

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Télématique

Introduction aux sockets de TCP et UDP

Luc Lavoie
Département d'informatique
Faculté des sciences
luc.lavoie@USherbrooke.ca

IFT585 Télématique (T09-Transport-Sockets_v101d) 1

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Rappels

<input type="checkbox"/> APDU	<input type="checkbox"/> Échange (dialogue)
<input type="checkbox"/> Message (message)	<input type="checkbox"/> Communication
<input type="checkbox"/> TPDU	<input type="checkbox"/> Connexion
<input type="checkbox"/> Segment (segment)	<input type="checkbox"/> Circuit
<input type="checkbox"/> NPDU	<input type="checkbox"/> Lien
<input type="checkbox"/> Paquet (packet)	<input type="checkbox"/> Canal
<input type="checkbox"/> LPDU	<input type="checkbox"/> Voie de transmission
<input type="checkbox"/> Trame (frame)	

IFT585 Télématique (T09-Transport-Sockets_v101d) 2

UNIVERSITÉ DE SHERBROOKE 2009-03-28

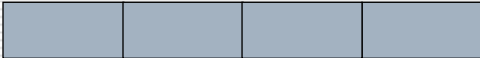
TCP et UDP

TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
<ul style="list-style-type: none"> <input type="checkbox"/> Avec connexion <input type="checkbox"/> Contrôle de séquence <input type="checkbox"/> Accusé de réception <input type="checkbox"/> Système de fenêtrage <input type="checkbox"/> Sert les applications comme FTP et HTTP <input type="checkbox"/> Moins rapide et plus fiable 	<ul style="list-style-type: none"> <input type="checkbox"/> Sans connexion <input type="checkbox"/> Pas de contrôle de séquence <input type="checkbox"/> Pas d'accusé de réception <input type="checkbox"/> Pas de fenêtrage <input type="checkbox"/> Sert les applications comme SNMP, DHCP, DNS et TFTP <input type="checkbox"/> Plus rapide et moins fiable

IFT585 Télématique (T09-Transport-Sockets_v101d) 3

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Format du segment UDP

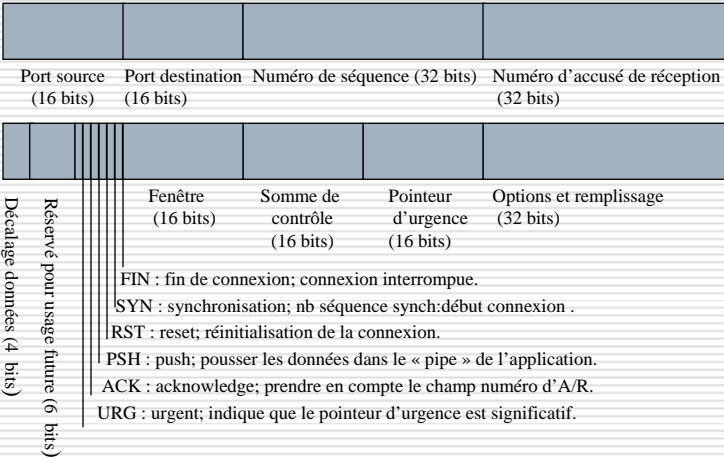


Port source (16 bits)	Port destination (16 bits)	Longueur (16 bits)	Somme de contrôle (16 bits)

IFT585 Télématique (T09-Transport-Sockets_v101d) 4

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Format du segment TCP



Port source (16 bits) Port destination (16 bits) Numéro de séquence (32 bits) Numéro d'accusé de réception (32 bits)

Fenêtre (16 bits) Somme de contrôle (16 bits) Pointeur d'urgence (16 bits) Options et remplissage (32 bits)

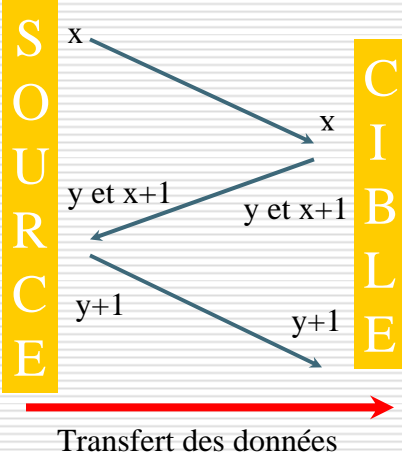
Décalage données (4 bits) Réserve pour usage futur (6 bits)

FIN : fin de connexion; connexion interrompue.
 SYN : synchronisation; nb séquence synch:début connexion .
 RST : reset; réinitialisation de la connexion.
 PSH : push; pousser les données dans le « pipe » de l'application.
 ACK : acknowledge; prendre en compte le champ numéro d'A/R.
 URG : urgent; indique que le pointeur d'urgence est significatif.

IFT585 Télématique (T09-Transport-Sockets_v101d) 5

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Ouverture de la connexion TCP (3-way handshake)



- La source envoie une demande de connexion avec un numéro de séquence x.
- Le destinataire répond avec un accusé de réception (x+1) et un nouveau numéro de séquence y.
- La source répond avec l'accusé de réception y+1.
- La connexion est établie.
- Le transfert des données peut débuter.

Transfert des données

IFT585 Télématique (T09-Transport-Sockets_v101d) 6

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Les accusés de réception TCP

(Accusés de réception prévisionnels)

Les segments qui ne font pas l'objet d'un accusé de réception dans un délai donné sont retransmis.

IFT585 Télématique (T09-Transport-Sockets_v101d) 7

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Fenêtres coulissantes TCP

La taille de la fenêtre est déterminée de façon dynamique. Le fenêtrage permet une meilleure utilisation de la bande passante.

IFT585 Télématique (T09-Transport-Sockets_v101d) 8

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Numéros de séquence et d'accusé de réception

Source	Dest.	Séq.	Acc.	...
1028	23	13	1	...

→

Source	Dest.	Séq.	Acc.	...
23	1028	1	14	...

←

Source	Dest.	Séq.	Acc.	...
1028	23	14	2	...

À la destination, le protocole TCP rassemble les segments pour reformer le message. Les segments seront retransmis.

IFT585 Télématique (T09-Transport-Sockets_v101d) 9

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Les numéro de port

- La couche de Transport utilise des numéros de ports correspondant aux applications auxquelles elle s'adresse.

FTP	TELNET	SMTP	DNS	FTP	HTTP	SNMP
21	23	25	53	69	80	161
TCP ou UDP						

- Les numéros inférieurs à 255 sont réservés aux applications publiques.
- Les numéros de 255 à 1023 sont attribués aux entreprises pour les applications à commercialiser.
- Les numéros supérieurs à 1023 (4095) ne sont pas attribués.

IFT585 Télématique (T09-Transport-Sockets_v101d) 10

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Sources et objectifs

- Sources
 - Matériel inspiré de Kurose (voir module précédent)
 - Folklore Java
- Objectifs
 - Séparer les services d'application des services de transport
 - Construire des applications client/serveur qui communiquent à l'aide de sockets

IFT585 Télématique (T09-Transport-Sockets_v101d) 11

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Vocabulaire

- Un **flot** est une séquence de caractères qui entre ou sort d'un processus.
- Un **flot d'entrée** est lié à la source d'entrée du processus, par exemple clavier ou socket.
- Un **flot de sortie** est lié à la source de sortie, par exemple moniteur ou socket.
- **flot et flux** :
 - dans les couches 1 à 3 (physique, liaison de données et réseau)
→ on parle de **flux** de données.
 - dans les couches 4 à 7 (transport, session, présentation, application)
→ on parle de **flot** de données.

IFT585 Télématique (T09-Transport-Sockets_v101d) 12

UNIVERSITÉ DE
SHERBROOKE

2009-03-28

Sockets

Contexte et définition

- API (*application programming interface*)
- Introduit en 1981 dans BSD4.1 UNIX
- Modèle client-serveur
- Conçu pour être utilisé par la couche application
- Offrant deux types de service
 - UDP : datagrammes non fiables
 - TCP : « flots d'octets » fiables
- Une interface grâce à laquelle un processus peut envoyer et recevoir des messages d'un autre processus
- Le socket est mis en oeuvre par la mise en commun de ressources de trois sources : l'application, la couche transport et le système d'exploitation

IFT585 Télématique (T09-Transport-Sockets_v101d) 13

UNIVERSITÉ DE
SHERBROOKE

2009-03-28

Sockets

Analogie

- Le transfert d'octets fiable et ordonné de TCP est analogue au « pipe » de Unix.

IFT585 Télématique (T09-Transport-Sockets_v101d) 14

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Illustration avec TCP

Socket : une porte entre processus d'application et protocole de transport bout-en-bout (UDP ou TCP)

Service TCP : transfert fiable d'octets d'un processus à un autre

IFT585 Télématique (T09-Transport-Sockets_v101d) 15

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Séquence global avec TCP

Préalable, le processus du serveur

- doit être actif
- doit avoir créé un socket qui accepte le « contact » avec le client

Le client contacte le serveur en

- créant un socket TCP (sur la base de l'adresse IP et du numéro du port du socket du processus du serveur)

Conséquent

- Le serveur crée un nouveau socket pour que le processus du serveur puisse communiquer avec le client
 - permet au serveur de parler avec plusieurs clients
 - les numéros de port source sont utilisés pour distinguer les clients
- Une connexion est établie entre le client et le serveur

IFT585 Télématique (T09-Transport-Sockets_v101d) 16

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Un exemple avec TCP

Exemple d'application client-serveur

1. Le client lit une ligne d'une entrée standard (flot **inFromUser**) et l'envoi au serveur via le socket (flot **outToServer**).
2. Le serveur lit une ligne du socket.
3. Le serveur convertit la ligne en capitales et la renvoie au client.
4. Le client lit, imprime, modifie la ligne du socket (flot **inFromServer**).

The diagram shows a large rounded rectangle labeled 'processus client'. At the top, 'clavier' (keyboard) and 'moniteur' (monitor) are connected to the process. Inside the process, there are two vertical boxes: 'inFromUser' (input from user) and 'outToServer' (output to server). Below these is a red box labeled 'socket TCP client'. At the bottom, there are two arrows: one pointing down labeled 'vers le réseau' (to the network) and one pointing up labeled 'du réseau' (from the network). Inside the process, there are also two vertical boxes: 'outToServer' (output to server) and 'inFromServer' (input from server). The flow is: clavier -> inFromUser -> outToServer -> socket TCP client -> vers le réseau. From the network, du réseau -> inFromServer -> outToServer -> inFromServer -> moniteur.

IFT585 Télématique (T09-Transport-Sockets_v101d) 17

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Interaction client/serveur avec TCP

Serveur (hostname, port X)

```
welcomeSocket =
  ServerSocket(X)

attend une requête de
connexion dans :
connectionSocket =
welcomeSocket.accept()

lit la requête dans
connectionSocket

écrit la réponse dans
connectionSocket

ferme
connectionSocket
```

Configuration de
la connexion TCP

Client

```
clientSocket =
Socket(hostname,X)

envoie une requête à l'aide de
clientSocket

répond à partir de
clientSocket

ferme
clientSocket
```

IFT585 Télématique (T09-Transport-Sockets_v101d) 18

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Exemple TCP : le client Java

```

import java.io.*;
import java.net.*;
class TCPClient {

    public static void main(String argv[]) throws Exception
    {
        String sentence;
        String modifiedSentence;

        Crée
        flot d'entrée
        BufferedReader inFromUser =
            new BufferedReader(new InputStreamReader(System.in));

        Crée socket client,
        se connecte
        au serveur
        Socket clientSocket = new Socket("hostname", 6789);

        Crée
        flot de sortie
        lié au socket
        DataOutputStream outToServer =
            new DataOutputStream(clientSocket.getOutputStream());
    }
}

```

IFT585 Télématique (T09-Transport-Sockets_v101d) 19

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Exemple TCP : le client Java, suite

```

        Crée
        flot d'entrée
        lié au socket
        BufferedReader inFromServer =
            new BufferedReader(new
            InputStreamReader(clientSocket.getInputStream()));

        envoi ligne
        au serveur
        sentence = inFromUser.readLine();
        outToServer.writeBytes(sentence + '\n');

        lit ligne
        du serveur
        modifiedSentence = inFromServer.readLine();
        System.out.println(" Reçu du serveur : " + modifiedSentence);

        clientSocket.close();
    }
}

```

IFT585 Télématique (T09-Transport-Sockets_v101d) 20

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Exemple TCP : le serveur Java

```

import java.io.*;
import java.net.*;

class TCPServer {

    public static void main(String argv[]) throws Exception
    {
        String clientSentence;
        String capitalizedSentence;

        ServerSocket welcomeSocket = new ServerSocket(6789);

        while(true) {

            Socket connectionSocket = welcomeSocket.accept();

            BufferedReader inFromClient =
                new BufferedReader(new
                    InputStreamReader(connectionSocket.getInputStream()));

```

Crée socket de démarrage au port 6789 →

Attend le contact du client, sur le socket de démarrage →

Crée flot d'entrée lié au socket →

IFT585 Télématique (T09-Transport-Sockets_v101d) 21

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Exemple TCP : le serveur Java, suite

```

        DataOutputStream outToClient =
            new DataOutputStream(connectionSocket.getOutputStream());

        clientSentence = inFromClient.readLine();

        capitalizedSentence = clientSentence.toUpperCase() + '\n';

        outToClient.writeBytes(capitalizedSentence);
    }
}

```

Crée flot de sortie lié au socket →

Lit ligne d'entrée du socket →

Écrit ligne de sortie dans le socket →

Fin de la boucle while, boucle avec retour et attend une autre connexion avec un client →

IFT585 Télématique (T09-Transport-Sockets_v101d) 22

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Séquencement global avec UDP

- ❑ l'émetteur inclut explicitement l'adresse IP et le port de destination dans chaque paquet
- ❑ le serveur doit extraire l'adresse IP et le port de l'émetteur dans le paquet reçu

- ❑ pas de connexion entre le client et le serveur
- ❑ les données transmises peuvent être reçues par l'application dans le désordre ou même perdues

du point de vue de l'application
 UDP fournit un transfert non fiable de datagrammes entre le client et le serveur

IFT585 Télématique (T09-Transport-Sockets_v101d) 23

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Interaction client/serveur avec UDP

Serveur (hostname,X)

```

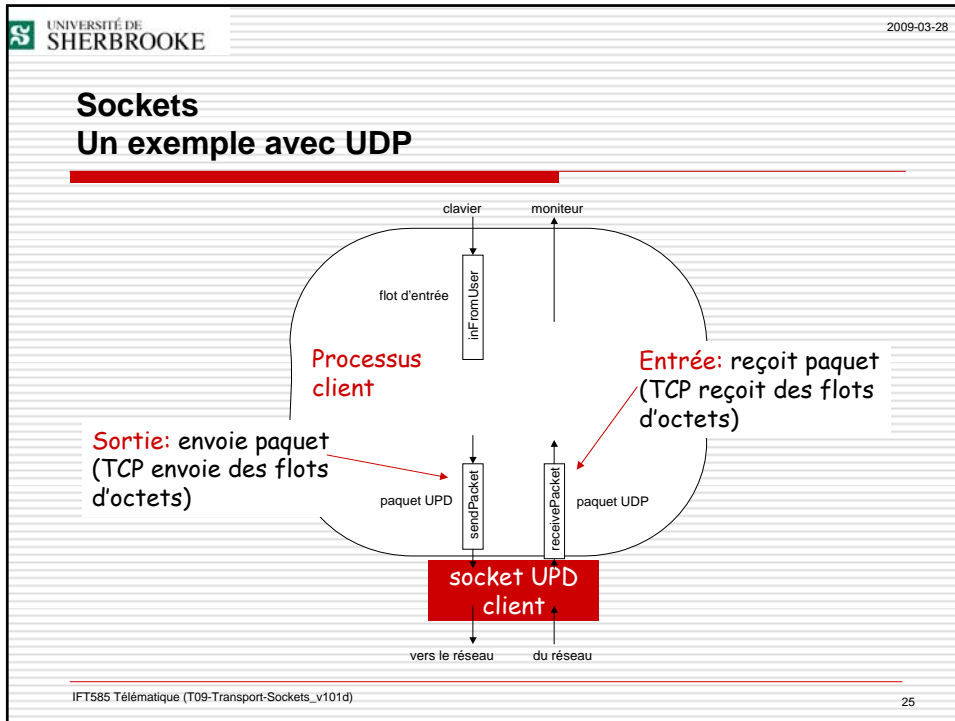
ServerSocket = DatagramSocket()
lit requête de ServerSocket
écrit réponse dans ServerSocket
spécifie adresse de l'équipement et numéro du port du client
                    
```

Client

```

créer socket, clientSocket = DatagramSocket()
↓
crée adresse (hostname, port=x), envoie datagramme requête à l'aide de clientSocket
↓
lit la réponse dans clientSocket
↓
ferme clientSocket
                    
```

IFT585 Télématique (T09-Transport-Sockets_v101d) 24



UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets Exemple UDP : le client Java

```

import java.io.*;
import java.net.*;

class UDPCient {
    public static void main(String args[]) throws Exception
    {
        Crée flot d'entrée → BufferedReader inFromUser =
                           new BufferedReader(new InputStreamReader(System.in));
        Crée socket client → DatagramSocket clientSocket = new DatagramSocket();
        Convertit le nom de l'équipement en adresse IP à l'aide du DNS → InetAddress IPAddress = InetAddress.getByName("hostname");

        byte[] sendData = new byte[1024];
        byte[] receiveData = new byte[1024];

        String sentence = inFromUser.readLine();
        sendData = sentence.getBytes();
    }
}
    
```

IFT585 Télématique (T09-Transport-Sockets_v101d) 26

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Exemple UDP : le client Java, suite

```

DatagramPacket sendPacket =
    new DatagramPacket(sendData, sendData.length, IPAddress, 9876);

clientSocket.send(sendPacket);

DatagramPacket receivePacket =
    new DatagramPacket(receiveData, receiveData.length);

clientSocket.receive(receivePacket);

String modifiedSentence =
    new String(receivePacket.getData());

System.out.println("FROM Serveur:" + modifiedSentence);
clientSocket.close();
}
}

```

Crée datagramme avec longueur des données à transmettre, adresse IP, port

Envoie datagramme au serveur

Lit datagramme du serveur

IFT585 Télématique (T09-Transport-Sockets_v101d) 27

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Exemple UDP : le serveur Java

```

import java.io.*;
import java.net.*;

class UDPServeur {
    public static void main(String args[]) throws Exception
    {
        DatagramSocket ServerSocket = new DatagramSocket(9876);

        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];

        while(true)
        {
            DatagramPacket receivePacket =
                new DatagramPacket(receiveData, receiveData.length);

            ServerSocket.receive(receivePacket);
        }
    }
}

```

Crée socket datagramme au port 9876

Crée espace pour datagramme à recevoir

Reçoit datagramme

IFT585 Télématique (T09-Transport-Sockets_v101d) 28

UNIVERSITÉ DE SHERBROOKE 2009-03-28

Sockets

Exemple UDP : le serveur Java, suite

```

String sentence = new String(receivePacket.getData());

    Obtient
    adresse IP
    et no de port
    de l'émetteur
    → InetAddress IPAddress = receivePacket.getAddress();
    → int port = receivePacket.getPort();

String capitalizedSentence = sentence.toUpperCase();

sendData = capitalizedSentence.getBytes();

    Crée datagramme
    à envoyer au client
    → DatagramPacket sendPacket =
      new DatagramPacket(sendData, sendData.length, IPAddress,
        port);

    Écrit
    datagramme
    dans le socket
    → ServerSocket.send(sendPacket);
  }
}
  
```

Fin de la boucle while,
boucle avec retour et
attend un autre datagramme

IFT585 Télématique (T09-Transport-Sockets_v101d) 29