusion	8

1. Copyright et Licence

Copyright (c) 2000,2011 Philippe Latu.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2000,2011 Philippe Latu.
Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License), version 1.2 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est incluse dans la section intitulée « Licence de Documentation Libre GNU ».

1.1. Méta-information

Cet article est écrit avec *DocBook*¹ XML sur un système *Debian GNU/Linux*². Il est disponible en version imprimable aux formats PDF et Postscript : [transmission.tcpip.pdf](http://www.linux-france.org/prj/inetdoc/telechargement/transmission.tcpip.pdf)³ | [transmission.tcpip.ps.gz](http://www.linux-france.org/prj/inetdoc/telechargement/transmission.tcpip.ps.gz)⁴.

2. Les couches réseau & transport de l'Internet

Avec le succès des services Internet (noms de domaines, courrier électronique, Web, etc.) et l'échec des couches hautes du modèle OSI (voir *Modélisations réseau*), les couches réseau & transport du modèle TCP/IP occupent une position *charnière* entre le traitement de l'information (les couches hautes ou services) et sa transmission (les couches basses).

Baptiser une modélisation du nom de ses protocoles n'est pas très académique. Le principe d'un modèle est de décrire «ce qui doit être fait» et non «comment ce doit être fait». Or, par définition, un protocole décrit comment une série d'opération doit être réalisée.

Cette transgression est justifiée par le fait que toute la «philosophie» du fonctionnement de l'Internet est résumée dans les deux protocoles TCP & IP.

2.1. Les protocoles étudiés

Le protocole de couche réseau IP

Un réseau IP (*Internet Protocol*) est un réseau à commutation de paquet sans mécanisme de contrôle d'erreur. Chaque paquet acheminé individuellement. C'est au destinataire des paquets de reconstituer, si besoin, la séquence des informations. Voir *Section 3, « Le protocole IP de la couche réseau »*.

Le protocole de couche réseau ICMP

Comme le protocole IP ne prévoit aucun contrôle, on a introduit un protocole supplémentaire au niveau réseau dont le rôle est uniquement de fournir des informations sur l'état du réseau : ICMP (*Internet Control Message Protocol*).

Le protocole de couche transport TCP

La fonction principale du protocole TCP est de fiabiliser les communications de bout en bout entre deux hôtes distants. Pour cela, il met oeuvre un jeu de mécanismes complexes : poignée de main à trois voies, numéros de séquence indépendants, fenêtrage de l'acquittement, etc. Voir *Section 4.1, « Le protocole TCP »*.

Le protocole de couche transport UDP

Le protocole UDP est un protocole de transport simplifié à l'extrême. Il suppose que le réseau est totalement fiable et ne prévoit aucun mécanisme de contrôle et correction d'erreur. C'est aux applications de gérer les problèmes de transmission le plus souvent à l'aide de boucles d'attente. Voir *Section 4.3, « Le protocole UDP »*.

2.2. Les unités de données

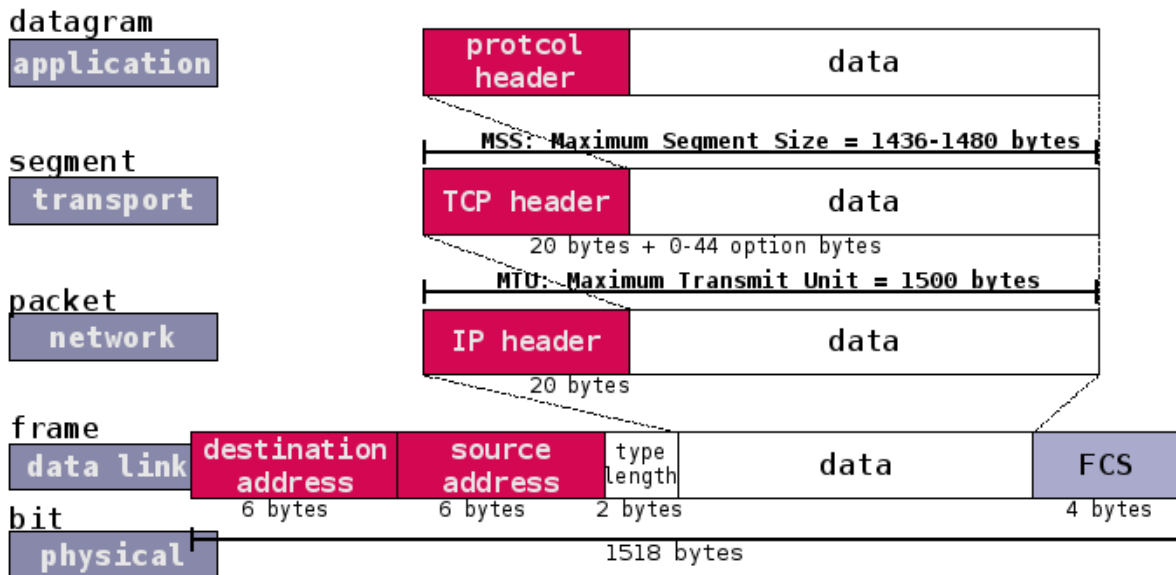
Les protocoles listés ci-avant échangent des données entre eux lors du passage d'une couche à l'autre. On parle d'unité de donnée de protocole ou *Protocol Data Unit* (PDU). Cette notion de PDU est générale et n'est pas très employée en dehors des présentations sur les modélisations. Avec l'utilisation systématique des protocoles de l'Internet, on a introduit un vocabulaire spécifique à chaque couche. Voici un schéma sur lequel figure ce vocabulaire ainsi que les dimensions en octets de chaque élément.

¹ <http://www.docbook.org>

² <http://www.debian.org>

³ <http://www.linux-france.org/prj/inetdoc/telechargement/transmission.tcpip.pdf>

⁴ <http://www.linux-france.org/prj/inetdoc/telechargement/transmission.tcpip.ps.gz>

Encapsulation - image seule⁵

Ce schéma fait apparaître le format de trame Ethernet. Même si la technologie Ethernet n'est pas directement liée aux protocoles de l'Internet, son format de trame tend à devenir universel. On le retrouve avec les technologies Wifi, les connexions ADSL/PPPOE et même de plus en plus sur les réseaux étendus.

Avant 1997, date à laquelle l'IEEE a incorporé le «vieux» format de trame Ethernet dans le standard officiel, on devait systématiquement distinguer deux formats de trames suivant le champ type/longueur.

- La définition de trame Ethernet II, celle qui utilise le champ type, a été intégrée avec les protocoles de l'Internet à partir des documents [RFC894 : A Standard for the Transmission of IP Datagrams over Ethernet Networks](#) et [RFC1042 : A Standard for the Transmission of IP Datagrams over IEEE 802 Networks](#). Le champ type de la trame indique le type du protocole de couche supérieure ; IP ou ARP dans la plupart des cas. Pour plus d'informations, voir la référence sur les [Valeurs du champ type de la trame Ethernet II](#).

Aujourd'hui, on devrait parler de *trame IEEE 802.3 avec encapsulation type*.

- La définition de trame IEEE 802.3 initiale, celle qui utilise le champ longueur, n'est jamais utilisée pour le trafic IP ; donc pour le trafic utilisateur. Seules les communications spécifiques entre équipements réseau utilisent ce format de trame associé à des protocoles spécifiques.

Aujourd'hui, on devrait parler de *trame IEEE 802.3 avec encapsulation longueur*.

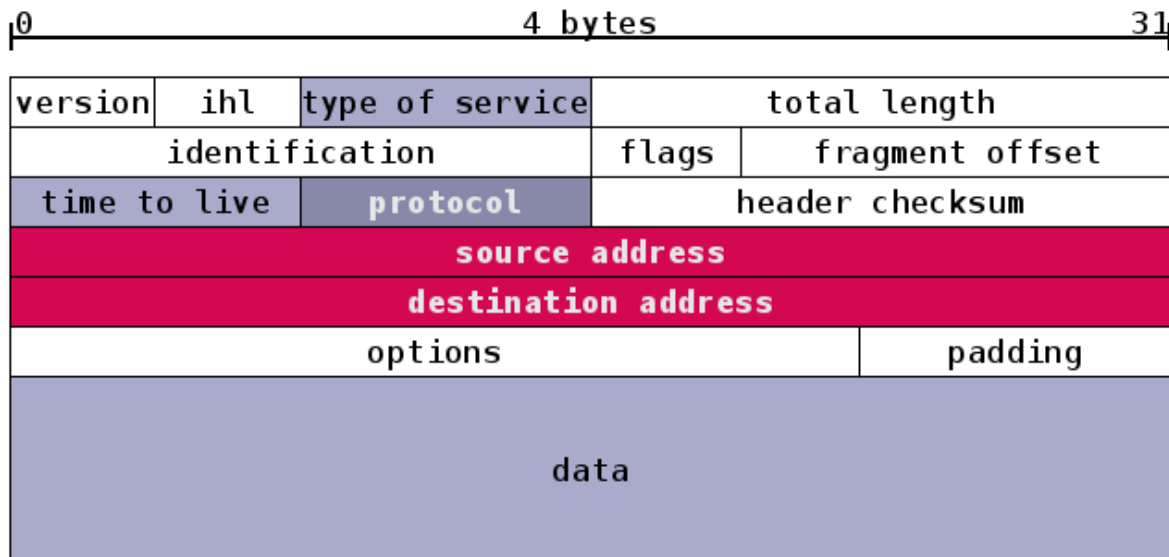
Pour plus d'informations, voir les références [Technologie Ethernet](#).

3. Le protocole IP de la couche réseau

En reprenant le principe à l'origine du réseau de communications militaires ARPANET, on doit pouvoir communiquer d'un point à un autre de l'Internet «quel que soit l'état du réseau». Une grande partie de l'infrastructure peut être détruite par une frappe nucléaire et les communications doivent toujours être possibles. C'est ce «mode de fonctionnement» qui a conduit à l'adoption d'un réseau à commutation de paquets fonctionnant en mode non connecté sans aucune hiérarchie. Si un autre principe avec supervision, hiérarchie et/ou mode connecté avait été retenu il suffirait qu'un point névralgique soit touché pour interrompre l'ensemble des communications.

Le fonctionnement du protocole IP qui a été décrit dans le document [RFC791 : INTERNET PROTOCOL](#), reprend l'ensemble des caractéristiques énoncées ci-avant. L'absence totale de mécanisme de contrôle et/ou de correction d'erreur est une caractéristique importante qui découle des mêmes principes. La fiabilisation des communications ne se joue pas au niveau réseau mais au niveau transport.

⁵ <http://www.linux-france.org/prj/inetdoc/articles/transmission.tcpip/images/encapsulation.png>

En-tête de paquet IP - image seule⁶

Version : 4 bits

Version du protocole IP codée sur 4 bits : 0100 pour IPv4 et 0110 pour IPv6.

Internet Header Length : 4 bits, IHL

Longueur de l'en-tête en mots de 32 bits. Cette valeur est utilisée pour distinguer la partie en-tête de la partie données du paquet. La représentation usuelle de l'en-tête se fait sur 32 bits de largeur. Comme les champs `Options` et `Padding` ne sont pas obligatoires, la valeur minimum du champ `IHL` est 5 (0101).

Type Of Service : 8 bits, TOS

Champ découpé en 2 parties. Les 3 premiers bits sont appelés *precedence* et les 5 derniers représentent le type de service. La définition d'origine prévoyait 3 choix : *low-delay*, *high-reliability* et *high-throughput*. Ce «marquage» des paquets est utilisable pour définir des flux prioritaires sur une interconnexion réseau «sous contrôle». Sur l'Internet, les opérateurs définissent leurs propres priorités ; donc leurs propres valeurs pour ce champ. Voir l'[Extrait du Linux Advanced Routing & Traffic control Howto](#).

Total Length : 16 bits

Longueur du datagramme : en-tête & données. La taille minimum est de 21 octets (en-tête + 1 octet de donnée). Comme ce champ est représenté sur 16 bits, la taille maximum est de $2^{16} - 1$, soit 64 Ko.

Identification : 16 bits

Chaque paquet IP reçoit un numéro d'identification à sa création. Il est possible qu'un paquet soit découpé en *fragments* avant d'atteindre sa destination finale. Chaque fragment appartient au même paquet IP. Chaque fragment possède le même numéro d'identification.

Flags : 3 bits

Ce champ contient 3 indicateurs d'état :

- *Reserved flag* : doit toujours être à 0.
- *Don't Fragment (DF)* : à 0 si le paquet peut être fragmenté ; à 1 s'il ne doit pas être fragmenté.
- *More Fragments (MF)* : à 1 si d'autres fragments sont attendus ; à 0 s'il n'y a pas/plus de fragments.

Fragment Offset : 13 bits

Position du fragment dans le datagramme courant. Cette position est comptée en octets.

Time To Live : 8 bits, TTL

Ce compteur est décrémenté à chaque traversée de routeur. Si la valeur 0 est atteinte, le paquet est jeté. Cela signifie qu'il ne peut être délivré à sa destination finale. La valeur initiale du champ `TTL` dépend du système d'exploitation utilisé.

⁶ <http://www.linux-france.org/prj/inetdoc/articles/transmission.tcpip/images/ip.header.png>

Protocol : 8 bits

Ce champ spécifie le protocole utilisé dans les données du paquet IP. Par exemple, la valeur 1 indique que le protocole utilisé est ICMP. On sait ainsi que ce paquet n'est pas destiné à une application. Les différentes valeurs de ce champs sont listées dans le fichier `/etc/protocols` sur les systèmes GNU/Linux ou *BSD.

Header Checksum : 16 bits

A chaque création ou modification d'un paquet, une somme de contrôle (*cyclic redundancy check*) est calculée sur son en-tête. Lorsque le paquet arrive à destination, cette somme est recalculée. Si le résultat diffère, c'est que le paquet a été endommagé lors de son trajet.

Source Address : 32 bits

Adresse IP de l'hôte qui a émis le paquet. Voir [Adressage IP](#).

Destination Address : 32 bits

Adresse IP de l'hôte qui doit recevoir le paquet. Voir [Adressage IP](#).

Options and Padding

Cette partie de l'en-tête est optionnelle. Ce champ est utilisé pour fournir des instructions spécifiques de distribution du paquet qui ne sont pas couvertes par les autres champs de l'en-tête. La taille maximum de ces instructions est limitée à 40 octets regroupés en double-mots de 32 bits. Les bits de *padding* servent à compléter le dernier double-mot de 32 bits.

Data

C'est le dernier champ du paquet IP. Il contient les «données» du paquet. Celles ci peuvent débiter par un en-tête de couche transport (4) qui donnera d'autres instructions à l'application qui recevra les données. Le champ *Data* peut aussi contenir un message ICMP qui ne contient aucune donnée utilisateur.

4. Les protocoles TCP & UDP de la couche transport

Relativement au protocole IP de la couche réseau, le protocole TCP de la couche transport ont un rôle de *fiabilisation* des communications entre 2 extrémités. Avec la couche transport, on aborde le domaine des communications *de bout en bout* indépendantes de l'état du sous-réseau. Les paquets peuvent être arrivés à destination par des chemins différents et dans le désordre.

Même si l'appellation courante de la modélisation n'a retenu que TCP comme protocole de transport, il existe deux protocoles distincts suivant le type de réseau utilisé. Le protocole TCP fonctionne en mode connecté et assure un service fiable tandis que le protocole UDP assure un service de datagramme en mode non connecté sans aucune garantie de fiabilité.

4.1. Le protocole TCP

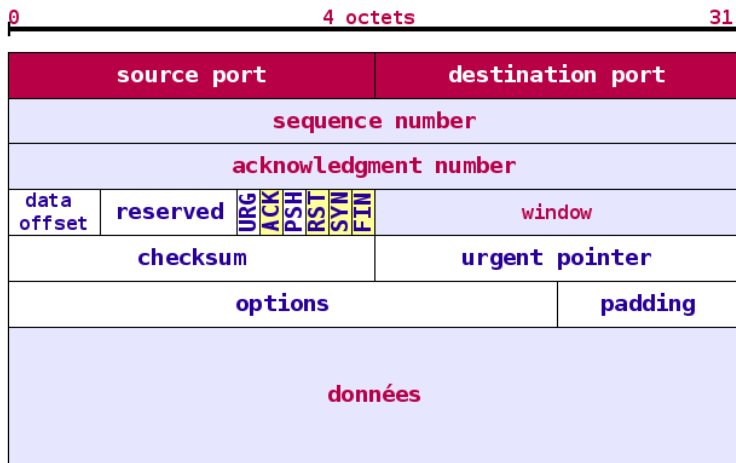
Historiquement, c'est le premier protocole de transport développé pour l'Internet. Les premières spécifications ARPANET prévoyaient un transport de l'information très fiable indépendant du type et de l'état du réseau. Le fonctionnement du protocole TCP a été décrit dans le document [RFC793 : Transmission Control Protocol](#).

- **Protocole de bout en bout.**

Les processus pairs des couches transport de deux équipements connectés dialoguent l'un avec l'autre sans rien connaître du réseau. C'est au **niveau IP** que l'on se préoccupe de la fragmentation et du réassemblage des paquets.

- **Protocole orienté connexion.**

La fiabilité du transport TCP dépend de l'établissement d'une connexion entre les processus pairs qui veulent dialoguer. L'établissement d'une connexion est réalisé par l'échange d'informations telles que le numéro de port, le numéro de séquence et la taille de fenêtre.



En-tête TCP - image seule⁷

Source Port : 16 bits

Numéro du port source. Ce numéro correspond au point de communication (*socket inet*) utilisé par le service de la couche application de l'émetteur.

Destination Port : 16 bits

Numéro du port destination. Ce numéro correspond au point de communication (*socket inet*) utilisé par le service de la couche application du destinataire.

Sequence Number : 32 bits

Le protocole TCP a besoin de garder une trace de toutes les données qu'il reçoit de la couche application de façon à être sûr qu'elles ont bien été reçues par le destinataire. De plus, le protocole doit être sûr que ces données ont été reçues dans l'ordre dans lequel elles ont été envoyées. Il doit retransmettre toute donnée perdue.

On affecte un numéro de séquence à chaque octet de donnée pour en garder une trace lors du processus de transmission, réception et acquittement. Dans la pratique, ce sont des blocs d'octets qui sont gérés en utilisant les numéros de séquence de début et de fin de bloc.

Les numéros de séquence sont nécessaires à la mise en oeuvre du système de fenêtre glissante du protocole TCP. C'est ce système qui garantit fiabilité et contrôle de flots de données.

Acknowledgment Number : 32 bits

Le rôle des numéros d'acquiescement est le même que celui des numéros de séquence. Simplement, chaque extrémité en communication initie son propre jeu de numéros. Ainsi chaque extrémité assure la fiabilisation et le contrôle de flux de façon autonome.

Data Offset : 4 bits

Nombre de mots de 32 bits contenus dans l'en-tête TCP. Indication du début des données. Tout en-tête TCP, avec ou sans options, est un multiple de mots de 32 bits.

Reserved : 6 bits

Champ réservé pour une utilisation ultérieure. Les 6 bits doivent être à 0.

Control bits : 6 bits

Ces bits sont les indicateurs d'état qui servent à l'établissement, au maintien et à la libération des connexions TCP. Leur rôle est essentiel dans le fonctionnement du protocole.

- URG : indique que le champ *Urgent Pointer* est significatif ; ie. une partie des données du segment sont urgentes.
- ACK : indique que le champ *Acknowledgment field* est significatif ; ie. le segment acquitte la transmission d'un bloc de d'octets.
- PSH : utilisation de la fonction Push
- RST : indique un arrêt ou un refus de connexion.

⁷ <http://www.linux-france.org/prj/inetdoc/articles/transmission.tcpip/images/tcp.header.png>

- SYN : indique une demande de synchronisation de numéro de séquence ; ie. demande d'ouverture de connexion TCP.
- FIN : indique que l'émetteur n'a plus de données à transmettre ; ie. demande de libération de connexion.

Window : 16 bits

Nombre d'octets de données à partir de celui indiqué par le champ Acknowledgment.

Checksum : 16 bits

Somme de contrôle sur 16 bits de l'en-tête et des données.

Urgent Pointer : 16 bits

Ce champ est interprété uniquement si le bit de contrôle URG est à 1. Le pointeur donne le numéro de séquence de l'octet qui suit les données urgentes.

Options : variable entre 0 et 44 octets

Il existe 2 formats d'options : un seul octet de catégorie d'option ou un octet de catégorie d'option suivi d'un octet de longueur d'option et de l'octet des données de l'option.

4.2. Le fonctionnement du protocole TCP

D'après le format du message TCP, on peut résumer le fonctionnement de la couche transport de la modélisation TCP/IP en quatre points. Le protocole TCP de la couche transport :

- segmente les données des applications,
- établit une connexion de bout-en-bout,
- émet les segments d'un hôte à l'autre,
- assure la fiabilité du transport des segments entre les hôtes connectés.

4.2.1. La segmentation des données

La fonction de transport est effectuée segment par segment de façon autonome. Plusieurs applications peuvent émettre leurs segments successivement. Ces segments peuvent avoir un ou plusieurs destinataires.

[Segmentation des données des applications par la couche Transport - image seule](#)⁸

4.2.2. L'établissement de la connexion

Les étapes de l'établissement de la connexion sont :

[Etablissement de la connexion - image seule](#)⁹

4.2.3. Contrôle de flux

Un contrôle de flux est nécessaire pour prévenir la congestion des transferts :

- Un serveur performant peut générer plus de trafic que le réseau ne peut en supporter.
- Un serveur peut être sollicité par un nombre très élevé de clients.

[Contrôle de flux TCP - image seule](#)¹⁰

4.2.4. fenêtrage

Un dispositif de fenêtrage de l'acquittement a été implanté pour préserver la bande passante. Si l'émetteur devait attendre un acquittement après l'émission de chaque segment, la qualité de la transmission serait dégradée.

⁸ <http://www.linux-france.org/prj/inetdoc/articles/transmission.tcpip/images/tcp-segment-480.png>

⁹ <http://www.linux-france.org/prj/inetdoc/articles/transmission.tcpip/images/tcp-connect-480.png>

¹⁰ <http://www.linux-france.org/prj/inetdoc/articles/transmission.tcpip/images/tcp-stream-480.png>

Fenêtrage de l'acquittement - image seule¹¹

4.3. Le protocole UDP

Le protocole UDP est apparu avec le développement des réseaux locaux dont la fiabilité permet de s'affranchir des fonctions de contrôle. C'est un protocole minimum sans garantie de délivrance des messages et sans séquençement.

Format de l'en-tête UDP - image seule¹²

Source Port : 16 bits

Numéro du port source. Ce champ est optionnel.

Destination Port : 16 bits

Numéro du port destination.

Length : 16 bits

Longueur en octets du datagramme UDP incluant l'en-tête et les données.

Checksum : 16 bits

Somme de contrôle sur 16 bits de l'en-tête et des données.

5. En guise de conclusion

Cette présentation est loin d'être exhaustive. Pour aller plus avant dans la compréhension des mécanismes de fonctionnement des protocoles, voici quelques documents de référence :

Modélisations réseau

Modèles OSI & Internet

Le document *Modélisations réseau*¹³ est une synthèse comparative entre les modélisations OSI et Internet. On montre que la modélisation OSI correspond au volet transmission de l'information et que le modèle Internet correspond au volet traitement de l'information. Les protocoles TCP et IP constituent la charnière entre les deux modèles.

Technologie Ethernet

Principes & normalisations

Le document *Technologie Ethernet*¹⁴ est une présentation générale des caractéristiques techniques d'Ethernet et de ses différentes normalisations.

RFC894 : *A Standard for the Transmission of IP Datagrams over Ethernet Networks*

Le document *RFC894 Standard for the transmission of IP datagrams over Ethernet networks*¹⁵ de 1984 spécifie la méthode standard d'encapsulation des datagrammes IP sur Ethernet.

RFC1042 : *A Standard for the Transmission of IP Datagrams over IEEE 802 Networks*

Le document *RFC1042 Standard for the transmission of IP datagrams over IEEE 802 networks*¹⁶ de 1988 spécifie la méthode standard d'encapsulation des datagrammes IP et des requêtes ARP sur les réseaux IEEE 802.

Valeurs du champ type de la trame Ethernet II

Le fichier *ETHER TYPES*¹⁷ contient la liste des différentes valeurs du champ type des trames Ethernet II.

¹¹ <http://www.linux-france.org/prj/inetdoc/articles/transmission.tcpip/images/tcp-window-480.png>

¹² <http://www.linux-france.org/prj/inetdoc/articles/transmission.tcpip/images/udp-header-480.png>

¹³ <http://www.linux-france.org/prj/inetdoc/articles/modelisation/>

¹⁴ <http://www.linux-france.org/prj/inetdoc/articles/ethernet/>

¹⁵ <http://www.faqs.org/rfcs/rfc894.html>

¹⁶ <http://www.faqs.org/rfcs/rfc1042.html>

¹⁷ <http://www.iana.org/assignments/ethernet-numbers>

Protocoles Internet

RFC791 : *INTERNET PROTOCOL*

Le document [RFC791 Internet Protocol](#)¹⁸ de 1981 spécifie le protocole IP.

Adressage IP

La maîtrise de l'adressage IP est essentielle à toutes les manipulations autour des réseaux. Elle justifie un document à part entière ; c'est l'objet de l'article [Adressage IP](#)¹⁹.

Extrait du *Linux Advanced Routing & Traffic control Howto*

La section [Gestionnaires de mise en file d'attente simples, sans classes](#)²⁰ décrit l'utilisation des valeurs du champ *Type Of Service* (TOS) de l'en-tête IP avec le noyau Linux.

RFC793 : *Transmission Control Protocol*

Le document [RFC793 Transmission Control Protocol](#)²¹ de 1981 spécifie le protocole TCP.

¹⁸ <http://www.faqs.org/rfcs/rfc791.html>

¹⁹ <http://www.linux-france.org/prj/inetdoc/articles/adressage.ip/>

²⁰ <http://www.linux-france.org/prj/inetdoc/guides/lartc/lartc.qdisc.classless.html>

²¹ <http://www.faqs.org/rfcs/rfc793.html>