

INF4375 - Paradigmes des échanges Internet

Communication réseau

Jacques Berger

Objectifs

Comprendre le fonctionnement d'une communication réseau

Prérequis

Aucun

Rappel : Le modèle OSI

7. Application
6. Présentation
5. Session
4. Transport
3. Réseau
2. Liaison de données
1. Physique

Communication réseau

Pour effectuer une communication sur un réseau, une application doit pouvoir communiquer avec l'interface matérielle de la carte réseau, peu importe le type de réseau (Internet, LAN, Wifi, etc)

Communication réseau

Très souvent, nous utilisons des bibliothèques qui effectuent ces communications pour nous

Ces bibliothèques simplifient l'écriture d'application distribuées et diminuent la complexité du code à écrire

Communication réseau

La librairie utilise un service de communication réseau offert par le système d'exploitation : le socket

Les données envoyées par le socket sont prises en charge par le système d'exploitation, qui les passera à son tour au driver de la carte réseau

Socket

Un socket s'utilise comme un fichier :

- On doit l'ouvrir

- On peut y lire des données binaires

- On peut y écrire des données binaires

- On doit le fermer

Socket

Dans le modèle ISO, le socket se situe dans la couche de transport (4)

Il est nécessaire de spécifier le protocole de transport à utiliser pour la communication réseau (ex. TCP, UDP)

Socket

Nous utilisons les sockets directement lorsque nous avons besoin d'effectuer une communication réseau de bas niveau

Socket

Les sockets peuvent également servir à effectuer une communication inter-processus sur une même machine

Notamment, en utilisant l'adresse IP 127.0.0.1 ou son alias «localhost»

Socket

Les données échangées au travers de sockets sont brutes, du binaire seulement

Nous devons donc gérer la sérialisation et l'encodage de caractères des données

Socket

Un socket est un stream bidirectionnel vers un autre processus, généralement sur une autre machine

Modèle client-serveur

Un serveur offre un service

Il attend indéfiniment les connexions des clients, il écoute les connexions sur un port précis

Un port est un numéro servant à distinguer les différents canaux de communication

Modèle client-serveur

Pour chaque connexion client que le serveur reçoit, le système d'exploitation fournit un socket pour la communication entre le client et le serveur

Le serveur démarre un thread qui sera chargé de traiter les demandes du client et fournit le socket à ce thread

Modèle client-serveur

Le serveur continue d'écouter les demandes de connexion des clients pendant que les threads traitent les connexions courantes

Protocole de communication

Le protocole de communication est le format et la séquence des messages échangés entre le client et le serveur

Un serveur HTTP est un serveur ordinaire à l'exception qu'il s'attend à recevoir des messages HTTP et qu'il répond des messages HTTP

Protocole de communication

Un serveur HTTP va effectuer le parsing et l'interprétation d'une requête HTTP et passer les données à une application web

L'application va produire un résultat et le serveur HTTP va retourner le résultat dans un message HTTP

Tout ça, au travers de sockets sur le port HTTP (80 par défaut)

Avantages

Développer directement avec des sockets permet :

- D'avoir un protocole d'échange personnalisé
- D'avoir une très grande flexibilité

Inconvénients

Par contre, ce type de développement :

- Nécessite plus de temps de développement et d'effort

- Est plus difficile à mettre en place (interopérabilité, sécurité, etc.)

Fureteur

Un fureteur est un client

Par contre, le fureteur et le serveur n'utilisent pas toujours une architecture client-serveur avec une communication bidirectionnelle continue

Fureteur

Souvent, le fureteur doit faire des requêtes asynchrones (Ajax) au serveur dans le but d'aller chercher l'information que le serveur voudrait lui communiquer

Internet

Sur Internet, l'hétérogénéité du matériel et du logiciel complexifie les communications

Internet

Pour faciliter les transferts et améliorer l'interopérabilité des applications, nous utilisons des formats de sérialisation standardisés

JSON

XML

HTML

Etc.

Internet

Les préoccupations courantes d'une communication Internet :

L'encodage de caractères

La sécurité

L'interopérabilité