
Rapport

Mini Projet 1

PROGRES



17/10/2022

Binôme :

- UNG Thierry

- ZOUHRI Bilal

Table de matières

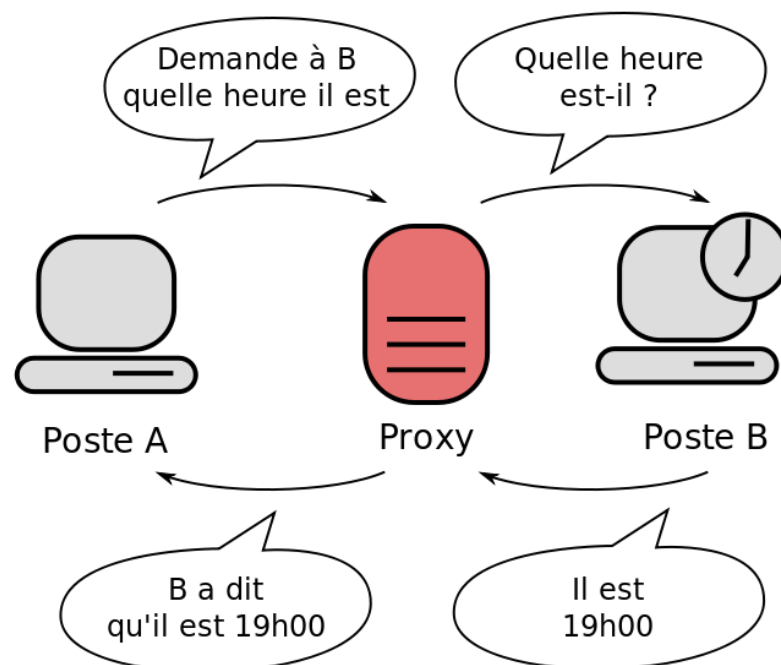
1	Introduction :	3
1.1.1	Relai :	3
1.1.2	Cache HTTP :	4
1.1.3	Sniffer HTTP :	4
1.1.4	Censeur HTTP :	4
2	Exercice 1 Relai TCP :	5
2.1.1	Explication fonctions.py :	5
3	Exercice 2 Relai HTTP :	6
3.1.1	Explication fonctions.py :	6
3.1.2	<u>Schéma de de fonctionnement du relai :</u>	<u>7</u>

1 Introduction :

1.1.1. Relai

Un relai (proxy) est un composant logiciel informatique qui joue le rôle d'intermédiaire en se plaçant entre deux hôtes pour faciliter ou surveiller leurs échanges.

Dans le cadre plus précis des réseaux informatiques, un proxy est alors un programme servant d'intermédiaire pour accéder à un autre réseau, généralement internet. Par extension, on appelle aussi "relai" un matériel comme un serveur mis en place pour assurer le fonctionnement de tels services. Le proxy se situe au niveau de la couche application (HTTP, FTP, SSH, etc. de niveau 7).



1.1.2. Cache HTTP

La plupart des relais assurent aussi une fonction de cache (en anglais *caching*) qui désigne le stockage temporaire des données (pages Web) fréquemment consultées, ce qui rend l'accès à l'avenir plus facile et plus rapide.

La mise en cache de document Web (ex : page web, images, ...) est utilisée afin de réduire la consommation de bande passante, la charge du serveur web (les tâches qu'il effectue), ou améliorer la rapidité de consultation lors de l'utilisation d'un navigateur web.

Un cache Web conserve des copies de documents transitant par son biais. Le cache peut, dans certaines conditions, répondre aux requêtes ultérieures à partir de ses copies, sans recourir au serveur Web d'origine.

1.1.3. Sniffer HTTP

Un analyseur de paquets, également connu sous le nom de renifleur de paquets, analyseur de protocole ou analyseur de réseau, est un programme informatique ou du matériel informatique, qui peut intercepter et enregistrer le trafic qui passe sur un réseau informatique ou une partie d'un réseau.

La capture de paquets est le processus d'interception et de journalisation du trafic. Au fur et à mesure que les flux de données circulent sur le réseau, l'analyseur capture chaque paquet et, si nécessaire, décode les données brutes du paquet, affichant les valeurs des différents champs du paquet, et analyse son contenu conformément à la RFC appropriée ou à d'autres spécifications.

1.1.4. Censeur HTTP

Le proxy désigne les limitations de l'information disponible par l'intermédiaire de ce réseau. Si l'URI demandée dans une requête GET du client contient un lien qui renvoie vers un site interdit, ce lien est remplacé par un message " Interdit " dans le corps de la réponse.

2 Exercice 1 Relai TCP

2.1.1. Explication fonctions.py

Client

La partie du client a des tâches simples, créé le socket client pour se connecter au relai, on tape ensuite le message que l'on veut envoyer au relai qui va l'envoyer au serveur, puis on attend de recevoir la réponse.

Relai

Le relai crée des sockets qui permette de se connecter au serveur et au client, quand il reçoit un message du client, il va l'envoyer au serveur. Le serveur envoie la réponse au relai, le relai va ensuite envoyer la réponse au client.

Serveur

Le serveur attend de recevoir le message du client via le relai, il va le décoder puis le renvoyer en majuscules au relai qui va l'envoyer au client.

3 Exercice 2 Relai HTTP

3.1.1. Explication fonctions.py

Client

On crée le socket client pour se connecter au relai, on tape ensuite la requête GET (l'Uri) que l'on veut envoyer au relai qui va l'envoyer au serveur.

Relai

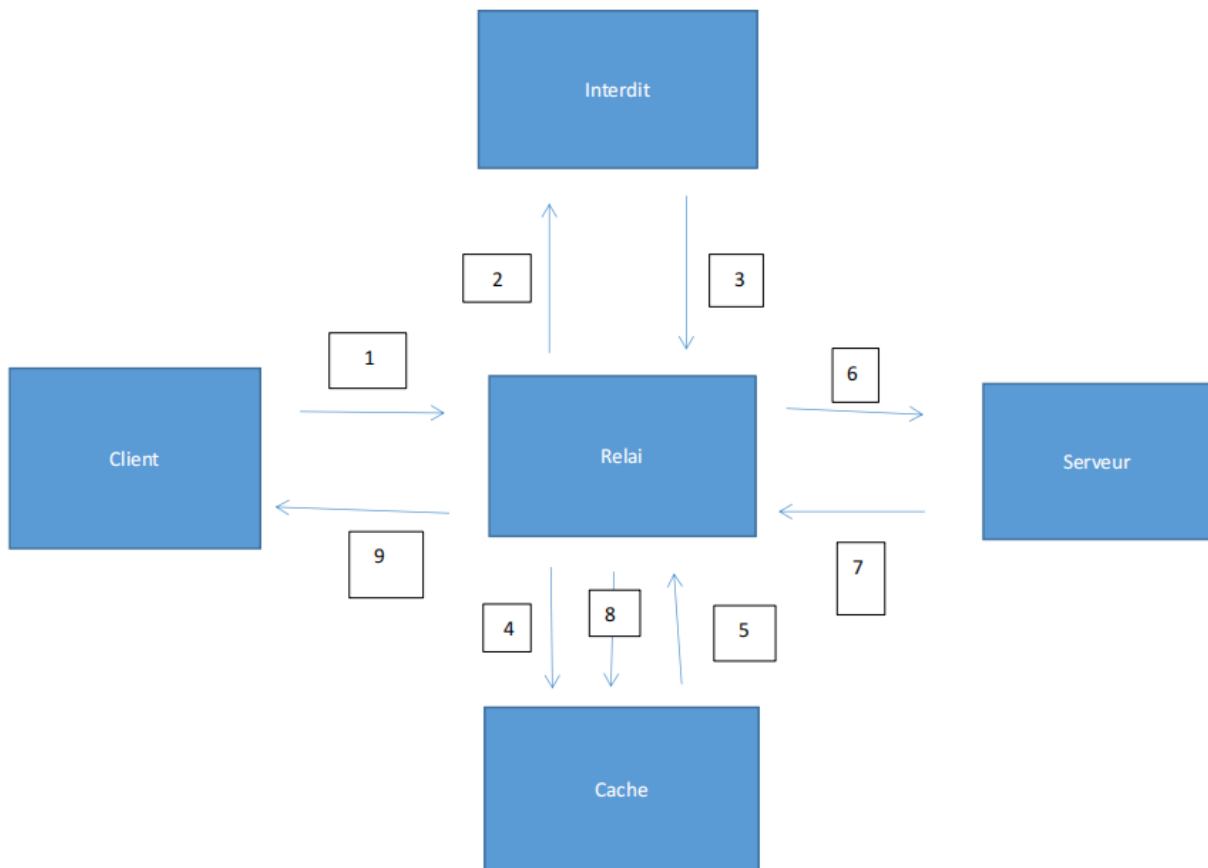
Le relai crée des sockets qui permette de se connecter au serveur et au client, quand il reçoit une requête du client, on a :

- Le relai stock la requête GET du client dans le fichier log client (sniffer.py)
- Le relai vérifie si l'Uri demandé existe dans sa liste des Uri censurés (censeur.py)
- Si l'Uri est interdit, le relai envoie au client 403 Forbidden (censeur.py)
- Si l'Uri n'est pas , le relai cherche si l'Uri existe dans sa liste cache (cache.py)
- Si l'Uri existe, le cache envoie son contenu directement au client 200 OK (cache.py)
- L'Uri n'existe pas, le relai demande au serveur de lui envoyer l'Uri demandé par le client
- Le relai stock la réponse du serveur dans le fichier log server (sniffer.py)
- Le relai envoie la réponse au client

Serveur

On crée le serveur, on attend de recevoir la requête du client, une fois reçu on essaye de se connecter au site web demandé , on récupère les infos du site, on l'encode et le renvoi au relai.

3.1.2. Schéma de fonctionnement du relai



Les étapes de fonctionnement de notre relai sont les suivantes :

- 1- Le client fait une requête
- 2- Le relai stock la requête Get dans son fichier de log client et vérifie si l'Uri demandé existe dans sa liste des Uri interdit.
- 3- Si l'Uri est interdit, le relai envoie au client 403 Forbidden (Uri interdit), si l'Uri n'est pas interdit on passe à l'étape 4
- 4- Le relai cherche si l'Uri existe dans sa liste cache
- 5- Si elle existe, le cache envoie son contenu directement au client (200 OK). Si l'Uri n'existe pas on passe à l'étape 6
- 6- Le relai demande au serveur de lui envoyer les informations de l'Uri demandées par le client
- 7- Le serveur envoie une réponse avec succès ou bien réponse négative (Uri non trouvé)
- 8- Le relai stock une copie du fichier reçu par le serveur si réponse positive du serveur et inscrit la réponse dans son fichier de log serveur
- 9- Ensuite le relai envoie la réponse ou le contenu du fichier au client

lien vidéo : <https://youtu.be/MiE21fKy-GI>
ps : écoute recommandé avec des écouteurs