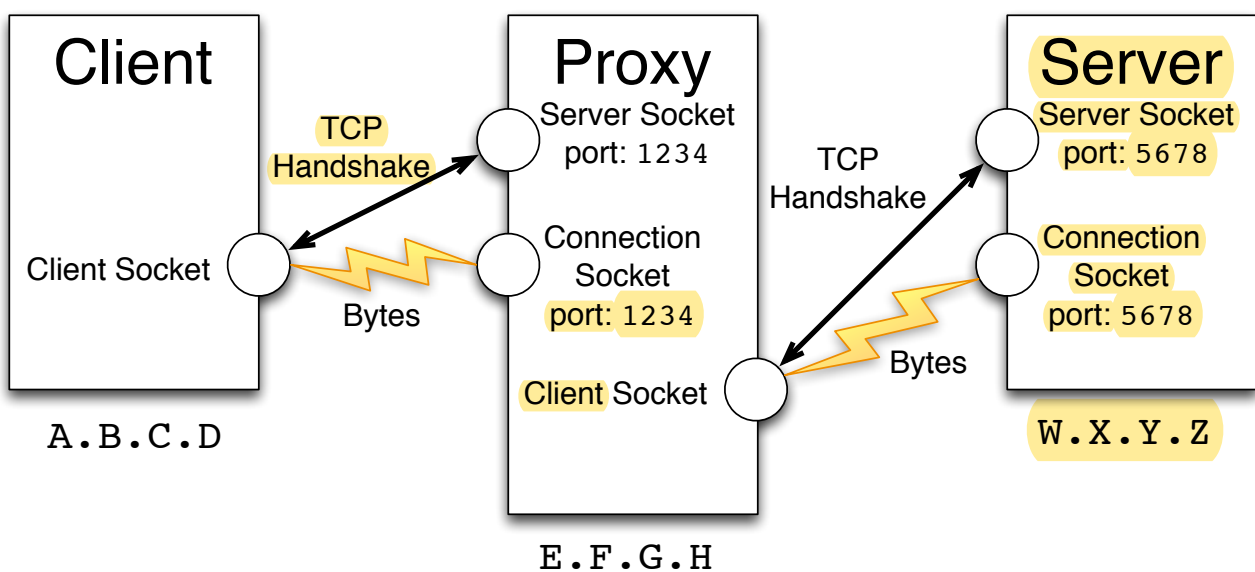


PROGRES - Mini-Projet 1

Sébastien Tixeul - Sebastien.Tixeul@lip6.fr

Exercice 1 - Proxy TCP

On cherche à programmer en Python un mécanisme de proxy entre un client et un serveur normalement connectés via TCP. Un proxy est un programme qui agit comme un serveur vis à vis du client, et comme un client vis à vis du serveur. Il retransmet au serveur toutes les données que le client auraient normalement transmises au serveur, et il retransmet au client toutes les réponses que le serveur auraient normalement transmises au client. Le schéma de connexion est résumé sur la figure suivante :



1. A partir des exemples de code vus en cours pour le client et le serveur séquentiel TCP, programmer le mécanisme de proxy en Python ; On considérera que l'adresse IP et le numéro de port du serveur sont stockés dans le code du proxy.
2. Tester les programmes client, proxy et serveur sur la même machine et sur des machines différentes (on pourra prendre comme serveur le serveur web installé par défaut, et comme client un navigateur web) ;
3. Tester le proxy avec plusieurs clients (ou plusieurs instances du même client) qui effectuent des requêtes simultanées sur le serveur (via le proxy) ;
4. Modifier le proxy pour qu'il puisse servir à plusieurs clients connectés simultanément, reprendre les tests des parties 2. et 3.



Exercice 2 - Proxy HTTP

On cherche à programmer en Python un proxy dédié au protocole HTTP. On considère donc que tous les échanges entre le client et le serveur (via le proxy) utilisent le protocole HTTP.

1. Modifier le proxy TCP pour qu'il fasse office de cache HTTP : la première fois qu'un fichier est fourni en argument de GET dans une requête HTTP, le proxy va chercher le fichier sur le serveur ; les fois suivantes, il envoie directement au client le fichier qu'il a stocké localement sans faire de nouvelle requête au serveur.



2. Modifier le proxy TCP pour qu'il fasse office de logueur HTTP, toutes les requêtes GET du client sont archivées dans un fichier de log (avec l'heure de la requête), toutes les réponses serveur aux requêtes GET sont archivées dans le fichier de log (avec la taille du fichier transmis).
3. Modifier le proxy TCP pour qu'il fasse office de censeur HTTP. Le proxy dispose maintenant d'une liste de fichiers interdits. Si le fichier demandé dans une requête GET du client fait partie de la liste des fichiers interdits, un fichier par défaut est retourné au client. Sinon, la requête est retransmise au serveur avant que la réponse correspondante soit retransmise au client.
4. Tester les proxy HTTP réalisés, d'abord individuellement (un proxy entre un client et un serveur), puis en les enchaînant (par exemple, un client est connecté à un cache HTTP qui est connecté à un logueur HTTP, qui lui même est connecté au serveur).

