

Extrait du référentiel : BTS Systèmes Numériques option A (Informatique et Réseaux)		Niveau(x)
<p>S7. Réseaux, télécommunications et modes de transmission</p> <p>S7.1. Concepts fondamentaux de la Transmission</p>	<p>Modèle en couches et protocoles de l'Internet : IP, ICMP, ARP, UDP, TCP, etc.</p>	<p>3</p>

Objectifs du cours :

Ce cours traitera essentiellement les points suivants :

- Le protocole UDP :
 - service minimal
 - interface entre UDP et l'application
- Résumé sur les protocoles de transport TCP et UDP

Pré-requis :

- Cours sur le protocole de transport TCP

LE PROTOCOLE UDP (USER DATAGRAM PROTOCOL)

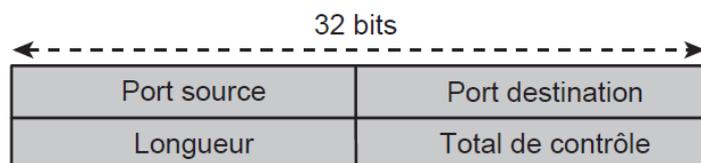
UDP est un protocole de transport sans connexion qui permet l'émission de messages sans l'établissement préalable d'une connexion. C'est un protocole non fiable, beaucoup plus simple que TCP, car il n'ajoute aucune valeur ajoutée par rapport aux services offerts par IP. L'utilisateur n'est donc pas assuré de l'arrivée des données dans l'ordre où il les a émises, pas plus qu'il ne peut être sûr que certaines données ne seront ni perdues, ni dupliquées, puisqu'UDP ne dispose pas des mécanismes de contrôle pour vérifier tout cela. De ce fait, il n'introduit pas de délais supplémentaires dans la transmission des données entre l'émetteur et le récepteur. C'est la raison pour laquelle il est utilisé par les applications qui ne recherchent pas une grande fiabilité des données ou qui ne veulent pas assumer la lourdeur de gestion des mécanismes mis en jeu dans le mode connecté.

SERVICE MINIMAL

UDP est efficace pour le transfert des serveurs vers les clients avec des débits élevés, en délivrant les données sous forme de datagrammes de petite taille et sans accusé de réception. Ce type de service est utile pour les applications en **temps réel**, telles que les émissions en flux continu d'informations audio et vidéo. En effet, pour celles-ci, la perte d'une partie des données n'a pas grande importance. Les jeux en réseau, le « streaming » (procédé permettant de lire des fichiers audio ou vidéo avant même que celui-ci soit totalement téléchargé, grâce à une mise en mémoire tampon) utilisent aussi UDP. D'autres applications de type questions-réponses, comptant de petites quantités de données, peuvent également utiliser UDP. De ce fait, l'erreur ou la perte d'un datagramme sont gérées directement par l'application elle-même, le plus souvent à l'aide d'un mécanisme de temporisation.

Au-dessus d'UDP, on trouve en particulier : le service d'annuaire DNS (Domain Name System), la transmission des informations de gestion de réseaux SNMP (Simple Network Management Protocol) ou d'informations de routage RIP (Routing Information Protocol).

Les messages UDP sont généralement appelés **datagrammes UDP**. Ils contiennent deux parties, un en-tête et des données encapsulées dans les paquets IP, comme les segments TCP.



Format du datagramme UDP

L'en-tête très simple compte quatre champs :

- **Port source** (16 bits). Il s'agit du numéro de port correspondant à l'application émettrice du paquet. Ce champ représente une adresse de réponse pour le destinataire.
- **Port destination** (16 bits). Contient le port correspondant à l'application de la machine à laquelle on s'adresse. Les ports source et destination ont la même signification que pour TCP.
- **Longueur** (16 bits). Précise la longueur totale du datagramme UDP, exprimée en octets. La longueur maximale des données transportées dans le datagramme UDP est de : $2^{16} - 4 \times 16$, soit 65 472 octets.
- **Somme de contrôle ou checksum** (16 bits). Bloc de contrôle d'erreur destiné à contrôler l'intégrité de l'en-tête du datagramme UDP, comme dans TCP.

INTERFACE ENTRE L'UDP ET L'APPLICATION

Les processus applicatifs utilisent des sockets UDP. Leur manipulation est très simple puisque le protocole n'est pas en mode connecté : il n'y a pas de procédure de connexion et donc pas de fermeture non plus. Comme pour TCP, du côté du serveur, il faut d'abord créer le socket et le paramétrer par la primitive « Bind », en lui associant le numéro de port correspondant. Puis il faut le placer dans un état d'attente des données du client (primitive « Listen »).

Côté client, il faut créer le socket. Le transfert des données peut commencer directement en utilisant des primitives « Read » et « Write » (ou Sendto et Receivefrom).



COURS

Le protocole de transport UDP

Cours sur le protocole
UDP-prof.doc

2^{ème} année

Page: 3/3

RÉSUMÉ

Deux protocoles de transport sont utilisés dans l'architecture TCP/IP. Le premier, TCP, est un protocole complet, destiné à pallier toutes les défaillances de l'interconnexion de réseaux. C'est un protocole en mode connecté qui met en oeuvre une détection et une correction d'erreurs, un contrôle de séquence, de flux et de congestion. Il est de ce fait complexe et lourd à gérer. TCP est indispensable pour toutes les applications qui transfèrent de grandes quantités d'informations et qui ont besoin de fiabilité dans les échanges de données.

UDP, lui, utilise le protocole IP pour acheminer un message d'un ordinateur à un autre, sans aucune valeur ajoutée (pas de connexion, pas de contrôle d'erreur, de contrôle de flux ni de contrôle de séquence) par rapport aux services rendus par IP. Il convient aux applications de type requête/réponse simples ou ayant des contraintes temporelles fortes.