Programmation de la couche transport

Introduction aux réseaux locaux

Plan de la présentation

- Services de transport
- Communication entre processus
- Programmation TCP et UDP

Services de transport

Lien client serveur

- Pour créer un lien entre une application du client et une application du serveur chaque processus doivent créer un socket.
- Le **socket** devient la porte d'entrer pour communiquer entre l'application et la couche transport.
- Le socket possède une interface de programmation qui permet d'envoyer et de recevoir de l'information sur de la couche transport.
- Cette interface peut généralement être configurée selon des besoins précis.

Options de services

- Il existe différentes caractéristiques à tenir compte selon la couche de transport utilisée.
- Lors de la programmation de processus client-serveur il faut valider les points suivants:
  - **Fiabilité du transfert**
  - **Débit du transfert**
  - **Synchronisation**
  - **Sécurité**

Fiabilité du transfert

- Certaines applications doivent être garantie que les informations envoyés seront bien reçues. Elles doivent donc pouvoir compter sur un transfert fiable:
  - Ex: Transfert de fichier, web, etc.
- Par contre, d'autres application peuvent être **tolérante à la perte** d'information. Elle continuerons de fonctionner si une partie de l'information est manquante.
  - Ex: Vidéo en ligne, jeu vidéo, etc.

Débit du transfert

- Une application peut être **sensible au débit** (à la vitesse) de la bande passante.
  - Ex: Téléphonie par IP doivent fournir un débit minimal constant.
- À l'opposer d'autres application peuvent être plus **élastique** et ne pas souffrir d'un débit qui change.
  - Ex: Courriel, transfert de fichiers, etc.

Synchronisation

- La **latence** correspond au temps entre la demande du client et la réception de la réponse suite à cette demande.
- La latence est influencée par :
  - La vitesse du lien
  - Le temps de traitement de l'information du coté du serveur
  - Le temps de traitement de l'information du coté du client.
- Une application tel que la téléphonie par IP peut souffrir d'une mauvaise synchronisation à cause de la latence du coté du serveur:
  - Ex: Longue pause entre les réponses et envoies (Le débit est bon, mais le temps de déplacement et de traitement est long)

Sécurité

- Le cryptage et l’authentification des données reçues sont deux aspects importants de la sécurité.
- Les navigateurs offrent généralement ce type de sécurité pour l’utilisation adéquat de site web tel que les banques et le commerce en ligne.
  - SSL est une couche que l’on peut ajouter au dessus de TCP pour assurer la sécurité des données.

Services TCP vs UDP

TCP

- Doit établir une connexion avant l’échange de de données.
- Fiabilité du transfert des données
- Service de contrôle de congestion et de ralentissement.
- Aucun service de sécurité.

- Ne garantie pas le transfert.
- Transfert immédiat sans assurer l’ordre de réception.
- Aucun service de sécurité.

Communication entre processus

À qui le paquet?

- Pour être en mesure de livrer adéquatement les messages d'un hôte à un autre hôte, Internet utilise l'**adresse IP** pour spécifier la source et la destination.
- Par la suite, lorsque l'hôte reçoit un message, il regarde qu'elle processus ou application écoute sur un **port** déterminé.
- Le port complète donc l'adressage complet d'un message.

Port

- Le port est un index sur 16 bits.
- Les ports entre 0 et 1023 sont généralement utilisés pour les protocoles connues :
  - 20 & 21: File Transfer Protocol (FTP)
  - 22: Secure Shell (SSH)
  - 23: Telnet remote login service
  - 25: Simple Mail Transfer Protocol (SMTP)
  - 53: Domain Name System (DNS) service
  - 80: Hypertext Transfer Protocol (HTTP) used in the World Wide Web
  - 110: Post Office Protocol (POP3)
  - 119: Network News Transfer Protocol (NNTP)
  - 143: Internet Message Access Protocol (IMAP)
  - 161: Simple Network Management Protocol (SNMP)
  - 443: HTTP Secure (HTTPS)
- Et une liste plus exhaustive : http://www.httplab.it/index.php?sec=PortDef

L’adresse IP

- Elle sert à identifier l’hôte (la machine)
- Il existe présentement deux version d’adresse IP
- IPv4
  - Sur 32 bits elle est composée d’une série de 4 octets séparés par des points et représenté par un nombre décimal.
  - Ex: 192.168.4.3
- IPv6
  - Sur 128 bits elle est composée d’une série de 8 nombres hexadécimaux de 2 octets.
  - Ex: 2A01:E35:2421:4BE0:CDBC:C04E:A7AB:ECF3

Table des ports ‘ouverts’

- Finalement, pour être en mesure d’envoyer les bons messages au bon processus le système d’exploitation maintient une lise des descripteurs (handle) des ports qui écoutent sur chaque processus.
- Cette liste contient généralement les informations suivantes:
  - Local IP:Port, Foreign:Port, State, PID

Programmation TCP

Étapes de communication

- 1. Le serveur doit être actif et il doit écouter sur un port.
- 2. Le client doit aller cogner à la porte(le port) du serveur et initier une connexion TCP.
- 3. Le serveur acceptera la connexion et créera un autre socket (un lien/tube) de communication spécifique à ce client.
- 4. L’échange de données prend place à l’aide de ce lien/tube.
- 5. La connexion est sevré.

Spécifiquement

- En Java
  - Regarder pour les classes:
  - ServerSocket et Socket de java.net.*
- En C
  - Regarder les fonctions suivantes:
  - WSAStartup(), WSACleanup()
  - socket(), bind(), connect(), listen(), accept()
  - send() et recv()
  - shutdown() et close()
  - La structure : struct sockaddr_in et sockaddr
  - Le fichier d’inclusion : #include <winsock2.h>

Programmation UDP

Étapes de communication

- 1. Le serveur doit être actif et il doit écouter sur un port.
- 2. Le client crée un socket de communication.
- 3. L’échange de données prends place en spécifiant, à chaque fois, la destination. La source est contenu dans la réception des messages.
- 4. Le client détruit le socket de communication.

Spécifiquement

- En Java
  - Regarder pour les classes:
  - DatagramSocket,DatagramPacket et InetAdress de java.net.*
- En C
  - Regarder les fonctions suivantes:
  - WSAStartup(), WSACleanup()
  - socket(), bind()
  - sendto() et recvfrom()
  - shutdown() et La structure : struct sockaddr_in et sockaddr
  - Le fichier d’inclusion : #include <winsock2.h>

Points importants

- Socket
- Adresse IP
- Port
- Caractéristiques des services d’un transport
- Différences entre TCP et UDP

Des questions?

Sources