

Objectifs du TD : Connaître les différences majeures entre UDP et TCP au niveau des transports de paquets. Comprendre l'impact de l'utilisation de ces deux protocoles sur les applications.

Notations et rappel :

Le protocole de transport **TCP** se charge de remettre les paquets qu'il reçoit dans l'**ordre**, afin de délivrer à l'application destinataire les données dans l'ordre dans lequel l'application expéditrice les a envoyées.

Le protocole **TCP** gère aussi la **duplication** : le même paquet ne peut être reçu remis deux fois à l'application destinataire.

Ni la duplication, ni la remise dans l'ordre des paquets n'est gérée par le protocole **UDP**.

1 Transport de paquets : acheminement, ordre, fiabilité et duplication

Exercice 1

Si deux paquets sont envoyés par la même source vers la même destination, suivent-ils le même chemin entre la source et la destination si les paquets sont :

1. des paquets UDP ?
2. des paquets TCP ?
3. des paquets IP ?

Exercice 2

1. Montrer comment deux paquets TCP p_1 et p_2 , expédiés dans un ordre donné, peuvent arriver dans l'ordre inverse dans la couche TCP de destination. Donner deux exemples amenant à ce type de «désordre».
2. Décrire une configuration de réseau (et/ou un cas) dans laquelle l'ordre de réception est forcément identique à celui de l'expédition.
3. Est-ce que les réponses seraient différentes dans le cas d'un protocole de transport comme UDP ?

Exercice 3

On souhaite mettre en place une application distribuée utilisant un protocole **orienté message, sans connexion et fiable**. Elle est destinée à s'exécuter dans le monde de l'*Internet* et adopte le protocole UDP. On suppose que l'application est implantée directement sur cette couche de transport (UDP). Elle doit alors mettre en place les éléments permettant d'assurer la fiabilité. Dans ce sens, on choisit de mettre en place un système d'accusé de réception. On suppose aussi qu'on dispose d'une primitive nommée **y-a-t-il-un-message** qui permet de savoir si un message est disponible ou pas en réception.

Note : pour l'instant, on ne se préoccupe pas de la possibilité de duplication : on admet qu'un message peut être dupliqué et que ça ne gêne pas l'exécution de l'application.

1. L'application a-t-elle besoin de la primitive **y-a-t-il-un-message** pour la mise en place de la fiabilité ?
2. On suppose qu'un accusé de réception est envoyé à chaque réception d'un message. Quel est le défaut le plus important à votre avis de ce choix ?
3. Que se passe-t-il si après plusieurs tentatives de réception, on n'a toujours pas d'accusé de réception ?
4. Que faut-il faire ?
5. Finalement, si on veut éviter la duplication dans une telle application que faut-il mettre en place ?

Exercice 4

Soient P_1 et P_2 deux programmes constituant une application distribuée. Construire un exemple de communication montrant comment une **duplication** de paquets peut se produire dans chacun des cas suivants :

1. P_1 et P_2 communiquent en TCP. Indiquer ce qui se passe au niveau des programmes (couche application des deux programmes).
2. P_1 et P_2 communiquent en UDP.

Exercice 5

Bien qu'un protocole comme UDP soit non fiable, dans le cas d'un réseau local un paquet ne peut se perdre que dans quelques cas précis. On admettra ici qu'il ne peut se perdre à cause d'un défaut électronique (*pas de disparition due au matériel*).

1. Illustrer deux exemples de perte de paquets UDP dans un réseau local composé d'un ensemble d'hôtes connectés et ne nécessitant pas de routeur. On admettra que les programmes (couche application) qui communiquent fonctionnent correctement.
2. Si on ajoute un routeur, illustrer un nouvel exemple de perte de paquet.
3. Reprendre les exemples des questions ci-dessus et expliquer dans chaque cas ce qui se serait passé si le protocole était TCP.