République Algérienne Démocratique et populaire
Ministère de l'Enseignement Supérieur et de la recherche Scientifique

Université Abderrahmane MIRA de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

Support de cours

Technologies Internet

Auteur : Dr. Abderrahmane SIDER

Table des matières

Chapitre 1 : Gouvernance, normes des réseaux et de l'Internet

1 Norme, normalisation et organismes de normalisation

Dans ce chapitre nous étudions la normalisation et celle des réseaux en particulier.

1.1.1 Définition de la normalisation

En France: Activité propre à établir, face à des problèmes réels ou potentiels, des dispositions destinées à un usage commun et répété, visant à l'obtention du degré optimal d'ordre dans un contexte donné. Le statut de la normalisation est régi en France par le Décret n° 2009-697 du 16 juin 2009.

En Algérie: L'activité propre à établir, face à des problèmes réels ou potentiels des dispositions destinées à un usage commun et répété, dans la confrontation des problèmes réels visant à l'obtention du degré optimal d'ordre dans un contexte donné. Elle fournit des documents de référence comportant des solutions à des problèmes techniques et commerciaux concernant les produits, biens et services qui se posent de façon répétée dans les relations entre les partenaires économiques, scientifiques, techniques et sociaux (Source: Loi 04-04 du 23 Juin 2004)

1.2 Définition de Norme

En Algérie: Document sans force obligatoire approuvé par un organisme de normalisation reconnu, qui fournit, pour des usages communs et répètes, des règles, des lignes directrices ou des caractéristiques, comprenant des prescriptions en matière d'emballage, de marquage ou d'étiquetage, pour des produits ou des procédés et des méthodes de production données (Source : Loi 04-04).

En France: Document, établi par consensus et approuvé par un organisme reconnu, qui fournit, pour des usages communs et répétés, des règles, des lignes directrices ou des caractéristiques, pour des activités ou leurs résultats, garantissant un niveau d'ordre optimal dans un contexte donné." (définition de l'ISO/IEC). Les organismes de normalisation reconnus sont: - au niveau mondial, l'ISO (Organisation internationale de normalisation), l'IEC (Commission électrotechnique internationale) et l'UIT (Union

internationale des télécommunications), - au niveau européen le CEN (Comité européen de normalisation) , le CENELEC (Comité européen de normalisation pour l'électrotechnique) et l'ETSI (institut européen des normes de télécommunications), - au niveau national, AFNOR, la Commission française pour l'ETSI, gérée par AFNOR, et l'UTE (union technique de l'électricité) qui est le membre français du CENELEC. Dans le cadre du Décret n° 2009-697 du 16 juin 2009, AFNOR anime le système central de normalisation composé des 25 bureaux de normalisation sectoriels, des pouvoirs publics et de 20 000 experts. AFNOR est le membre français du CEN et de l'ISO et assume les responsabilités attribuées à la France à ce titre. Des homologues d'AFNOR sont présents dans de nombreux pays, par exemple le DIN en Allemagne, BSI (British Standard Institute) au Royaume-Uni, ANSI (American National Standard Institute) aux Etats-Unis, IANOR en Algérie.

2 Types et appellations des normes

2.1 Norme de jure ou de droit

Norme définie et adoptée par un organisme officiel de normalisation, sur le plan national ou international. De jure (« from the law » ou « de droit ») s'oppose à de facto (« from the fact » ou « de fait »). Il est conseillé de réserver le terme norme au document qui est reconnu par un organisme officiel et standard à celui qui ne l'est pas, mais qui s'est imposé de soi.

2.2 Norme de facto ou Norme de fait

Norme qui n'a pas été élaborée, ni définie, ni entérinée par un organisme officiel de normalisation comme l'ISO, l'IANOR ou l'AFNOR, mais qui s'est imposée par la force des choses, parce qu'elle fait consensus auprès des utilisateurs, d'un groupe d'entreprises ou encore d'un consortium.

Une norme de facto peut découler des spécifications décrites par une seule entreprise (en anglais, *proprietary standard*). Par ailleurs, aujourd'hui, en informatique, il n'est pas rare qu'une norme de facto devienne une norme de jure.

2.3 Norme propriétaire

Ensemble de règles et de prescriptions techniques établies par une entreprise ou un groupement professionnel. Par exemple le protocole DecNet est propriété de Dec.

2.4 Norme internationale

Norme adoptée par l'ISO ou l'IEC (International Electronical Commission).

3 Organisme de Normalisation en Algérie[1]

L'institut Algérien de Normalisation (IANOR) a été érigé en établissement public à caractère industriel et commercial (EPIC) par Décret Exécutif n° 98-69 du 21 Février 1998 modifié et complété par le Décret exécutif Décret exécutif n° 11-20 du 25 janvier 2011. Il est chargé de :

1. L'élaboration, la publication et la diffusion des normes algériennes.

2. La centralisation et la coordination de l'ensemble des travaux de normalisation entrepris par les structures existantes et celles qui seront créées à cet effet.

3. L'adoption de marques de conformité aux normes algériennes et de labels de qualité ainsi que la délivrance d'autorisation de l'utilisation de ces marques et le contrôle de leur usage dans le cadre de la législation en vigueur.

4. La promotion de travaux, recherches, essais en Algérie ou à l'étranger ainsi que l'aménagement d'installations d'essais nécessaires à l'établissement de normes et à la garantie de leur mise en application.

5. La constitution, la conservation et la mise à la disposition de toute documentation ou information relative à la normalisation.

6. L'application des conventions et accords internationaux dans les domaines de la normalisation auxquels l'Algérie est partie.

7. Assure le secrétariat du Conseil National de la Normalisation (CNN) et des Comités Techniques de Normalisation.

[1] www.ianor.dz

L'Institut Algérien de Normalisation est en outre le point d'information algérien sur les Obstacles Techniques au Commerce (OTC) et ce conformément à l'accord OTC de l'Organisation Mondiale du Commerce.

4 De la gouvernance mondiale des réseaux

Les principaux organismes de normalisation, dans le domaine des réseaux numériques sont

ISO (International Standardization Organization)

IUT-T (International Union of Telecommunication - section Telecommunication) (ex-CCITT)

IEEE (Institute of Electrical and Electronic Engineers)

IETF / IRTF (Internet Engineering/Research Task Force)

ANSI, ECMA, AFNOR, etc.

4.1 Identification des normes et exemples de normes

La dénomination d'une norme doit tenir compte d'un ensemble de critères :

- son origine (ISO, IEEE, etc)
- son domaine d'application (réseaux publics/privés/locaux/, téléphone, etc)
- sa zone d'application (européenne, internationale, etc)

Exemples de normes :

- Les normes ISO sont préfixées par IS (International Standard)

ISO/ IS 8208 (=X.25/L3), ISO / IS 8802.3 (=Ethernet), etc.

- Les normes IUT-T sont nommées à l'aide d'une lettre suivie d'un point et d'un numéro :

IUT-T / X.25, IUT-T / X.400 (Messagerie), IUT-T / V.24 (Jonction pour la transmission de données numériques sur lignes téléphoniques), etc.

- Les noms des normes IEEE :

IEEE 802.5 (Token Ring), etc.

- Les normes de l'IETF/IRTF sont appelées des RFC ("Request For Comments") :

RFC 791 (Internet Protocol), RFC 768 (UDP), RFC 793 (TCP), ...

Deux modèles l'un de jure (OSI) et l'autre de facto « TCP/IP » sont actuellement les plus

connus. Nous les étudions dans les sections 5 et 6 respectivement.

5 Le modèle de référence OSI de l'ISO

5.1 Présentation

Norme de description de l'architecture générale des réseaux informatiques:

L'OSI = "Open Systems Interconnection : reference model"

Modèle en 7 couches

Les noms de la norme :

- ISO : IS 7498
- CCITT : X200 (nouvellement IUT-T)
- AFNOR : NF.Z.70.001

5.2 Notion de couche, de protocole et de service

Une couche est spécialisée dans un ensemble de fonctions particulières. Elle utilise les fonctionnalités de la couche inférieure et propose ses fonctionnalités à la couche supérieure.

Un système est un ensemble de composants formant un tout autonome.

Une entité est l'élément actif d'une couche dans un système.

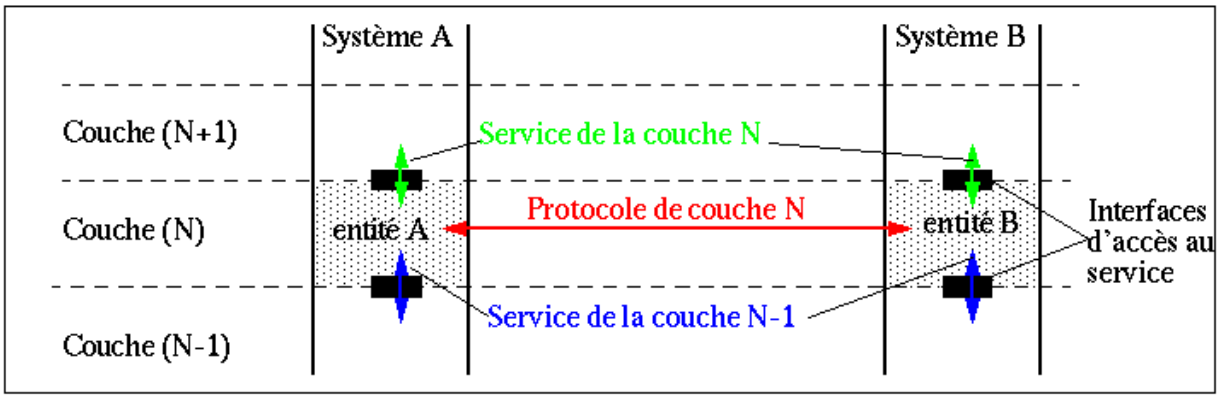
- entités homologues (paires) : entités de même couche situées dans des systèmes distants

Le protocole d'une couche N définit l'ensemble des règles ainsi que les formats et la signification des objets échangés, qui régissent la communication entre les entités de la couche N.

Le service d'une couche N définit l'ensemble des fonctionnalités possédées par la couche N et fournies aux entités de la couche N+1 à l'interface N/N+1.

Notation : on note N_X (ou encore X(N)) l'objet de type X appartenant à la couche N.

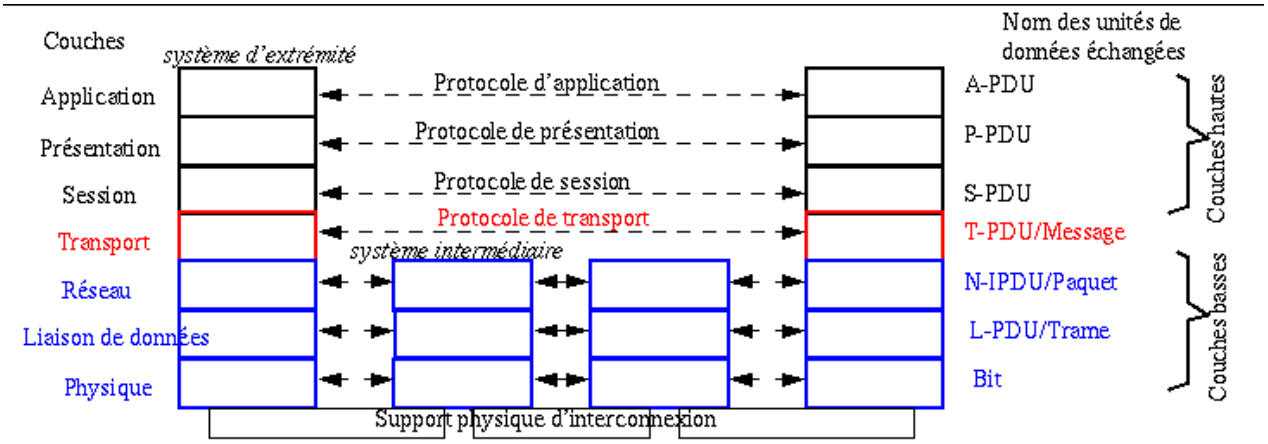
- ex : N-entité ou entité(N)



L'architecture d'un réseau est définie par l'ensemble des couches et la description des protocoles et des services de chacune d'elles.

5.3 Architecture générale du modèle OSI

Le modèle OSI possède sept couches



Le modèle décrit simplement ce que chaque couche doit réaliser (le service), les règles et le format des échanges (le protocole), mais pas leur implantation.

5.4 Les sept couches du modèle OSI

La couche Physique (couche 1)

Fournit les moyens mécaniques, optiques, électroniques, fonctionnels et procéduraux nécessaires à l'activation, au maintien et à la désactivation des connexions physiques nécessaires à la transmission de trains de bits.

Les systèmes sont interconnectés réellement au moyen de supports physiques de communication. Ces derniers ne font pas partie de la couche Physique.

La couche Liaison de données (couche 2)

Assure la transmission d'informations entre (2 ou plusieurs) systèmes immédiatement adjacents. Détecte et corrige, dans la mesure du possible, les erreurs issues de la couche inférieure. Les objets échangés sont souvent appelés trames ("frames").

La couche Réseau (couche 3)

Achemine les informations à travers un réseau pouvant être constitué de systèmes intermédiaires (routeurs). Les objets échangés sont souvent appelés paquets ("packets").

La couche Transport (couche 4)

Assure une transmission de bout en bout des données. Maintient une certaine qualité de la transmission, notamment vis-à-vis de la fiabilité et de l'optimisation de l'utilisation des ressources. Les objets échangés sont souvent appelés messages (de même pour les couches supérieures).

La couche Session (couche 5)

Fournit aux entités coopérant les moyens nécessaires pour synchroniser leurs dialogues, les interrompre ou les reprendre tout en assurant la cohérence des données échangées.

La couche Présentation (couche 6)

Se charge de la représentation des informations que les entités s'échangent. Masque l'hétérogénéité de techniques de codage utilisées par les différents systèmes.

La couche Application (couche 7)

Donne aux processus d'application les moyens d'accéder à l'environnement de communication de l'OSI. Comporte de nombreux protocoles adaptés aux différentes classes d'application.

Note : les fonctionnalités locales des applications proprement dites sont hors du champ de l'OSI donc de la couche Application !

Conclusion

Les trois premières couches constituent les couches basses où les contraintes du réseau sont perceptibles. Fonctions élémentaires spécialisées dans la transmission.

La couche Transport est une couche charnière, d'adaptation ou intermédiaire, associée le plus souvent aux couches basses.

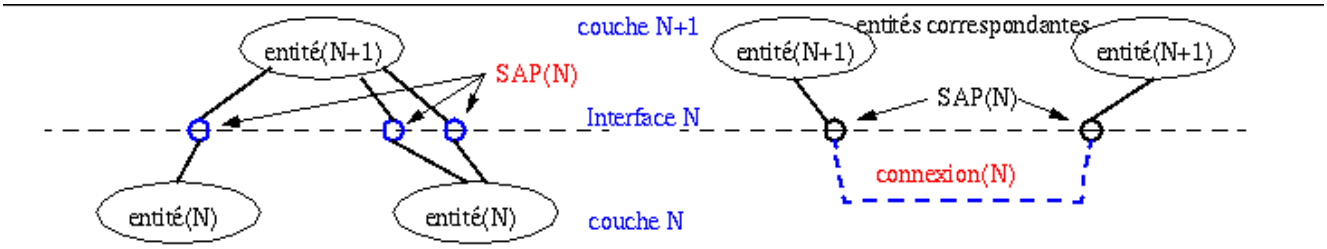
Les trois dernières couches constituent les couches hautes où les contraintes de l'application sont perceptibles. Fonctions complexes et variables adaptées aux traitements applicatifs.

Attention : La norme stipule clairement qu'il s'agit d'un modèle de référence et par conséquent, suivant le contexte dans lequel on se trouve et les besoins de communication, certaines fonctionnalités de certaines couches peuvent ne pas être utilisées (protocoles alternatifs, classes de protocole, options, etc.).

5.5 Connexion, SAP et types de connexion

SAP(N) : "service access point"

- point identifié où les services sont fournis par l'entité(N) à une entité(N+1)
  - ex : adresse



Connexion (N) : association d'entités homologues pour le transfert de données

- entités correspondantes : entités associées par la même connexion

Extrémité de connexion (N) : terminaison d'une connexion (N) à un SAP (N).

- connexion bipoint : connexion comportant exactement 2 extrémités.
- connexion multipoint : connexion comportant plus de 2 extrémités.

5.6 Les modes de communication

On peut distinguer deux grands modes de communication:

- communication en mode connecté (appelé aussi "with connection")
- communication en mode non connecté (appelé aussi "connectionless" ou par abus de langage "datagramme")

Le mode non connecté

- 1 seule phase (ou 0!) :
  - le transfert de données
- chaque unité de transfert de données est acheminée indépendamment
- les entités communicantes ne mémorisent rien ("memoryless").
- les messages échangées sont auto-suffisants ("self-content")

Le mode connecté

- 3 phases :
  - phase d'établissement de la connexion
  - phase de transfert de données
  - phase de libération de la connexion
- un contexte (réparti) est partagé par les membres de la connexion :
  - par exemple : le numéro du paquet
- permet (facilite) le contrôle et la gestion du transfert de données :
  - contrôle d'erreur, contrôle de flux, maintien en séquence, etc.
- les messages échangés comportent des informations qui ne sont utilisables que grâce à la connaissance de ce contexte :
  - par exemple : le numéro de paquet / la largeur de la fenêtre coulissante

5.7 Les primitives de service

Le protocole(N) s'appuie sur le service(N-1), donc la description du service est nécessaire à la compréhension du protocole.

Le service n'est accessible qu'à l'intérieur d'un **système** : la façon d'accéder au service ne doit pas être normalisée !

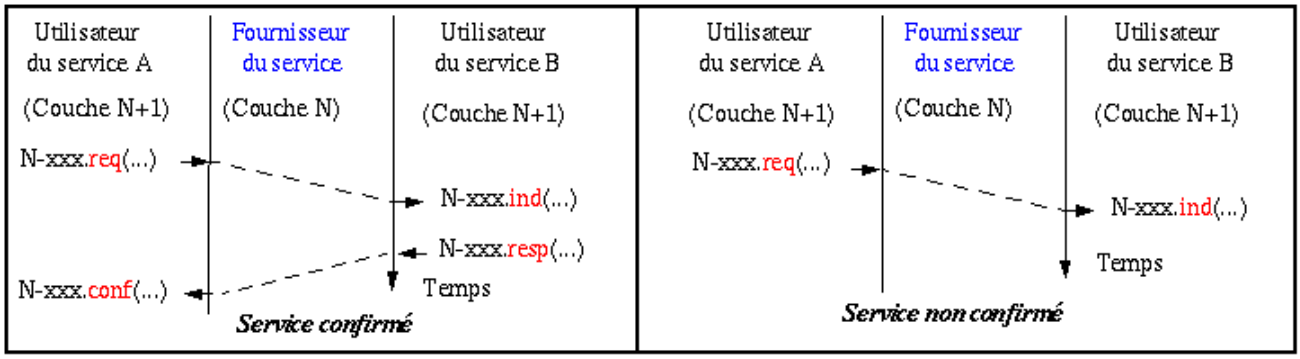
Les primitives spécifient le service :

- Ce sont des objets abstraits (puisque le service est abstrait !)
- Echangées à travers un interface idéal (sans perte et sans délai)
- Leur représentation ressemble à une procédure avec des paramètres
  - préfixe : l'initiale de la couche
  - suffixe : le type de primitive
  - Exemples : . T_Data.req(T_SDU) . LLC_Data.req(ad-locale, ad-distante, L_SDU, classe-de-service)

Il existe quatre types de primitives :

- Request : Une entité sollicite un service
- Indication : Une entité est informée d'une demande de service
- Response : Une entité a rendu le service, si possible
- Confirmation : Une entité est informée que le service a été rendu

Par exemple :



De très nombreuses variantes d'enchaînement des types de primitives de service [1 à 4] existent !

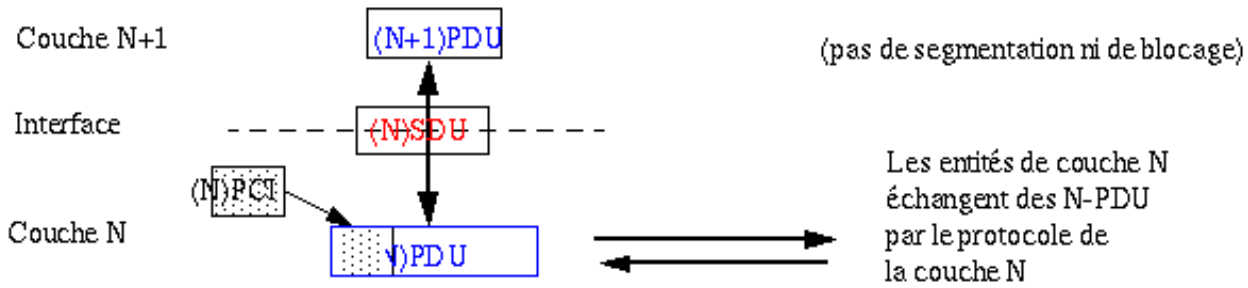
5.8 Les unités de données

SDU (N) :

- unité de données spécifique au service (N), dont l'intégrité est préservée d'une extrémité à l'autre d'une connexion.
  - peut être de taille quelconque.
  - exemple : un fichier à transmettre

PDU (N) :

- unité de données spécifique au protocole(N), adaptée à la transmission, constituée par les informations de contrôle du protocole (PCI(N)) et éventuellement par des portions de données issues du SDU(N)
  - par ex. : une trame, un paquet.



IDU(N) :

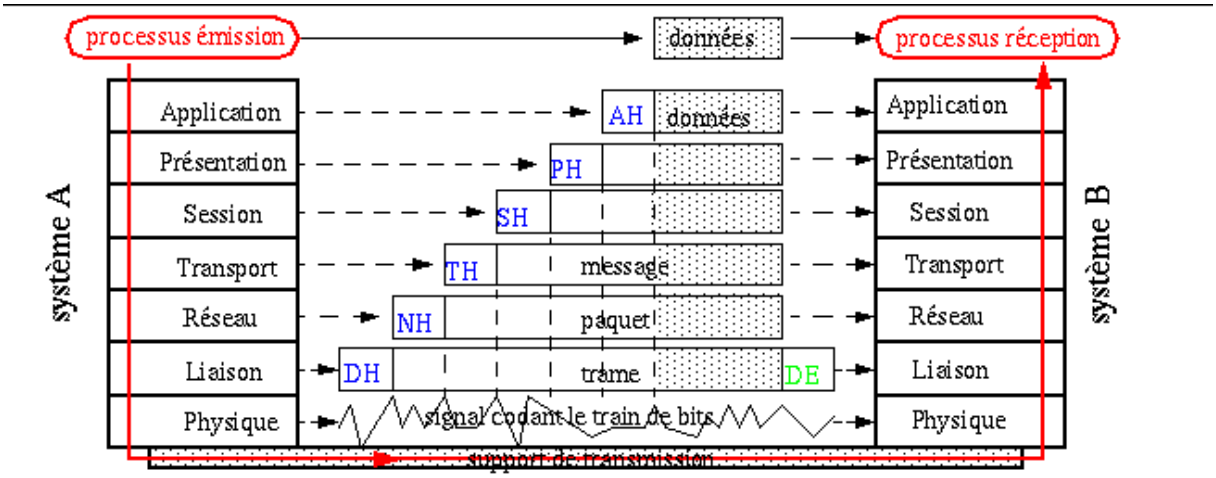
- unité d'information transférée en une seule interaction à l'interface de 2 couches, constituée d'information de contrôle d'interface (ICI(N)) et tout ou partie d'une SDU(N).
  - dépend du système d'accueil (notamment leur format).

- dépendant de l'implantation
- par ex. : les tampons (buffers) utilisés pour charger le fichier

5.9 Transmission des données

L'encapsulation :

Les données d'une couche sont encapsulées dans une unité de données de la couche inférieure. A la réception, les en-têtes sont enlevés à chaque passage d'une couche jusqu'à recouvrement des données à la couche application. Ce dernier processus est appelé dés-encapsulation.



6 Modèle TCP/IP, l'Internet et RFC

L'Internet Society (ISOC) créée en 1992, est un organisme international à but non lucratif qui supervise le développement de l'internet en veillant à ce qu'il reste ouvert. Il exerce une autorité technique et morale sur l'IAB et l'ICANN. Le modèle TCP/IP est composé de quatre couches (Application, Transport, Réseau, Hôte-Réseau) et sera abordé en détail dans le chapitre suivant. Ici nous allons nous contenter de présenter l'organisation de l'ISOC et le processus de création des fameuses RFC, normes de l'Internet.

6.1 L'IAB

L'Internet Architecture Board est un comité chargé du suivi de l'évolution des protocoles TCP/IP. Il supervise l'IETF et l'IRTF.

6.2 L'ICANN

L'Internet Corporation for Assigned Names and Numbers, est une entreprise créée en 1998 chargée de la gestion de plusieurs bases de données d'identifiants uniques. De plus, elle est charge de la distribution des adresses IP, de la gestion des noms de domaines de haut niveau (.com, .org, .uk, .dz, .fr …) des numéros identifiant les protocoles de l'Internet (Assigned Numbers) et maintient les serveurs DNS de la zone racine. Auparavant ses services étaient gérés par l'IANA (Internet Assigned Numbers Authority).

6.3 L'IRTF (Internet Research Task Force)

L'IRTF (www.irtf.org) est un organisme de l'ISOC chargé de :

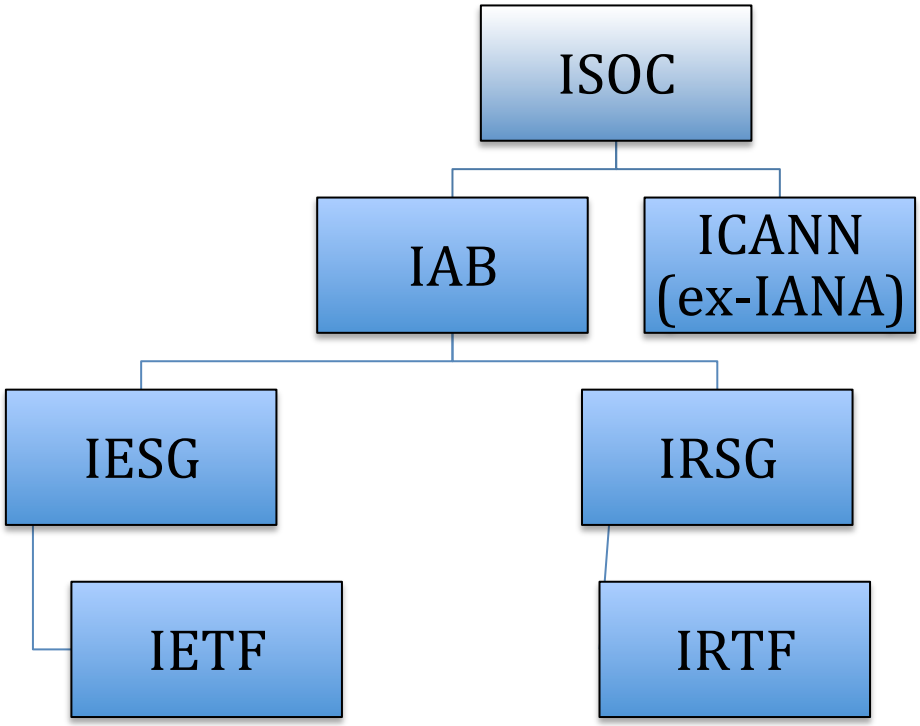
- prévoir l'évolution des protocoles et des technologies de l'Internet sur le long terme
- prépare les futurs travaux de l'IETF
- est géré par l'IRSG (Steering Group)

6.4 l'IETF (Internet Engineering Task Force)

L'IETF :

-Etablit les spécifications et réalise les premières implantations des nouveaux protocoles du modèle TCP/IP

Produit des normes Internet appelées RFC (Request For Comment)



6.5 Le processus IETF (Internet Engineering Task Force)

La devise de l'IETF est : «Le consensus sur l'essentiel, et du code qui fonctionne ». L'IETF est composé de groupes de travail qui établit réellement une norme. L'unanimité du groupe de travail (working group) n'est pas indispensable à l'adoption d'une proposition mais une proposition qui ne rallie pas la majorité des membres du groupe ne sera pas acceptée. Il n'est pas requis qu'une proposition soit défendue par un pourcentage donné des membres du groupe mais la plupart des propositions qui sont défendues par plus de 90% des membres ont des chances d'être acceptées, quant à celles qui sont défendues par moins de 80% des membres, elles sont souvent rejetées. Les groupes de travail de l'IETF ne votent pas vraiment mais peuvent recourir à un vote à main levée pour vérifier que le consensus est atteint.

Les documents de type non standard peuvent émaner de l'activité des groupes de travail de l'IETF, ou de personnes indépendantes souhaitant que leurs réflexions ou leurs propositions techniques soient accessibles à la communauté de l'Internet. Presque toutes les propositions pour une publication dans les RFC (Request for Comment) sont revues par l'IESG (Internet Engineering Steering Group), qui conseillera ensuite l'éditeur des RFC quant à l'intérêt de la publication du document. L'éditeur des RFC décidera alors de publier ou non le document et si l'IESG a rédigé une note, de l'inclure ou non dans le document. Ces notes de l'IESG servent à marquer certaines réserves par rapport

à la proposition, l'IESG ayant alors le sentiment qu'un avertissement quelconque à l'utilisateur peut être utile.

Pour un document standard, normalement, un groupe de travail de l'IETF produira un « brouillon Internet », destiné à être publié dans les RFC. L'étape finale de l'évaluation de la proposition au sein du groupe de travail est un « dernier appel », qui dure en principe deux semaines, pendant lequel le président du groupe de travail demande dans la liste de diffusion du groupe si la proposition connaît des problèmes notables. Si, à la suite de ce « dernier appel », il apparaît que le groupe est d'accord sur l'acceptation de la proposition, celle-ci est alors transmise à l'IESG, pour évaluation. La première étape de l'évaluation par l'IESG consiste en un « dernier appel » à l'échelle de l'IETF, envoyé à la principale liste de diffusion de l'IETF. Les personnes qui n'ont pas suivi les travaux du groupe peuvent ainsi apporter leurs commentaires à la proposition. En principe, ce « dernier appel » dure deux semaines pour les propositions qui viennent de groupes de travail de l'IETF et quatre semaines pour les autres.

L'IESG prend les résultats du « dernier appel » à l'IETF comme base pour ses délibérations au sujet de la proposition. L'IESG peut approuver le document et demander sa publication ou renvoyer la proposition à son (ses) auteur(s) pour des révisions fondées sur l'évaluation qu'elle en a faite. C'est ce même processus qui est utilisé à chaque étape pour les documents standard.

Normalement, les propositions sont intégrées aux documents standard au titre de *standards proposés* (proposed standard) mais lorsque des incertitudes pèsent quant à l'adéquation de l'approche ou si une phase de tests supplémentaire paraît nécessaire, un document est d'abord publié sous statut *expérimental*. Les standards proposés sont censés être de bonnes idées sans problème technique connu. Après six mois au minimum comme standard proposé, l'accession d'une proposition au statut de *pré-standard* peut être discutée. Les pré-standards doivent avoir fait preuve de la clarté de leur documentation et démontré que tous les problèmes de droit d'auteur éventuels sont repérés, et solubles. On y arrive en exigeant au moins deux implémentations génétiquement indépendantes et inter-opérables de la proposition, avec, le cas échéant, des exercices séparés de procédures d'obtention de licences. Ce qui implique aussi que toutes les caractéristiques séparées du protocole soient implémentées à plusieurs reprises. Toute caractéristique ne répondant pas à ces exigences doit être retirée avant que le processus puisse continuer à se dérouler. Les standards IETF peuvent donc se

simplifier au fur et à mesure qu'ils progressent. On retrouve la seconde moitié de la devise de l'IETF : « du code qui fonctionne ».
La dernière étape du processus de standardisation IETF est le *standard Internet*. Si la proposition a connu un succès commercial significatif après au moins quatre mois au statut de pré-standard, son accession au statut de standard Internet peut être discutée.
Deux différences essentielles sont notables si l'on compare le type *standard* de l'IETF avec la manière dont les choses se passent dans d'autres organisations de standardisation. D'abord, le résultat final de la plupart des processus de standardisation est à peu près équivalent au statut de *standard proposé* pour l'IETF. Une bonne idée mais pas d'exigence d'un code qui fonctionne vraiment. Ensuite, le consensus sur l'essentiel au lieu de l'unanimité peut donner des propositions avec moins de caractéristiques ajoutées destinées à faire taire une objection particulière.
Si une RFC n'est pas dans le processus standard (standard track) alors elle est soit historic (standard ancien, désuet) ou best-current practices (conseils et recommnedations pour l'implantation des standards ou de type informational (Documentation informative non normalisée)

Les RFCs sont numérotées sous la forme NNNN et sont pubilées sous forme de fichiers texte ayant un nom sous la forme nnnn.txt. On peut accéder au texte de toute RFC dont on connait le numéro en tapant directement son URL dans un navigateur sous la forme www.ietf.org/rfc/rfcNNNN.txt. Si on ne connait pas le numéro on peut se servir d'un moteur de recherche sur les RFCs à l'addresse www.ietf.org/rfc.html.

7 Conclusion

Dans ce chapitre nous avons que les équipements élctroniques et les réseaux d'information sont en fait régis par des normes techniques. Ces normes peuvent être de différents types et sont définis par des organismes distincts. La norme internationale des réseaux est le modèle OSI, mais en pratique, pour les réseaux WAN par paquets et les LAN dernièrement, la norme TCP/IP définie par l'IETF est une norme de facto. Dans ce chapitre, nous avons revu la norme OSI, et dans le prochain chapitre nous verrons en détail les protocoles de Transport et la norme TCP/IP elle-même.

Chapitre 2: Le modèle TCP/IP, TCP et UDP

1 Généralités

1.1 Protocoles de transport OSI

OSI a défini 5 classes de TP (Transport Protocol) :

- TP0 : Pas de multiplexage, Pas de reprise sur erreurs
- TP1 : Pas de multiplexage, Reprise sur erreurs signalées par le niveau Réseau
- TP2 : Multiplexage, Avec ou sans contrôle de congestion, Pas de reprise sur erreurs
- TP3 : Multiplexage, Avec ou sans contrôle de congestion, Reprise sur erreurs signalées
- TP4 : Multiplexage, Avec ou sans contrôle de congestion, Reprise sur erreurs signalées et non signalées

TP classe 0 : adaptée aux réseaux fiables

TP classe 4 : adaptée aux réseaux **non fiables (ex. IP)**

1.2 Protocoles de transport TCP/IP

- ***UDP*** *(User Datagram Protocol)* **~ TP classe 0**
  Défini en 1980 (RFC 768)
  Très simple : aucun contrôle d'erreurs ou de congestion
  Adapté aux réseaux fiables ou bien si la fiabilité n'est pas requise
  Agressif pour le réseau (peu recommandé en général)
- ***TCP*** *(Transmission Control Protocol)* **~ TP classe 4**
  Défini en 1981 (RFC 793)
  Complexe : connexions fiables, contrôle de flux, contrôle de congestion
  Adapté aux réseaux non fiables
  Très adaptatif (très recommandé en général)
- ***DCCP*** *(Datagram Congestion Protocol)* **~ TP classe 2**
  Défini en 2006 (RFC 4336)
  Complexité moyenne : connexions non fiables, avec contrôle de congestion
  Corrige le principal défaut de UDP (l'agressivité)

1.3 Origines de l'Internet

En 1962: DARPA a initié un projet de recherches sur un réseau qui permettra l'interconnexion d'ordinateurs appartenant à des structures différentes du gouvernement (il y a là un besoin d'ouverture de ces structures sur leur environnement).

En 1969 : Quatre ordinateurs hébergés au MIT, l'UCLA, Stanford, Santa Barbara et l'UTAH sont connectés dans le premier réseau de données informatique : "l'ARPANET"

Entre 1972-1990: le NCP (Network Control Protocol) est remplacé par le TCP/IP. Des applications apparaissent (la messagerie en 1972, le DNS en 1982, le WWW en 1989).

1990- Aujourd'hui : Le réseau qui transportaient des fichiers de données, des messages et des documents transporte aujourd'hui des échantillons de voix sur IP (VoIP) grâce à des protocoles adaptés, suivi du développement des applications de télévision/radio (streaming).

Les protocoles IP, TCP et UDP ainsi que tous les protocoles applicatifs qui ont été développés font partie de ce qui a été convenu d'appelé le modèle TCP/IP. Celui-ci a été normalisé par l'IETF. Il est constitué de quatre couches: La couche hôte-réseau regroupe les fonctionnalités des couches liaison de données et physique du modèle OSI et où les couches transport et application sont des couches de bout en bout. Une couche est dite de bout en bout si elle est implantée sur les entités terminales de communication mais pas sur les entités intermédiaires (routeur, switchs).

Pile TCP/IP

Applications (HTTP, SMTP, POP, FTP, DNS, etc.)
Transport (TCP/UDP, etc.)
Internet (IP, ICMP, RIP, OSPF, etc.)
Hôte-Réseau

2 Protocole UDP (User Datagram Protocol)

UDP (RFC 768) est un protocole non fiable et sans connexion. Il permet à une application d'envoyer un message à une autre avec un minimum de fonctionnalités (pas de garantie d'arrivée, pas de contrôle de flux, ou de congestion.) Il est parfait pour les applications

telles que les requêtes de configuration dynamique DHCP, les requêtes DNS et les échanges RIP ou dans les applications audio et vidéo.

Le champ « protocole » du paquet IP transportant un datagramme UDP contient la valeur 17. Dans la figure suivante, on voit bien que l'entête offre des fonctionnalités minimales, avec les ports désignant les processus, un champs longueur donnant la taille du champs « Données », et un CRC.

Port Source	Port Destination
Longueur	Checksum
Données provenant de la couche supérieure	

3 TCP

TCP a pour objectif de fournir un service de communication de **processus à processus**, dans un environnement réseau complexe. Le champ « protocole » du paquet IP transportant un segment TCP contient la valeur 06.

3.1 Motivations

TCP fournit un moyen d'établir une communication fiable entre deux tâches exécutées sur deux ordinateurs autonomes raccordés à un réseau de données. Le protocole TCP s'affranchit le plus possible de la fiabilité intrinsèque des couches inférieures de communication sur lesquelles il s'appuie. TCP suppose uniquement que les couches de communication qui lui sont inférieures lui procurent un service simple de transmission de paquet sans garantie de qualité de service.

TCP s'intègre dans une architecture multicouche des protocoles, juste au-dessus du protocole Internet IP. Ce dernier permet à TCP l'envoi et la réception de segments de longueur variable, encapsulés dans un paquet Internet appelé aussi "datagramme". Le datagramme Internet dispose des mécanismes permettant l'adressage d'un hôte IP source et un destinataire, quelles que soient leur position dans le réseau. Le protocole IP s'occupe aussi de la fragmentation et du réassemblage des segments TCP lors de la traversée de réseaux de plus faibles caractéristiques (la MTU en particulier). Le protocole IP transporte aussi les informations de priorité, compartimentation et classification en termes de sécurité relatives aux segments

TCP. Ces informations se retrouvent alors transmises de bout en bout de la communication.

3.2 Fonctionnement TCP

Comme introduit, TCP est conçu pour fournir un service de transmission de données fiable entre deux processus s'exécutant sur deux machines raccordées sur un réseau de paquets comme IP. Pour pouvoir assurer ce service même au dessus d'une couche de protocole moins fiable, les fonctionnalités suivantes sont nécessaires:

8. Transfert de données de base
9. Correction d'erreur
10. Contrôle de flux
11. Multiplexage
12. Gestion de connexions
13. Priorité et Sécurité

Ces fonctionnalités sont décrites en grandes lignes dans les paragraphes qui suivent.

3.2.1 Transfert de données de base

TCP est capable de transférer un flux continu de données entre deux processus, en découpant ce flux en « chunck » « petites pièces ». En général, TCP décide de lui-même là où le flux de données doit être coupé. Chaque « pièce » est mise ensuite dans un segment TCP. Le champ « numéro séquence » indiqué dans chaque segment fournit au TCP récepteur le numéro du premier octet des données dans ce segment par rapport au flux global découpé. De cette façon, le récepteur peut reconstituer le flux initial même si les segments arrivent dans un ordre différent de celui dans lequel ils ont été envoyés.

Parfois on a besoin de savoir que toutes les données soumises à TCP ont bien été émises localement (le buffer entre la couche application et la couche TCP est vidé) et/ou ont été bien remises à l'autre bout de la connexion au processus homologue (dans ce cas-là, le contenu du buffer TCP est passé à la couche application même s'il n'est pas encore plein comme le prévoit le fonctionnement de base). Pour cette fonction on activera la fonction "push" de TCP (le flag PSH).

3.2.2 Contrôle d'erreur

TCP doit considérer et traiter les cas de données perdues, erronées, dupliquées, ou arrivées dans le désordre à l'autre bout de la liaison Internet. Ceci est réalisé par l'insertion d'un numéro de séquence, et par l'obligation d'émission d'un "accusé de réception" (ACK) par le

TCP destinataire. Si l'accusé de réception n'est pas reçu au bout d'un temps prédéfini, le paquet sera réémis. Côté récepteur, les numéros de séquence sont utilisés pour reconstituer dans le bon ordre le flux original, et éliminer les paquets dupliqués. L'élimination des erreurs physiques de transmission se fait par encodage d'un Checksum à l'émission, re-calcul de ce Checksum par le destinataire, et élimination des paquets pour les quels les deux valeurs ne correspondent pas.

Tant que TCP fonctionne correctement, et que le réseau Internet n'est pas saturé, aucune faute de transmission ne devrait transparaître dans la communication. TCP est donc sensé récupérer les erreurs de la transmission Internet.

3.2.3 Contrôle de flux

TCP fournit un moyen au destinataire pour contrôler le débit de données envoyé par l'émetteur. Cet objectif est atteint en retournant une information de "fenêtre" avec chaque accusé de réception (tout segment ayant le bit ACK à 1) indiquant la capacité de réception instantanée en termes de numéros de séquence. Ce paramètre noté "window" indique le nombre d'octets que l'émetteur peut envoyer avant une autorisation d'émettre ultérieure.

3.2.4 Multiplexage

Pour permettre à plusieurs tâches d'une même machine de communiquer simultanément via TCP, le protocole définit un ensemble d'adresses et de ports pour la machine. Un "**socket**" est défini par le couple adresse IP plus un port sur la même machine (le champs port). Une connexion nécessite la mise en place de deux sockets l'une sur celui qui reçoit la demande de connexion et l'autre sur celui qui l'initie. Une socket peut être utilisée par plusieurs connexions distinctes sur la même machine comme dans le cas d'un socket serveur web.

L'affectation des ports aux processus est établie par chaque ordinateur. Cependant, il a semblé judicieux de réserver certains numéros de ports pour des services caractérisés et souvent utilisés. Ces services standard sont alors être atteints via ces ports "réservés". L'affectation des ports réservés est gérée par l'ICANN.

3.2.5 Connexions

Les mécanismes de fiabilisation et de contrôle de flux décrits ci-dessus imposent à TCP l'initialisation et la maintenance de certaines informations pour chaque communication. La combinaison de ces informations, dont les sockets, les fenêtres, et les numéros de séquence formeront ce que nous appelons une connexion. Chaque connexion est identifiée de manière unique par sa paire de sockets, définissant chacun des deux sens de la communication.

Lorsque deux processus désirent communiquer, leur TCP respectifs doivent tout d'abord négocier et établir une connexion (initialiser ces informations d'état de part et d'autre). Lorsque la communication s'achève, elle sera fermée, en libérant ses ressources à d'autres usages.

3.2.6 Priorité et Sécurité

Les exploitants de TCP peuvent indiquer le degré de sécurité et la priorité de la communication établie. TCP permet cependant de ne pas traiter ce besoin.

3.3 Format de l'en-tête

Les segments TCP sont envoyés encapsulés dans des datagrammes Internet. L'en-tête IP transmet un certain nombre de paramètres, tels que les adresses Internet source et destinataires. L'en-tête TCP est placé à la suite, contenant les informations spécifiques au protocole TCP.

0 31

Port Source								Port Destination
Numéro de séquence								
Numéro d'acquittement								
Data Offset	Réservé	U	A	P	R	S	F	Fenêtre
Checksum								Pointeur de données urgentes
Options								Bourrage
Données provenant de la couche application								

Notez qu'une case représente une position bit.

Afin de comprendre les champs de l'en-tête du protocole TCP, il convient de revenir sur les besoins pour lesquels TCP a été créé.

TCP (Transmission Control Protocol) défini dans la RFC 793 est le successeur du protocole NCP (Network Control Protocol), le premier protocole de transport de l'ARPANet. L'objectif de TCP est de fournir un transfert **fiable,** entre **deux processus distants (de bout en bout),** d'un **flux d'information** qui est de taille finie mais

inconnue au début du transfert. Le flux est **bidirectionnel** entre les deux processus distants.

3.3.1 Port, Port source et port destination (16 bits)

Commençons par l'identification des processus distants eux-mêmes. Les **hôtes** sur lesquels les deux processus s'exécutent sont supposés abriter la couche Internet et la couche Transport. Au niveau de la couche Internet, chaque hôte est identifié par son adresse IP. La question qui reste est donc celle de distinguer les processus s'y exécutant. « Le port » est l'identifiant du processus émetteur/récepteur. Le « port source » est donc l'identifiant du port du processus émetteur. Le port destination est l'identifiant du processus destination.

Lorsque on observe un échange TCP et même UDP, on remarque qu'entre deux hôtes A et B, les segments partant de A vers B comportent la même paire (x=port source sur A, port Destination sur B=y). De même les segments en provenance de B vers A ont toujours la paire (y=port source sur B, port destination sur A=x). Il y a donc deux processus identifiés par x sur A et y sur B qui communiquent. Lorsque un processus quelconque souhaite utiliser la couche réseau de son hôte, il fait une demande au système d'exploitation (module réseau), le système répond en lui donnant le numéro de port et sauvegarde l'association entre le numéro de processus et le numéro de port. Ainsi, lorsque des segments arrivent plus tard ayant comme port destination une valeur donnée, le module réseau cherche dans cette table et identifie le processus à qui les données doivent être remises.

3.3.2 Numéro de Séquence

Avec TCP, l'hôte A doit pouvoir transmettre de façon fiable un message vers la couche TCP d'un hôte B. Il est possible évidemment d'envoyer le message dans un seul segment si on connaissait sa taille au début de la transmission, ce qui n'est pas souvent le cas (un flux de streaming audio/vidéo, un capteur qui transmet de façon continue les valeurs de température de son environnement, par exemple). Même si on pouvait transmettre le message dans un segment TCP, le problème qui se poserait est que pendant la transmission du message, les liens de communications seront occupés par ce seul message (et donc d'autres hôtes ne pourront pas communiquer en utilisant les mêmes

liens), d'autres part, il suffit d'une seule erreur physique de transmission pour que le message soit corrompu, et dans ce cas on doit retransmettre tout le message. La solution est donc de découper le message en petites parties appelées segments « chuncks », de les envoyer au récepteur qui recollera les segments dans le bon ordre afin de reconstituer le message entier transmis initialement par A. Dans ce cas, une erreur de transmission sur un segment déclenchera la **retransmission** de seulement ce segment corrompu et non pas de tout le message, ce qui permet de minimiser l'utilisation de la bande passante du réseau.

Le numéro de séquence (sur 32 bits) contenu dans les segments allant de A vers B identifie le numéro d'ordre du premier octet du champs « données » du segment TCP dans le flux transmis par A.

Sans ce numéro de séquence, la couche TCP sur B ne pourra pas réordonner les segments qui lui arrivent dans un ordre différent de celui de leur envoi. Le protocole IP en effet utilise un protocole de routage dit « best effort », c'est-à-dire la couche IP tente de remettre les paquets allant de A vers B mais elle n'assure pas une remise dans le même ordre que celui d'envoi. Les routes sur Internet changent fréquemment et il arrive souvent que deux paquets en cours de transmission d'un hôte A vers un hôte B suivent deux chemins différents. Et c'est pour cette raison que l'ordre d'arrivée peut différer de l'ordre d'envoi. Le Numéro de Séquence permet ainsi de rendre la couche transport indépendante du comportement de la couche Internet. Rappelons que si les donnés ne sont pas « recollées » dans le bon ordre, TCP ne serai pas « fiable ».

3.3.3 Numéro d'Acquittement

Le Numéro d'Acquittement sert à l'hôte B pour dire à l'hôte A quel **prochain** octet il attend de lui ou quel est l'octet attendu de A dans le prochain segment qu'il transmettra vers B. La valeur de ce champs n'est valable (A le prend en considération) que si le drapeau « ACK » est aussi mis à 1 dans le même segment. Ainsi si A a transmis un segment avec Numéro de Séquence égal à N et taille égale à S, à un certain moment il doit recevoir un segment de B comportant comme Numéro Acquittement la valeur N+S+1. Autrement, l'hôte A continuera de retransmettre le segment en question. Notons que tant que ce numéro d'acquittement N+S+1 n'est pas reçu par A, sa liste de « segments en attente d'acquittement » continuera à inclure le segment original, ce qui occupe de la mémoire sur A. Lorsque A reçoit ce numéro d'acquittement, il peut enlever

le segment en question de sa liste, ce qui libère de la mémoire sur l'hôte A. C'est pourquoi il est important pour TCP de confirmer l'arrivée des segments aussi rapidement que possible. Car sur les deux hôtes A et B, il peut y avoir des dizaines de processus utilisant TCP au même moment, ce qui aura un grand impact sur l'utilisation mémoire, notamment sur les serveurs réseau (serveur web, de messagerie, ftp…) et les appareils mobiles (téléphones intelligents, lecteur de poche, montre intelligente etc.). Si chaque octet n'était pas confirmé par TCP, la fiabilité de la transmission ne serait pas assurée, le champs Numéro Acquittement sert exactement cette fonction.

3.3.4 Champ Checksum

Chaque segment doit pouvoir être vérifié s'il a été correctement transmis. Le champs Checksum (Somme de contrôle) sert à calculer un hash sur le champ data + une pseudo-en-tête TCP. A la réception d'un segment donné, l'hôte B recalcule la somme avec le même algorithme sur le champs data plus la pseudo-entête). Le pseudo-en-tête ne contient pas le champ checksum puisque il n'est pas encore calculé ! C'est donc la vérification basée sur ce champ qui déclenche l'Acquittement du segment avec un Numéro d'Acquittement égal à S+N+1. Ainsi ce champ sert lui aussi à assurer que chaque segment est transmis de façon fiable.

3.3.5 Champ 'Data Offset

Ce champ indique la taille de l'en-tête TCP comme un multiple de 32 bits (4 octets). L'en-tête TCP a une longueur variable à cause du champ 'Options' qui a une taille variable, les options TCP peuvent avoir en effet une taille de 1, 2, ou 4 octets. Si la taille de l'en-tête avec les options n'est pas un multiple de 4, des octets de bourrage appartenant au « champ Padding » sont rajoutés.

3.3.6 Champ Padding

Ou 'Données de bourrage'. Données rajoutées juste pour que les données du segment TCP commence sur un octet multiple de 4.

3.3.7 Champ Urgent Pointer

Ou 'Pointeur de données urgentes', numéro d'octet du champs data du segment où commencent les données urgentes. Les données urgentes sont des données hors du flux

normal insérées par l'application en réaction à un évènement exceptionnel. En mettant le drapeau 'Urgent ' à 1, une entité TCP peut indiquer à son entité homologue de prendre en considération les données urgentes dans le segment TCP qui sont mises en plein dans les données du flux normal. Le champs Urgent pointer constitue l'adresse du premier octet des données urgentes. Les données urgentes sont remises immédiatement par le TCP récepteur au processus en communication.

3.3.8 Champ 'Control bits' appelés aussi flags

Ou 'Drapeau'. Un ensemble de bits appelés eux-mêmes drapeau. Les bits les plus significatifs et les plus utilisés sont notés par leur une lettre majuscule ou par trois lettres majuscules qui résument leur nom. Ainsi UAPRSF où A correspond à ACK, S à SYN, U à URG, R à RST, P à PSH et F à FIN indiquent les bits du champs drapeau qui s'appellent Aknowledgemnt (Acquittement), Synchronize (Synchronisation), Urgent (Données urgentes), Reset (Réinitialisation), Push (Pousser) et Finalize (Terminaison) respectivement.

SYN est mis à un uniquement lors de la phase d'ouverture de connexion

ACK sert à rendre significatif le Numéro Acquittement

URG sert à signaler la présence de données urgentes. Dans ce cas le champ 'Pointeur de données urgentes' devient significatif.

RST sert à réinitialiser la connexion

PSH sert à indiquer au TCP récepteur de remettre immédiatement les données qui se trouvent dans son buffer au processus récepteur et de ne pas attendre qu'il soit plein.

FIN sert uniquement dans la phase de fermeture de connexion

3.3.9 Champ WINDOW

Ou 'Fenêtre de réception'

Ce champ a pour objectif de permettre au récepteur d'indiquer à l'émetteur sa capacité de réception. Au début de l'internet la taille mémoire des hôtes TCP était très limitées. Il convenait donc d'informer le processus émetteur de la taille des buffers dont dispose le récepteur en un instant donné. Elle permit donc de mettre en place un contrôle de flux. Plus tard, on se rendit compte que pour permettre à TCP d'augmenter son débit, on devait augmenter le nombre de segments en cours de transit, ou ce qu'on appelle la fenêtre d'émission. En effet si on assimile la connexion entre deux entités A et B à un tube de longueur Delta (Delta est le temps d'aller-retour ou Round Trip Time divisé par

deux) et que le débit de la connexion est R alors on devait mettre R*Delta octets dans le tube pour le remplir. C'est aussi le nombre d'octet que l'émetteur mette dans le tube avant que ne le premier octet n'arrive au récepteur. On voit donc que le nombre de segments en transit varie au cours du temps en fonction du RTT. Plus le RTT augmente, plus l'émetteur met plus de données dans la connexion. Mais à un certain moment, en fonction de la surcharge des routeurs intermédiaires, quelques uns de ses segments peuvent être perdus. Les algorithmes de détection et de gestion de la fenêtre d'émission ont été alors développés afin de prendre en compte la congestion dans le réseau (voir plus loin les algorithmes 'Slow start', 'Tahoe' et 'Reno'.)

Champ 'Option' (0 à 40 octets)

Le champ Option répond au besoin d'étendre les fonctionnalités de TCP sans modifier radicalement la structure de l'en-tête TCP. Les options sont le dernier champ de l'en-tête avant les données elles-mêmes. Une option peut tenir sur une taille de 1, 2, 3, 4, 6, 8, 10 ou 18 octets selon son type et ses paramètres. C'est la raison pour laquelle un champ appelé 'Padding' est prévu après les options. Il permet d'arrondir la taille de l'en-tête à un multiple de 4 octets pour des raisons de performances. Certaines options sont utilisées uniquement pendant la phase d'ouverture de connexion. La négociation des options communes est d'ailleurs l'un des objectifs de la phase d'ouverture de connexion, en plus de la synchronisation des numéros de séquence initiaux. Dans la suite, nous présentons quelques unes des options les plus importantes.

L'option Window Scale Facor (WSOPT)

Définie dans RFC 1323
(http://www.networksorcery.com/enp/protocol/tcp/option003.htm)
Le champ WINDOW (16 bits) permet d'indiquer une taille de fenêtre de réception d'au maximum 64 KB. Si cette valeur était idoine pour les jours où les débits maxima étaient de 4Kbits par seconde, cela n'est plus vrai dans les réseaux modernes où les débits peuvent aller de 1Mbps à 1Gbps. Il est aisé de constater qu'avec un débit de 100 Mbps et un délai de 80 ms, la fenêtre de réception est égale à 100 .10^6.80.10^-3=8000*1000=8Mb=1MB. Comment exprimer cette valeur sur un champ de 16 bits ? Impossible. C'est là que rentre en jeu l'option WSOPT (définie sur 3 octets, dont le premier qui désigne son type vaut '3', le deuxième octet contient la taille de 3 et le dernier octet contient le facteur d'échelle lui-même). Ainsi si WSOPT est utilisée, la fenêtre de réception est calculée comme WINDOW* 2^(valeur de l'octet facteur

d'echelle). Ainsi 1M= 64KB*2^4. La valeur 4 est utilisée dans ce cas comme valeur de l'option WSOPT avec 'FF' comme valeur de WINDOW.

L'option TSOPT (Timestamp) RFC 1323 (type 8, taille 10 octets)

(ref http://www.networksorcery.com/enp/protocol/tcp/option008.htm)
Cette option permet de faire fonctionner deux mécanismes : RTTM (Round Trip Time Measurement et PAWS (Protect Against Wrapped Sequence). Le RTTT permet entre autres de fixer le Timeout associé à la retransmission des segments, et PAWS de se protéger contre le renouvellement d'un numéro de séquence si la taille du flux dépasse 4GB.

L'option TSOPT est sur 10 octets divisés en 4 parties : Kind (8bits) dont la valeur est 8, Length (8bits) dont la valeur est 10, TimestampValue (TsVal) (32 bits) et TimeStamp Echo Reply (TSecr) (32 bits). TSval contient l'estampille temporelle actuelle du TCP envoyant l'option. TSecr contient l'estampille temporelle reçue le plus récemment d'un TCP distant. Pour être un TSecr valide, le bit ACK doit être mis à 1. Et s'il n'est pas valide, TSecr doit contenir un zéro.

Un TCP peut envoyer un TSOPT s'il a reçu récemment une estampille ou s'il a reçu un TSOP durant la phase d'ouverture de connexion (SYN initial).

L'option SACK Permitted

L'option SACK Permitted (Selective Aknowldgment Permitted) (2 octets, Kind=4, Length=2) (http://www.networksorcery.com/enp/protocol/tcp/option004.htm) définie dans la RFC 2018 peut être utilisée dans un segment SYN (ouverture de connexion) pour signifier à l'autre partie de la connexion qu'on peut recevoir des segments avec l'option SACK (Kind = 5).

L'option SACK

Envoyé après l'ouverture de connexion, cette option de taille variable (kind=5) contient la liste de segments acquittés avec un seul segment. Donc contrairement au fonctionnement de base de TCP qui permet d'acquitter un seul segment, cette option permet d'acquitter plusieurs segments reçus. L'émetteur ne va donc pas retransmettre les segments dont les ACK sont en cours de transit et pour lesquels le timeout peut arriver bientôt. Car il suffit que le premier ACK arrive et l'option SACK donne la liste des segments pour lesquels le TCP récepteur ne va pas envoyer des ACK individuels.

3.4 Contrôle de congestion

Le contrôle de flux permet de définir une fenêtre de réception qui indique la quantité de donnée qu'un récepteur peut recevoir dans ses tampons et traiter par ses capacités de calcul disponibles, mais ne permet pas de s'adapter à la situation des nœuds intermédiaires du réseau (routeurs, chemins). Informellement parlant, l'objectif des algorithmes de contrôle de congestion est d'envoyer autant de segments de données que possible sans dépasser la fenêtre de réception du récepteur certes mais aussi tant qu'aucun segment n'est perdu (maximiser les performances de TCP). Cependant, si un segment venait à être perdu (ACK non arrivé après un RTT), l'algorithme suppose que le chemin est congestionné et réduit le nombre de segments en transit tant que la situation le nécessite. La première phase est appelée 'phase d'évitement' de congestion et la seconde 'phase de traitement' de la congestion. Les solutions adoptées pour la première phase et la seconde ont évolué afin de détecter au plus tôt la congestion et ensuite de réduire de façon aussi raisonnable que possible l'impact de la congestion dans la seconde phase. Les algorithmes 'Slow start', 'Reno' et 'Tahoe' indiquent l'évolution des techniques de contrôle de congestion. A noter que bien trois bits appelés ECN sont définis dans le champ 'Control bits' de l'en-tête TCP, ces bits ne sont pas actuellement utilisées et leur utilisation est réservée.

Avec TCP, chaque partie utilise une « fenêtre en émission ». La taille de cette fenêtre est min(awin, cwnd) où awin (advertised window) est la taille de la fenêtre que propose l'émetteur et cwnd (congestion window) est un entier qui évolue en fonction des pertes de paquets constatées. « cwnd » permet donc de traiter les problèmes de congestion (i.e. de surcharge) du réseau là où « window » permet de traiter les problèmes de surcharge du récepteur !

3.4.1 Fonctionnement du contrôle de congestion TCP

TCP utilise alternativement deux algorithmes « slow start » et « congestion avoidance ». Le passage d'un algorithme à l'autre est guidé par une valeur seuil : l'entier slow start threshold (ssthresh)

14. **Slow start**

    au démarrage : cwnd = 1 segment

    à chaque ACK reçu portant sur une valeur plus grande que les ACK précédemment

reçus : cwnd = cwnd +1

on transmet (envoie sur le réseau) la fenêtre (plusieurs segments) de taille min(awin, cwnd)

Notons qu'à chaque RTT (Round Trip Time), la taille de cwnd double (RTT 1 : 1 paquet + 1 ACK ; RTT 2 : 2 paquets + 2 ACK; RTT 3 : 4 paquets + 4 ACK; ...). Cette croissance est exponentielle.

15. **Congestion avoidance**

à chaque ACK reçu portant sur une valeur plus grande que les ACK précédemment reçus : cwnd = cwnd + 1/cwnd

on transmet la fenêtre de taille min(awin, cwnd)

-Initialisation : ssthresh = 65535

-Fonctionnement : if (cwnd < = ssthresh) cwnd += 1; else cwnd += 1/cwnd;

-Ce que fait TCP en cas de perte ; la perte est détectée lorsque un timeout associé à un segment expire sans que le ACK correspondant n'arrive : ssthresh=max(cwnd/2, 2); cwnd= 1

-Si trois (03) ACK identiques sont reçus, TCP n'attend pas l'expiration du timer et réémet les segments (Fast Retransmission).

-Selon les versions de TCP, le comportement diffère :

TCP tahoe : redémarrage à slow start (cwnd = 1)

TCP Reno : ssthresh = max(cwin/2, 2), cwnd = ssthresh, entre en mode « fast recovery »

TCP New Reno : voir plus bas

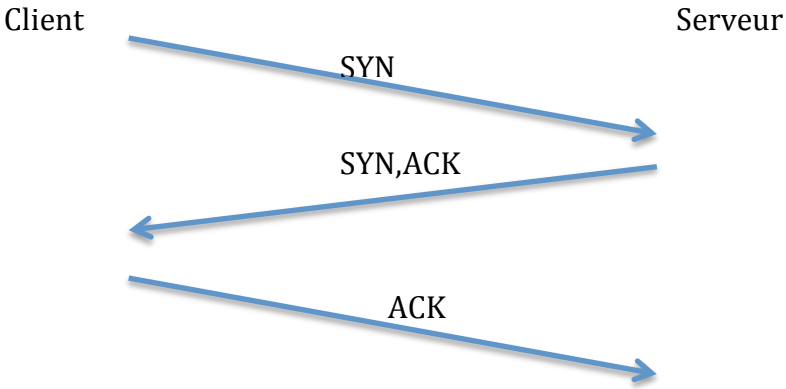
16. Mode « fast recovery » : réémission du 1er segment non acquitté (perdu) puis attente d'un ACK, si non réception, retour au mode « slow start. »

17. TCP New-Reno modifie le comportement de TCP Reno lorsqu'on reçoit un ACK "partiel" c'est-à-dire un ACK qui acquitte au moins le segment perdu (celui qui nous a fait entrer en Fast Recovery) mais pas tous les segments envoyés. Lors de la réception d'un tel ACK, TCP Reno quitte le mode de Fast Recovery ce qui pose un problème de performance lorsque plusieurs segments d'une même fenêtre sont perdus (les segments sont souvent perdus en rafale). TCP New-Reno ne quitte le mode de « Fast Recovery » que si l'ACK reçu acquitte tous les segments envoyés. A la réception d'un ACK partiel, New- Reno retransmet immédiatement le segment suivant le dernier segment acquitté dans cet ACK, diminue la taille de la fenêtre du nombre de segments acquittés par cet ACK partiel et retransmet un segment si c'est permis par la taille de cwnd.

3.5 Ouverture et Fermeture de connexion TCP

L'ouverture de connexion se fait avant de pouvoir envoyer les données. L'ouverture se base sur les flags SYN et ACK et se déroule en trois étapes (Three-way Handshake) comme décrit dans la figure suivante:

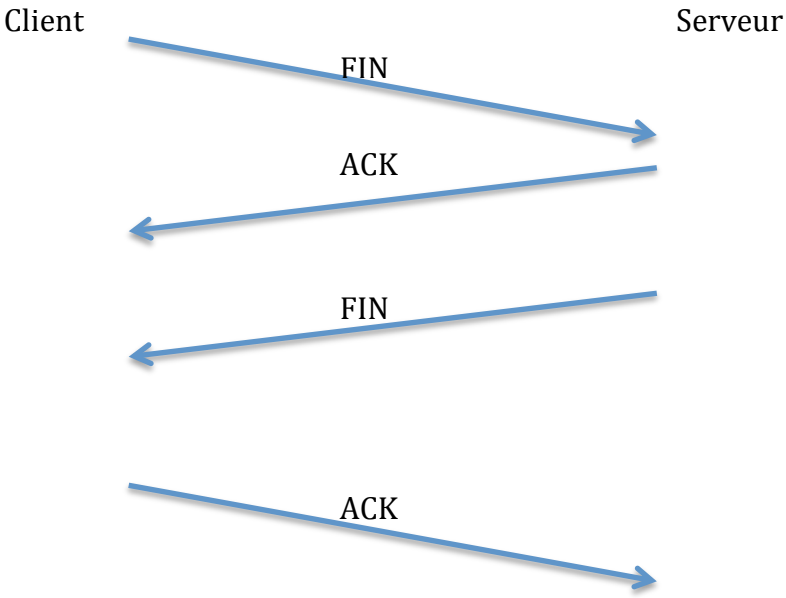
Le client commence par créer le contexte de communication appelé TCB (Transmission Control Bloc) relatif à la connexion en cours d'ouverture. Ensuite, il envoie un segment où le drapeau SYN est mis à 1, le champs numéro de séquence généré de façon aléatoire et une ou plusieurs options dans le champs options (voir options ci-dessus). Le serveur a déjà ouvert passivement une connexion (la fonction listen). Ceci lui permet de recevoir sur le port indiqué par lui, le segment envoyé par le client. A son tour il crée un TCB pour cette connexion, génère un champs numéro de séquence initial, remplit le champs Accusés de réception avec "numéro de séquence reçu +1" , met les bits ACK et SYN à 1, insère ses propres options TCP et calcule enfin le champs Checksum. Le segment résultat est envoyé au client. Le client, à la réception de ce segment, renvoie un nouveau segment ayant le champs ACK à 1, les options acceptées, un numéro de séquence augmenté de 1, et un "Accusé de réception" calculé comme par le serveur et considère désormais la connexion établie. A la réception de ce segment, le serveur de son côté considérera la connexion établie. L'envoi des données peut maintenant commencer.



Après que toutes les données aient été échangées, le client ou le serveur peut initier la fermeture de la connexion. Celle-ci se déroule en quatre phases.

L'application sur le client demande la fermeture de la connexion à l'entité TCP. L'entité TCP s'assure d'envoyer toutes les données dans le buffer dédié à l'application puis émet un segment ayant le flag FIN mis à 1. Le récepteur du segment FIN, renvoie un segment

ACK et indique à son application la demande de fermeture. Lorsque l'application serveur est prête à la fermeture, elle l'indique à son entité TCP qui envoie un segment FIN à son tour et attend le segment ACK correspondant. A sa réception, le serveur considère la connexion fermée et libère toutes les ressources qui lui étaient allouées. Le client lui attend le double du MSL (Maximum Segment Life, environs 2 minutes sur IP) avant de libérer les ressources.



4 Conclusion

TCP et UDP deux protocoles normalisés par l'IETF, répondent à des besoins différents des applications s'exécutant sur un réseau de données. Dans ce chapitre, nous avons examiné les fonctions les importantes de TCP. D'autres détails concernant par exemple l'estimation correcte du RTT (à la base du calcul du timeout associé à la retransmission) sont tout aussi importants et illustrent les développements continus apportés à TCP afin d'améliorer ses performances ou de répondre à de nouveaux besoins comme la sécurité ou la qualité de service.

Chapitre 3 : Le système DNS

1 Introduction

Il est très difficile pour un utilisateur de l'internet de se souvenir de beaucoup d'adresses IP associées à plusieurs machines. D'où l'idée d'associer un nom à chaque machine et de créer un mécanisme qui permet de trouver l'adresse IP correspondant au nom d'hôte. Ceci a donné naissance dans un premier temps à un mécanisme de résolution statique se basant sur un fichier (il existe toujours sous Unix/Linux et même sous Windows : /etc/hosts) qui contient un ensemble de lignes correspondant chacune à une paire (nom, adresse IP). Cette solution est bonne tant que l'administrateur d'un hôte Internet n'a pas décidé de changer l'adresse correspondant à un nom. S'il le fait, il doit prévenir d'abord tous les utilisateurs ayant cette adresse qui doivent à ce moment éditer manuellement leur fichier /etc/hosts pour changer l'adresse et répercuter le fichier sur toutes les machines qu'ils gèrent. Cette procédure est contraignante et longue d'où a émergé le besoin d'une solution dynamique qui ne se base pas sur le fichier /etc/hosts. Au lieu de cela, dans chaque machine, un résolveur contacte un serveur mandataire (spécifié par une ligne dans /etc/resolv.conf) et lui passe une requête de résolution.

Le serveur mandataire fait la résolution et rend la réponse au client d'où est partie la requête initiale.

Le système DNS (Domain Name System) centralise la gestion des noms sur Internet; il est géré par l'ICANN (pour une partie) et par les organisations elles-mêmes pour les serveurs de domaines privés.

2 Nommage dans l'Internet

Chaque machine sur internet possède un nom d'hôte (ex. www, mail, elearning….) et un nom de domaine (ex. univ-bejaia.dz, gmail.com, univ-bouira.dz, ….etc). La concaténation des deux fournit ce qui appelé le FQDN (Fully Qualified Domain Name) d'un hôte qui l'identifie complètement sur internet (par exemple, www.univ-bejaia.dz est le FQDN de la machine ayant un nom d'hôte « www » et un nom de domaine « univ-bejaia.dz».)

Tous les noms sur Internet sont organisés sous une forme hiérarchique (un arbre avec racine). Le domaine racine appelé point (.) n'est jamais mentionné explicitement mais existe en tant que « domaine racine ». Il existe 13 serveurs DNS dans le monde qui ont autorité sur le domaine racine, et ceux-ci sont administrés par l'ICANN. Au second niveau de cette hiérarchie figure ce qu'on a appelle les TLD (Top Level Domain) ou domaines de niveau supérieur qui sont eux-mêmes de deux types : TLD génériques (gTLD) comme .com, .org, .net, .edu, .mil, .biz, .museum et TLD de pays « country code TLD »(ccTLD) comme .dz, .fr, .us, .ca, .uk, .dk, etc.

3 Architecture du système DNS

L'espace de nom hiérarchisé est distribué sur des serveurs. Un serveur est dit avoir autorité sur un domaine s'il contient dans sa configuration une ligne de type SOA (Start of Authority). Un serveur ayant autorité sur le domaine racine est dit serveur racine. Chaque serveur de niveau n dans la hiérarchie a dans sa configuration une ligne de type NS indiquant l'adresse IP d'un serveur de niveau n+1 ayant autorité sur chaque domaine de niveau n+1. Ainsi les serveurs racines ont des lignes sous la forme "com. IN NS #IP1 org. IN NS @IP2, etc dans leur base de donnée. A la feuille de la hiérarchie de nom, on trouve le serveur DNS ayant autorité directe sur le domaine d'un hôte internet. Ce serveur contient en plus d'une ligne de type SOA, des lignes de type A (adresse IPv4) de type AAAA (adresse IPv6) ou MX (serveur de messagerie). Ces types sont appelés type de ressource et chaque ligne de configuration est d'un type RR (Ressource Record) donné.

Les hôtes d'un réseau local sont configurés statiquement ou par DHCP pour leur indiquer un serveur DNS du même réseau qui s'occupe d'effectuer des requête vers le système DNS afin de trouver l'adresse IP correspondante à un nom donné. Ce serveur local est appelé "serveur mandataire". En plus de faire les résolutions, il gère aussi un cache des résolutions effectuées afin d'optimiser l'utilisation de la bande passante ascendante et de répondre rapidement aux requêtes des clients du réseau local.

Les clients du réseau local, le serveur mandataire, les serveurs racines et ou serveurs ayant autorité sur leur domaine utilisent le protocole DNS (UDP, port 53) pour les requêtes/réponses DNS. Une approche basée sur TCP et le port 53 est utilisée aussi entre serveur DNS maître et esclave pour la réplication des données.

4 Résolution DNS

On appelle résolution DNS, le processus lancé par un serveur DNS mandataire ou n'importe quel client DNS (comme dig, host ou nslookup) afin de retrouver la valeur d'une ressource DNS dans le système DNS (hiérarchie de nommage et serveurs).
Pour y arriver, le serveur mandataire peut procéder de deux façons différentes: itérative ou récursive. Dans la suite, on suppose que la requête porte sur une ressource de type A (on cherche une adresse IP) pour le nom "www.domaine.tld."

4.1 Résolution itérative

Le serveur mandataire "A" demande à un serveur racine "B" de lui donner un serveur responsable sur "tld". Le serveur B répond à A en lui donnant la liste des serveurs ayant autorité sur "tld", car il les a dans sa base de données sous forme "tld IN NS @IP1" "tld IN NS @IP2". A la réception de la réponse, le mandataire A interroge le serveur "C" ayant l'adresse @IP1. « A » demande à « C » de lui fournir la liste des serveurs DNS ayant autorité sur "domaine". Comme « C » connait cette liste à travers ses enregistrements de la forme "domaine.tld. IN NS @IP3", « C » répond à « A » et fournit l'adresse @IP3. Maintenant, le serveur « A » envoie une requête pour le serveur « D » ayant l'adresse @IP3 et lui demande de lui fournir la réponse pour une requête de type A sur le nom www.domaine.tld.. Le serveur « D » répond à « A » et lui communique l'adresse IP de la machine ayant ce nom car elle figure dans sa configuration sous la forme d'une ligne www.domaine.tld. IN A @IP4 car le serveur « D » a autorité sur le domaine "domaine.tld" (une ligne de type SOA dans sa configuration indique cette autorité). A la réception de la réponse de « D », le serveur mandataire répond au client qui lui a adressé la requête. Le serveur mandataire enregistre aussi la réponse dans son cache pour une éventuelle requête d'un autre client du réseau local qui sera ainsi servi plus rapidement à partir du cache. Le client peut maintenant ouvrir une connexion TCP ou UDP en utilisant l'adresse IP au lieu du nom que l'utilisateur a fourni. Notez que dans ce processus, les requêtes envoyées par le mandataire « A » jusqu'à trouver l'adresse IP de « D » sont de type 'NS', qui indique des ressources de types serveurs de noms. Par contre la requête adressée à « D » lui-même est la même reçue du client initial.

4.2 Résolution recursive

Dans la résolution récursive, les informations « utilisées » sont les mêmes que celles échangées dans une résolution itérative mais les communications n'ont pas lieu de façon itérative. Ainsi, le serveur racine « B » ne répond pas au serveur mandataire mais interroge lui-même le serveur "C" car il connaît son adresse IP. Le serveur « C » interroge lui-même le serveur « D » et ainsi de suite. Lorsque le serveur ayant autorité sur le domaine recherché est atteint, celui-ci répond à celui qui l'a interrogé, et la réponse est relayée dans le chemin inverse des requêtes. Ainsi, « D » répond à « C » qui relaie la réponse à « B » qui répond au serveur mandataire « A ». Finalement « A » répond au client initial qui l'a contacté et met à jour son cache. Notez aussi que la requête du client est envoyée par A pour B qui la relaie pour C et C qui la relaie pour D.

5 Conclusion

Le système DNS est une approche dynamique permettant de distribuer le problème de nommage et de sa résolution entre hôte du réseau Internet. Un réseau local ne peut fonctionner sans au moins un serveur mandataire. Une organisation qui gère un ensemble de serveurs publics doit gérer au moins un serveur DNS ayant autorité sur son domaine afin de permettre aux utilisateurs de ses services internet d'user de noms au lieu des adresse IP.

Chapitre 4: DHCP

1 Introduction

A l'origine, dans les réseaux locaux ou internet de petite taille; on pouvait assigner une adresse IP et un masque, une passerelle, et un serveur mandataire à chaque machine manuellement.

Le fichier /etc/network/interfaces (debian) ou similaire contient une configuration pour chaque interface.

Aujourd'hui, les réseaux locaux peuvent contenir des milliers de machines, un fournisseur d'accès internet doit assigner une configuration à chaque hôte qui se connecte à son réseau…Dans ces cas, on a besoin de pouvoir configurer dynamiquement (lorsque la machine se connecte) chaque hôte. La configuration est valable durant une durée (bail), après expiration de laquelle, chaque hôte doit redemander une autre configuration. D'où le protocole DCHP (Dyanmic Host Control Protocol). DHCP utilise UDP (port 67, 68). Avant DHCP, dans le monde UNIX existait un protocole qui servait à booter (en chargeant un noyau) des terminaux sans disque (diskless terminals). Le protocole DHCP a remplacé BOOTP et a repris les fonctionnalités qu'il offrait.

2 Avantages de DHCP

DHCP permet :

- d'allouer des adresses IP aux machines, à la demande, lorsqu'elles se connectent au réseau (un ordinateur de bureau est allumé, un terminal mobile rejoint un réseau WIFI, un modem ADSL est connecté au réseau d'un FAI…)
- d'avoir une gestion centralisée des adresses IP.
- d'éliminer, pour l'administrateur réseau, la corvée de la configuration des nouvelles machines.
- Le nombre d'adresses IP disponibles peut être inferieur au nombre de machines du réseau.

3 Configuration d'un serveur DHCP

Le serveur DHCP écoute avec une socket UDP sur le port 67. A la réception d'une requête, il répond en plusieurs étapes à un client en se basant sur sa configuration. Celle-ci contient en général au moins les éléments suivants:

A- une table (un ou plusieurs intervalles d'adresses IP) valides et une table des adresses IP réservées qui peuvent être attribuées dynamiquement à une machine en fonction d'un nom ou adresse MAC ou assignée de façon manuelle (cas d'un serveur).

B- Des paramètres de configuration valides pour tous les clients du réseau (Masques, adresse de la passerelle, serveur DNS mandataire, proxy, etc…)

C- La durée du bail: la période de temps durant laquelle le client peut utiliser l'adresse attribuée.

4 Processus d'attribution

Le processus DHCP se déroule en quatre (04) phases (stages):

- Découverte (DISCOVER): Le client envoie une demande de configuration sur le réseau en diffusion, plusieurs serveurs DHCP peuvent être en écoute et donc recevoir la demande.
- Offre (OFFER) :Tous les serveurs DHCP répondent au client en lui faisant une offre,
- Demande (REQUEST): Le client répond à un serveur parmi ceux qui ont offert en lui précisant qu'il accepte l'offre proposée.
- Accusé de réception(ACK): Le serveur DHCP confirme le bail avec sa durée et les options DHCP associées, et met à jour sa table des adresses IP allouées.

Un client peut cesser volontairement d'utiliser une adresse IP (par exemple lors de l'extinction de l'ordinateur). Pour ce faire il envoie un message de type DHCPRELEASE.

Si d'autre part, un client se souvient d'une configuration récente (lors d'un démarrage ou redémarrage par exemple), il peut omettre certaines phases citées ci-dessus. Le protocole stipule que dans ce cas le client envoie un message DHCPREQUEST contenant l'option 'Requested IP Address' rempli à l'adresse qu'il a obtenue auparavant. Le serveur

répond avec un message DHCPACK. Les autres types de messages ne sont pas utilisés dans ce cas (DHCPDISCOVER et DHCPOFFER).

5 Options DHCP

En plus de l'adresse IP proprement dite qui est l'objectif premier de DHCP, celui-ci permet aussi de gérer de façon centralisée toute la configuration réseau d'un terminal TCP/IP. A la phase ACK, le serveur DHCP rajoute d'autres paramètres au client. Ces paramètres sont répertoriés dans la RFC 2132 . A titre d'exemple nous avons :

-domain-name : nom de domaine DNS auquel appartient la machine

-domain-name-servers : liste de serveur DNS à interroger pour effectuer des résolution DNS.

-subnet-mask : masque sous-réseau

-broadcast-address : adresse de diffusion générale

-gateway-address : adresse de la passerelle

6 Conclusion

Le protocole DHCP permet à un client TCP/IP d'obtenir à la base une adresse IP valide durant une période de bail. D'autres options peuvent y être rajoutés par le server DHCP. L'intérêt d'un tel service est la possibilité, pour l'administrateur réseau, de gérer de façon centralisée la configuration de tous les terminaux réseau.

Chapitre 5: Service de messagerie Internet et Protocoles associés

1 Introduction

La messagerie électronique est un moyen de communication différé ou asynchrone entre utilisateurs d'un réseau informatique. Différé car contrairement à la messagerie instantanée (le chat), le récepteur du message n'est pas sensé répondre immédiatement à l'émetteur et vice-versa. La messagerie électronique a été le second service après l'émulation de terminal (telnet) et le FTP a connaitre un développement avant même le service web. Aujourd'hui encore ce service est très utilisé par les entreprises et les particuliers. L'objectif de ce chapitre est de présenter l'architecture du service de messagerie électronique et ses protocoles.

2 Architecture du service de messagerie

Le service de messagerie est constitué de trois entités distinctes qui coopèrent et communiquent par le biais de protocoles bien défini afin d'assurer un service entre utilisateurs.

1. Les MTA (Mail Transfer Agent) ou ce qui est communément un serveur de messagerie sortant. C'est lui qui s'occupe de livrer un message envoyé par un utilisateur donné au récepteur. Comme exemple de logiciels Assurant ce rôle : Sendmail (l'un des plus utilisées), Postfix (promis à remplacer Sendmail) et Exchange (Microsoft).
2. Les MUA (Mail User Agent) servent pour l'expéditeur et le destinataire est un logiciel client pour le MTA (lors d'un envoi) et un client pour le MDA (lors d'une réception). Comme exemple, nous avons Outlook (MS) et Thunderbird (Mozilla), Evolution… Avec l'apparition du web, des clients web légers aussi sont utilisées comme par exemple "roundcube", "horde" (libres et installable) ou ceux qu'on trouve sur les hébergeurs gratuits de courrier comme Yahoo, Hotmail et Gmail.
3. Les MDA (Mail Delivery Agent) : servent à récupérer le courrier par un lecteur de courrier en ligne ou hors ligne. Ce serveur est aussi appelé Serveur de courrier

entrant. Comme exemple de MDA, nous avons Dovecot, maildrop-Courrier, Cyrus IMAP (IMAP), et Procmail (livraison dans $HOME/Maildir/).

3 Envoi et réception d'un message

Nous mettons dans le cas d'un exemple d'envoi d'un message d'un émetteur ayant l'adresse user1@domain1.tld. Vers un autre utilisateur user2@dmaoin2.tld.

- L'expéditeur utilise son logiciel de courrier électronique (lourd, léger ou mobile) pour envoyer un courrier électronique qui n'est qu'un simple ficher texte ayant une enveloppe (expéditeur, récepteur, adresse de retour, sujet du message, corps du message) vers son serveur de courrier sortant. Le MUA utilise pour cela soit le protocole SMTP (Simple Mail Transfer Protocol) soit IMAP (Internet Mail Application Protocol). Le SMTP est un protocole de niveau application utilisant TCP et le port 25. IMAP est aussi en TCP mais utilise le port.

Le serveur de messagerie sortant de l'utilisateur user1, à la réception du message, extraie le domaine domain2.tld de l'adresse du destinataire et effectue une recherché DNS de type MX pour chercher l'adresse du serveur de messagerie du domaine domain2.tld. Une fois trouvée l'adresse IP de ce serveur, il ouvre une connexion avec lui et lui transfère le message de user1 et ferme la connexion.

- Le serveur de messagerie du destinataire "user2", délivre le message vers le MDA du domaine domain2 avec le protocole LMTP (Lightwight Message Transfer Protocol). Souvent le MDA est un logiciel sur la même machine que le MTA. Le MDA met le message reçu dans la boite de l'utilisateur user2.
- L'utilisateur destinataire utilise son MUA pour récupérer auprès de son MDA, ses messages entrants. L'échange se fait soit en POP3 (Post Office Protocole), soit en IMAP)

4 Protocole SMTP et POP3

Les protocoles SMTP et POP3 sont basés sur TCP (port 25 et 110 respectivement). Dans la suite, nous montrons comment envoyer un courrier avec les commandes SMTP et comment utiliser POP en ligne de commande.

4.1.1 Exemple SMTP

Nous nous connectons à un serveur SMTP

Neptune :absider$ telnet mail 25

Trying X.X.1.13...

Connected to mail.univ-bejaia.dz.

Escape character is '^]'.

220 mx1.univ-bejaia.dz ESMTP Postfix

HELO neptune

250 mx1.univ-bejaia.dz

MAIL FROM: <ar.sider@univ-bejaia.dz>

250 2.1.0 Ok

RCPT TO: <agent.externe@gmail.com>

554 5.7.1 <agent.externe@gmail.com>: Relay access denied

RCPT TO: <ar.sider@univ-bejaia.dz>

250 2.1.5 Ok

DATA

354 End data with <CR><LF>.<CR><LF>

Hello from me to me. Ceci est un mail envoyé avec telnet en tapant des commandes smtp.

.

250 2.0.0 Ok: queued as 3599B119EF3

quit

221 2.0.0 Bye

Connection closed by foreign host.

L'échange peut comprendre (en fait il est nécessaire souvent) une phase d'authentification si le message est destiné à un utilisateur d'un autre domaine (commande AUTH). En effet, il faut noter ici que RCPT TO :<@gmail.com> a échoué car le serveur ne permet pas d'envoyer à un autre domaine si on n'est pas encore authentifié.

Le message proprement dit es t compris commence aprés DATA et est terminé comme il est dit pas le serveur avec <CR><LF>.<CR><LF> (retour chariot, point , retour chariot).

4.1.2 Exemple POP

Dans ce cas on se connecte pour voir nos messages.

Neptune: absider$ telnet mail 110

Trying X.X.1.13...

Connected to mail.univ-bejaia.dz.

Escape character is '^]'.

+OK mail.univ-bejaia.dz POP3 server ready

USER ar.sider

+OK hello ar.sider, please enter your password

PASS xxxxxxx

+OK server ready

STAT

+OK 919 55343088

RETR 1

+**OK** message follows

Received: from mail.univ-bejaia.dz (LHLO mx1.univ-bejaia.dz) (X.X.1.13)

by mail.univ-bejaia.dz with **LMTP**; Sun, 17 Feb 2015 21:18:23 +0100 (CET)

Hello from me to me. Ceci est un mail envoyé avec telnet en tapant des commandes smtp.

quit

+**OK** mail.univ-bejaia.dz POP3 server closing connection

Dans l'échange POP, on voit qu'on ne peut accéder à sa boite sans authentification (lignes USER puis PASS). Ensuite, la commande STAT permet de savoir quel est le nombre total de messages (ici 919) et leur taille (en octet). La commande RETR avec pour paramètre le numéro d'un message (entre 1 et 919) permet d'obtenir la totalité d'un message. La commande TOP x y, elle, permet d'afficher y lignes du message numéro x. Enfin, la commande DELE n permet de supprimer le message numéro n. Notons comment le MDA interrogé a reçu ce message (**LMTP**).

N.B : Vous pouvez tester ces commandes avec votre compte Gmail ou Yahoo. Mais vous devez d'abord permettre l'accès POP3 (Paramètres du comptes).

5 Conclusion

Dans ce chapitre, nous avons présenté l'architecture des systèmes de messagerie et ses protocoles. La messagerie reste avec le web l'un des services les plus utilisés de l'internet.

Série de TD N°1

Exercice 1 :

On considère un échange sur un réseau local Ethernet. L'en-tête Ethernet mesure 18 octets. Les données issues de la couche application mesurent 500 octets. Elles sont transmises au protocole de la couche transport :UDP. L'en-tête UDP mesure 8 octets. Le protocole utilisé pour la couche réseau est IP : son en-tête mesure 20 octets.

1. Représentez sur un schéma la pile de protocoles utilisés dans cet échange. Indiquez pour chaque couche la constitution des PDU en précisant la taille des PCI et SDU.
2. Calculez la taille en octets d'une trame Ethernet émise sur le réseau local.
3. Sachant que le débit du réseau Ethernet est 10 Mbit/s, quelle est la durée de transmission d'une trame sur le réseau?
4. Quel est le pourcentage de la bande passante occupé par les en-têtes ?
5. Quelle serait la durée de transmission de la trame en l'absence des en-têtes ?
6. Calculez le débit utile (taille des données utiles/durée de transmission des données)

Exercice 2:

Dans cet exercice, vous allez étudier la RFC 1939.

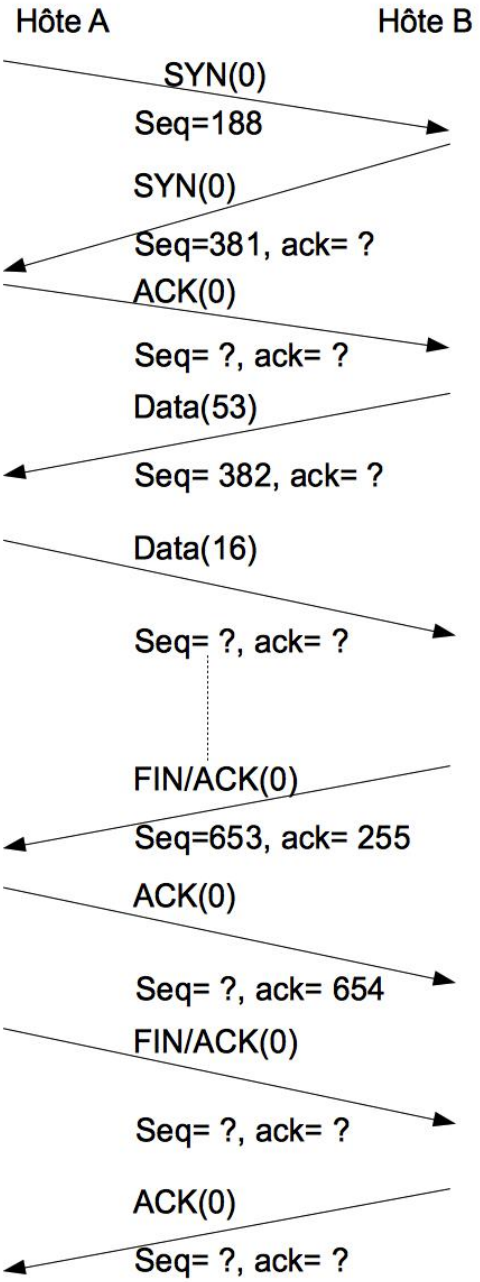
- A. À quel protocole se rapporte la RFC 1939 ? Quelle est sa date de publication ? Quelle est sa catégorie ? Que signifie la notation STD:53 dans l'en-tête de la RFC? Quelles sont les RFC qui ont mis à jour la RFC 1939 ? Quelle RFC a été rendue obsolète par la RFC 1939 ?

B. Consultez la RFC pour répondre aux questions suivantes. Au dessus de quel protocole de niveau transport fonctionne le protocole POP3? Sur quel port ? Comment se termine un message transmis par POP3 ? Quelle est la fonction de la commande TOP ? Quels sont ses arguments ? Son implémentation est-t-elle obligatoire ?

Série de TD N°2

Exercice 1 :

La figure suivante correspond à un échange FTP. Il est donné pour chaque segment, sa nature (en majuscule au-dessus de la flèche), le nombre d'octets transmis (entre parenthèse) et les numéros de séquence (Seq) et d'acquittement (Ack). Complétez la figure. Que pouvez-vous conclure par rapport à un Ack envoyé et au prochain Seq reçu ?



Exercice 2 :

Indiquez les assertions correctes.

A. Sur un réseau Ethernet utilisant TCP/IP, le contrôle de flux se réalise au niveau
    a. Physique
    b. MAC
    c. IP
    d. TCP
B. Quelles affirmations sont correctes?
    a. UDP est plus rapide que TCP
    b. TCP est plus fiable que UDP
    c. UDP utilise les numéros de port
    d. TCP utilise des numéros de port
C. Une connexion TCP est établie entre
    a. La passerelle source et la passerelle destination
    b. Le PC source et le PC destination
    c. Le PC source et le routeur d'accès internet
D. Que deviennent les segments TCP non reçus en cas de congestion sur Internet?
    a. Ils sont définitivement perdus
    b. Ils sont retransmis par le réseau
    c. Ils sont retransmis par l'émetteur
E. Un émetteur qui vient de recevoir un segment TCP avec une valeur du champ 'window' de 2048 peut envoyer :
    a. 2048 segments de suite
    b. Un segment de 1024 octets
    c. Deux segments de 1024 chacun
    d. Un segmente de 2048 octets

Exercice 3 :

-Le paramètre du champs TCP "fenêtre" est calculé de la façon suivante : Fenêtre = Débit x RTT où RTT est le Time To Return (Temps d'aller retour).
Pour chacune des valeurs ci-dessous du RTT calculer le champ window d'une connexion lorsque le débit est de 10Mbps, 100Mbps, 1000Mbps et 10Gbps:
10ms, 100ms, 200ms, 500ms, 1s.
-

Série de TD N°3

Exercice 1
Pour chacune des affirmations ci-dessous, cochez la ou les réponses correctes. Lorsque c'est possible corrigez l'énoncé erroné.

1- Les serveurs DNS permettent
- a- d'associer à un nom de domaine une adresse IP
- b- d'associer à un nom de machine une adresse IP
- c- à un utilisateur de l'internet d'utiliser directement l'adresse IP
- d- de garder en mémoire les pages fréquemment consultées par les internautes

2- Lors d'une requête DNS, quel serveur connaît forcément l'adresse IP de « eccp.poste.dz » ?
- a- le serveur faisant autorité sur "dz"
- b- le serveur DNS faisant autorité sur "poste.dz"
- c- le serveur faisant autorité sur "eccp.poste.dz"
- d- le serveur faisant autorité sur tous les postes de l'internet

3-Dans une résolution DNS récursive
- a- le serveur DNS mandataire connaît au moins l'adresse d'un serveur racine
- b- chaque serveur interrogé renvoie au serveur mandataire l'adresse du serveur suivant
- c- le serveur DNS mandataire relaie toutes les requêtes
- d- l'adresse IP recherché est envoyée directement au DNS primaire

4-Concernant un enregistrement DNS
- a- un RR de type CNAME donne le nom du serveur faisant autorité sur la zone pointée
- b- un RR de type PTR donne le nom du serveur DNS pour le domaine pointé
- c- un RR de type A donne le nom de domaine correspondant à l'adresse recherchée
- d- un enregistrement de type NS donne l'adresse IP correspondant au nom de domaine recherché.

5- Le système DNS est géré par :
- a- l'ICANN
- b- L'ICANN, les registrar, les FAI et organisations connectées à Internet
- c- L'utilisateur final
- d- L'ISO

6-Le protocole DNS utilise:
- a- UDP et le port 53
- b- TCP et le port 53
- c- UDP, TCP et le port 53

Exercice 2: Pour la suite, dessinez d'abord l'arbre du nom, et supposez les noms (A,B,C…) et adresses IP des serveurs ayant autorité sur chaque (sous-)domaine.
1- Donnez la liste des requêtes, leurs source et destination, type et nom de ressource qu'un serveur mandataire « M » enverra et recevra pour résoudre récursivement une requête du genre «Adresse IPV4 » pour « www.cs.toronto.edu »
2- Même question en supposant une requête itérative.

TP N°1

Modèle réseaux

Exercice **1 :**

Dans cet exercice, on s'intéresse aux organisations de normalisation internationale.

A. Visitez le site www.itu.int. Lisez l'historique, les sections membres et retrouvez la définition de UIT (ITU) ? Qui peut en être membre ? Quand est-ce qu'elle est née? Quels sont les membres de droit algérien ? Elle s'appelait comment avant ? Quel est son domaine de compétence ? Citez quelques normes de l'ITU.

Visitez le site www.iso.org. Donnez la définition de l'ISO (Section à propos de l'ISO) qui en est membre ? Qui sont les membres algériens de l'ISO ? Quand est-ce qu'elle est née? Quels sont ses domaines de compétences ? Retrouvez sur le site quelques normes de l'ISO.

C. Retrouvez le site officiel de l'IEEE.C. Qu'est-ce que l'IEEE ? Qui en est membre ? Quand est-ce qu'elle est née ? Quels sont ses domaines de compétences ? Citez quelques normes de l'IEEE.

Exercice 2:

A. Accéder à www.ietf.org. Que signifie IETF ?

B. Accéder à www.ietf.org/rfc.html. Que signifie RFC ? Cherchez la RFC 1939. A quel protocole se rapporte la RFC 1939 (titre)? Quelle est sa date de publication ? Quels en sont les auteurs ? Quelle est sa catégorie ? Que signifie la notation STD:53 dans l'en-tête de la RFC? Quelles sont les RFC qui ont mis à jour la RFC 1939 ? Quelle RFC a été rendue obsolète par la RFC 1939 ?

C. Consultez la RFC pour répondre aux questions suivantes. Au dessus de quel protocole de niveau transport fonctionne le protocole POP3? Sur quel port ? Comment se termine un message transmis par POP3 ? Quelle est la fonction de la commande TOP ? Quels sont ses arguments ? Son implémentation est-t-elle obligatoire ?

TP N°2

TCP/UDP

Outils : wireshark , wget

Procédure :

1. Télécharger la trace (la capture) de l'adresse (http://packetlife.net/captures/HTTP.cap)
2. Ouvrir Wireshark, Aller dans File->Open puis choisir le fichier téléchargé en 1.
3. Examiner le fichier ouvert et répondre aux questions suivantes

Questions :

1. Enumérer les colonnes du tableau du centre, quels est le sens de chacune ?
2. Positionner le curseur sur la ligne N°1 et examiner le contenu du tableau en dessous. Enumérer la liste des protocole que vous reconnaissez. Quel est le numéro du protocole IP dans une trame Ethernet ?
3. Dans un tableau approprié indiquer pour les trois premières frames (1,2,3) ,les valeurs des champs suivants : Champs @IP-Source @IP-Destintation, Port-Source, Port-Destination, Flags, champs numéro de séquence, champs numéro acquittement. Quelle l'est l'adresse IP du client dans ce dialogue ? Et celle du serveur ? Comment avez-vous fait pour les identifier ? Quels sont les programmes qui sont en communication C/S ? Comment sont-ils identifiés ?
4. La ligne 4 concerne quel protocole de niveau application ? Quel type de message est envoyé ? Quelle version du protocole de niveau application est utilisée ? Combien d'octets contient-t-elle ?
5. Continuez à donner les valeurs des champs numéro séquence, numéro acquittement et flag pour le restant des segments. Quelles sont les règles de leur évolution ?
6. Analyser et discuter de la ligne 36. Quelle est la taille de l'image ? Combien de segments TCP a nécessité son envoi ?
7. A quoi est dû la ligne 37 ?
8. Comme pour la question 1, refaites le tableau pour les lignes 38-40. Que pouvez-vous en dire ?
9. Faites un graphe montrant l'évolution de la fenêtre d'émission du serveur vers le client. Discutez-là.
10. Re-éxaminer maintenant le champs Options TCP des trois premières lignes . Les lignes à partir de 4 les contient-t-elles toutes ? Qu'en concluez-vous ? Quelle(s) RFC(s) décrit chaque option ? Pour chaque option indiquer son utilité (contrôle de congestion, contrôle de flux, connexion fiable,etc...).

Exercice 2 : Télécharger la trace (http://packetlife.net/captures/RIPv2.cap)

Questions :

1. Quel adresse destination est-t-elle utilisée ? Quelle est sa classe ?
2. Quel protocole de transport/port source/port destination est utilisé ici ? Quelle la longueur du segment de transport, son checksum ?
3. Quels réseaux annoncent t-il ce routeur ?
4. Comparez avec la session de l'exercice 1.

Série de TP N°3

DNS

La commande dig (Domain Information Groper) est un programme qui permet d'inetrroger le système DNS. Le format général de la commande est comme suit :

$dig TypeDeRessource NomDeRessource [@Adresse IP Serveur] [+trace] [+[no]recurse] où

TypeDeRessource : mx, ns, A, AAAA, -x (résolution inverse) etc...

NomDeRessource : fqdn, nom de domaine

Adresse IP Serveur : Adresse d'un serveur DNS à utiliser

+(no)recurse : requête (itérative) récursive

+trace : permet d'afficher les échanges entre le client dig et chacun des serveurs impliqués

Exercice 1 :

1-Afficher le contenu de /etc/hosts, quel est son contenu ? Le format de chaque ligne ? Expliquez chaque ligne

2-Afficher le contenu de /etc/hostname, est-ce un fqdn ? Un nom d'hôte ? Utilisez le programme hostname pour avoir le nom de la machine, son fqdn et son domaine (option -s, -f, -d).

3-Afficher le contenu de /etc/resolv.con. Quel est son format ?

4-Lancez dig A www.google.dz

4-1 Quelle est la version de dig ?

4-2 Quelles sont les différentes sections affichées ? Quel est le format de chaque enregistrement ?

4-3 Quel serveur a répondu ?

4-4 Quelles sont les différentes options DNS utilisées et disponibles ? Que signifie « ra » et « rd » ?

4-5 Quelle est donc le serveur DNS mandataire de votre machine ?

Exercice 2 :

En vous servant de l'outil dig, retrouvez les informations suivantes :

1-La liste des « serveurs racine »

2-L'adresse IP de chaque serveur racine trouvé en 1

3-L'adresse IPv6 de chaque serveur en 1

4-La liste des serveurs ayant autorité sur le domaine 'com. ', 'fr.' et 'dz.'

5-La liste des serveurs ayant autorité sur les domaines « google.com » «la.com », et « klothnet.com »

6-Faites la résolution inverse pour 193.194.94.2, 8.8.8.8

TP N°4

DHCP

Outils utilisés : linux, tcpdump, wireshark,

Objectifs : Etude avancée du protocole DHCP

Exercice 1 (Rappels et commandes réseau Linux de base):

1. A quoi sert une adresse IP ? Comment la retrouver sur un système Linux ?
2. A quoi sert le masque associé à une adresse IP ? Comment le retrouver sur uns système Linux ?
3. A quoi sert la passerelle IP associée à une interface ? Comment la retrouver sur un système Linux ?
4. Décrivez le Processus ARP pour une remise directe et une remise indirecte. Afficher le « cache arp » de votre machine.
5. De quelle autre information un hôte IP a-t-il besoin si il est utilisé par un être humain ?

Exercice 2 (DHCP):

1. Sachant que le protocole DHCP s'appelle « bootp » sous Linux, chercher le port utilisé dans le fichier /etc/services avec la commande nécessaire. Ensuite, réalisez une capture de trafic DHCP avec tcpdump. Quel est le filtre approprié ?
2. Ouvrir le fichier de capture réalisé en 1. Quel protocole de transport est utilisé par DHCP ? Pourquoi à votre avis, on n'a pas eu besoin de générer du trafic DHCP comme on l'a fait pour le trafic DNS ?
3. Enumérez la liste des types de messages utilisés par DHCP.
4. Pour chaque message de type donné en 3, énumérez, l'adresse IP Source et l'adresse IP Destination
5. Même question que 4, mais énumérez les adresses mac source et mac destination
6. Quelles sont les valeurs du champ option de DHCP pour chaque type de message ? Comment s'appelle l'option de passerelle, son numéro ?
7. Quelle est la durée du bail de la configuration offerte? Que veut dire l'option « Rebinding Time » ?
8. Comment s'appelle l'option indiquant les serveurs DNS mandataires ? Quel est son contenu ?

Remarque : les messages DHCP capturés seront multiples pour chaque type de message. Aussi, vous pouvez restreindre l'affichage dans Wireshark uniquement lié à une certaine @mac source/destination

TP N°5

Messagerie électronique

Objectif : Etude des types SoA et MX, enveloppe de message électronique, MIME et SMIME, SPF

Outils : dig ; webmail yahoo et gmail.com

Exercice 1 :

1. Pour chacun des domaines suivants, identifiez l'administrateur du domaine DNS, le(s) serveurs de messagerie entrant, le serveur de courrier sortant : gmail.com, yahoo.com, cerist.dz.
2. Quelle est la forme de l'enregistrement de type SoA ?
3. Quelle est la forme de l'enregistrement de type MX ?
4. effectuez la résolution inverse pour les MTA et les MDA. Que constatez-vous ?

Exercice 2 :

En affichant le contenu d'un message dans votre boite yahoo ou gmail,

1. Listez la chaine des MDA qui ont servi à la réception du message (chaine Received from X by Y)
2. Identifiez le MTA qui a, le premier, envoyé le message
3. Identifiez le MDA qui a reçu, le premier, le message entrant vers votre boite.
4. Identifiez les en-têtes de l'enveloppe du message
5. A quoi sert l'en-tête nommé Dkim Signature
6. Que signifie l'en-tête portant sur SPF ?
7. Etudiez la norme MIME S/MIME et donnez les paramètres utilisés dans ce message

Textes de référence du cours

1. Stéphane Lohier, Aurélie Quidelleur, « Le réseau Internet, Des services aux infrastructures », Dunod, 2010.
2. Douglas Comer, « TCP/IP Architecture, protocoles et applications », 3ème édition. Dunod, 2000.
3. Robin Burk, Martin Bligh, Thomas Lee et al., « TCP/IP Blueprints ». SAMS Pblishing, 1997.
4. Andrew Tanenbaum, David Wetherall, « Réseaux, 5ème édition. » PEARSON Education/Prentice Hall, 2011.

Nota : Ceci est un support de cours et ne remplace pas le cours lui-même. Pour vos remarques et suggestions ou pour signaler toute erreur, merci de me contacter à l'adresse ar.sider@univ-bejaia.dz.