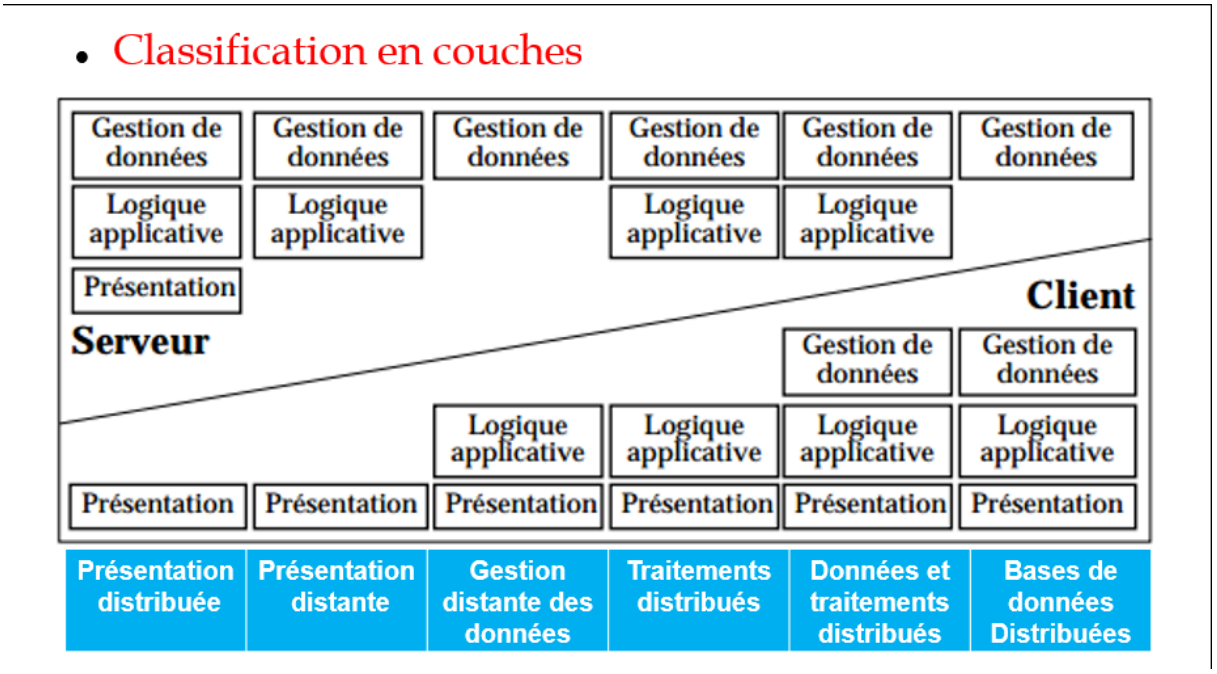
Faculté des Sciences Dhar El Mahraz, Fès **Département Informatique**

Module Programmation réseaux, SMI S6 : session de rattrapage
2014-2015
Durée : 1h30

Exercice 1 : Architecture Client – Serveur

1. Donner schématiquement, le découpage en couches proposé par Gartner Group



2. Comment est appelée la couche qui permet les interactions entre l'utilisateur et l'application ?

couche de Présentation

3. Comment est appelée la couche qui permet de contrôler le dialogue entre la logique métier et les informations stockées par l'application ?

couche données

4. Qu'est-ce qui permet de déterminer si une application est "1 tiers", "2 tiers", "3 tiers" ou "n tiers"?
   a. Le découpage de l'application en différentes entités logicielles.
   b. Le nombre de serveurs utilisés pour cette application.

c. La répartition des postes clients.

5. Comment qualifier une architecture d'une application dans laquelle les couches de présentation, applicative, de données sont sur le poste client et les données manipulées par cette application sont sur un serveur central.
   a. architecture 1 tiers centralisée
   b. architecture 2 tiers centralisée
   c. architecture 1 tiers répartie
   d. architecture 2 tiers répartie

Exercice 2 : URL et HTTP

1. On souhaite se connecter via le protocole HTTPS (HTTP sécurisé) au serveur de nom www.monsite.ma (attaché sur le port 443) sur la ressource de chemin /~teaching/programmation_reseau/test.php avec comme identifiant « SMI» nom et mot de passe « 1234 ».
   Cette page réalise la conversion d'un montant monétaire en dirham vers l'EURO : le champ (clé de requête) « m » accueille le montant en dirham. Proposer une URL permettant de convertir 300 dirhams en EURO.

   HTTPS://SMI:1234@www.monsite.ma:443/~teaching/programmation_reseau/test.php?m=300

2. Écrire la requête HTTP 1.1 (une ligne est suffisante) émise par le client permettant de télécharger la vidéo http://www.example.com/test/video.ogm. **Pour cela, on suppose que la connexion est déjà établie entre le client et le serveur**

GET /test/video.ogm HTTP/1.1

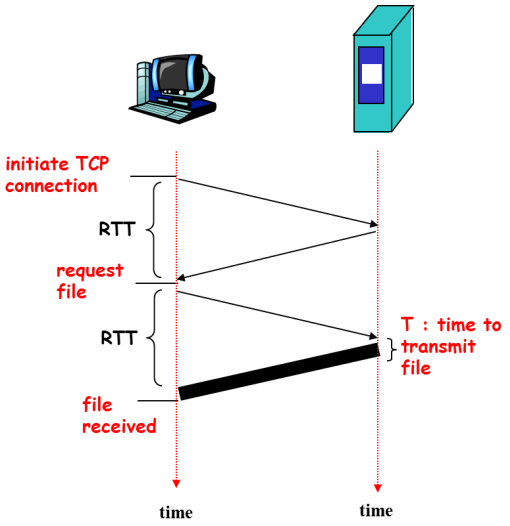
3. On considère le schéma suivant :

RTT: temps aller-retour entre le client et le serveur

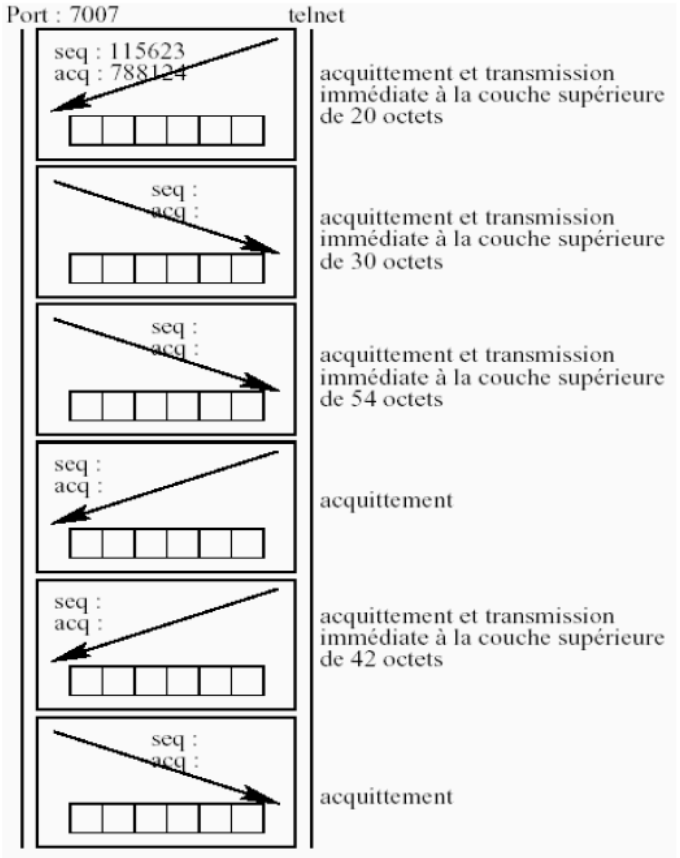
T : le temps moyen que met le serveur pour émettre un objet demandé

Considérons le cas ou le client sollicite la page index.html contenant 5 images

- Si l'échange entre le client et le serveur utilise la version 1.1 du HTTP, quel est le temps total (en fonction de T et RTT) que mettra le client pour recevoir index.html ?
- Si la version de http est 1.0, que devient ce temps total ?



Exercice 3 : TCP
Compléter le diagramme avec les numéros de séquence, d'acquittement (en décimal) et la séquence des 6 bits de drapeaux



On rappelle que les 6 bits de drapeau sont :

URG: Urgent pointer is valid	RST: Reset the connection
ACK: Acknowledgment is valid	SYN: Synchronize sequence numbers
PSH: Request for push	FIN: Terminate the connection

URG	ACK	PSH	RST	SYN	FIN

Exercice 4 : Lecture de code

On considère un système client-serveur avec des sockets TCP en mode connecté.
Compléter sur la copie la fonction main du serveur aux points marqués /* N. ??? */
(On n'écrira que les instructions qui manquent aux points N).

```c
#include "params.h"
void clientProcessing(int clisock, struct sockaddr_in*  cliaddr);
void traitantUsr1(int sig);
int main(void){
  int mess_sock;
  struct sockaddr_in seraddr;
  if((mess_sock= /* 1.??? */ )<0) ERRNO_EXIT;
  seraddr.sin_family = /* 2.??? */;
  seraddr.sin_addr.s_addr = /* 3.??? */;
  seraddr.sin_port = htons(PORT_SERVEUR_UDP) /* défini dans params.h */ ;
  /* Attachement d'un nom à la socket mess_sock 4.??? */
  /* Transformation de la socket en socket de service 5.??? */
  while(1){
    int clisock;
    struct sockaddr_in cliaddr;
    int cliaddr_len = sizeof(cliaddr);
    /* connexion d'un client 6.??? */
    int child_pid;
    if(!(child_pid= fork())){
      /* fermeture de la socket inutilisée 7.??? */
      clientProcessing(clisock, &cliaddr);
      exit(0);
    } else if (child_pid>0) {
      /* fermeture de la socket inutilisée 8.??? */
      /* association du traitantUsr1 au signal SIGUSR1 9.???*/
    } else ERRNO_EXIT;
  }
  exit(0);
}
```