Couche application : HTTP

But

Le but de ce TP est de vous faire découvrir le protocole applicatif le plus utilisé sur le réseau internet : HTTP (Hypertext Transfer Protocol) pour la consultation de sites web.

Introduction

En télécommunication un protocole est une convention utilisée par deux machines qui veulent communiquer entre-elles dans le but d'assurer un service (ex. transmettre un e-mail, télécharger un fichier ...).

Dans le cas du réseau Internet, les protocoles les plus couramment utilisés sont spécifiés par l'IETF (Internet Engineering Task Force) [1] sous la forme de documents textuels appelés RFC (Request for Comments). Les protocoles applicatifs définis à IETF sont pour la plupart textuels. Cette caractéristique est très commode, car elle facilite la compréhension et permet d'expérimenter le protocole directement en tapant les messages au clavier (ce qui sera le sujet du TP).

1 Consultation d'un site web avec HTTP

1.1 Introduction sur le protocole HTTP

Le protocole HTTP est défini par les RFC numérotés de 7230 à 7235 [2].

La table 1 contient un exemple de session HTTP, dans laquelle un client s'est connecté à un serveur pour lui demander de lui envoyer le contenu de la page `/test.html`.

1.1.1 Requête HTTP

Une requête HTTP est constituée d'une ligne contenant :
- une méthode (ici `GET` pour consulter un document)
- le chemin de la ressource (ici `/test.html`) sur laquelle la méthode doit s'exécuter
- la version du protocole utilisé (ici `HTTP/1.0`)

Cette ligne est optionnellement suivie d'un ou plusieurs en-têtes pour indiquer des informations additionnelles (ex. version du navigateur, préférences de l'utilisateur, etc.). Ces en-têtes ont un format similaire aux en-têtes des e-mails vus précédemment. Dans notre exemple l'en-tête `User-Agent` indique que le client est un navigateur Firefox dans sa version 17.0.

La fin de la requête est indiquée par une ligne vide (c'est-à-dire deux retours à la ligne successifs).

Note importante : dans la plupart des protocoles de l'IETF, le retour à la ligne s'effectue avec la séquence "`\r\n`" alors que les systèmes Unix/Linux utilisent simplement le caractère "`\n`". Si vous ne respectez pas cette convention, le serveur ne traitera pas correctement vos requêtes.

1. `http://www.ietf.org/`
2. Publiés en juin 2014. Tout comme son congénère SMTP, HTTP est en perpétuelle évolution. La toute première spécification (version 0.9) a été écrite en 1991 (`http://www.w3.org/Protocols/HTTP/AsImplemented.html`), elle tenait sur deux pages seulement. Il a ensuite été formalisé dans la version 1.0 en 1996 (RFC 1945, 60 pages) et la version 1.1 en 1997 (RFC 2068, 162 pages). Depuis ce temps la version 1.1 a été réécrite deux fois afin de lever les ambiguïtés et introduire de nouvelles fonctionnalités : une fois en 1999 (RFC 2168, 176 pages) et une fois en 2014 (RFC 7230-35, 305 pages). En 2015 a été publiée la version 2 du protocole (RFC7540, 96 pages) qui est sémantiquement équivalente à 1.1, mais qui utilise une syntaxe alternative pour optimiser l'utilisation de la bande passante et améliorer la réactivité des services. Il ne faut pas se laisser impressionner par la taille des documents, les bases de ce protocole sont restées simples et faciles à appréhender.

Client	Serveur

`GET /test.html HTTP/1.0`
`User-Agent: Mozilla/5.0 Firefox/17.0`
	`HTTP/1.0 200 OK`
	`Content-Type: text/html`
	`Content-Length: 79`
	
	`<html>`
	`<head><title>test</title></head>`
	`<body>`
	` This is a test`
	`</body>`
	`</html>`

TABLE 1 – Exemple de session HTTP

1.1.2 Réponse HTTP

La réponse du serveur est constituée d'une ligne de statut optionnellement suivie par un ou plusieurs en-têtes, d'une ligne vide, puis du corps de la réponse.

La ligne de statut contient :

— la version du protocole (ici `HTTP/1.0`)

— un code à 3 chiffres indiquant le résultat de la requête. Tout comme dans SMTP le chiffre des centaines correspond à une famille de réponses : succès (`2xx`), redirection (`3xx`), erreur due au client (`4xx`), erreur due au serveur (`5xx`).

— un message textuel correspondant au code (ici `OK`)

Les en-têtes qui peuvent suivre apportent des méta-données sur le contenu de la réponse. Dans notre exemple il y en a deux :

— `Content-Type` pour indiquer le type de contenu (ici : une page HTML)

— `Content-Length` pour indiquer la longueur de la réponse

Le corps de la réponse contient le contenu à distribuer (dans notre exemple, il s'agit du contenu du fichier `test.html`).

1.2 Utilisation de netcat

Les serveurs HTTP requièrent en général que la requête soit envoyée dans un laps de temps court (typiquement quelques secondes). Pour éviter que la connexion soit fermée brutalement, il est préférable de taper votre requête dans une commande (`echo`) et de rediriger la sortie vers l'entrée de `netcat`.

Ainsi pour consulter le fichier identifié par l'URL : `http://un.serveur.com/rep/fichier.txt` il faudra taper la commande suivante

```
echo -ne 'GET /rep/fichier.txt HTTP/1.0\r\n\r\n' | nc un.serveur.com 80
```

(note : par convention 80 identifie le service HTTP)

1.3 Envoi de requêtes HTTP

Envoyez des requêtes pour consulter plusieurs fichiers sur un serveur de votre choix (par exemple celui de l'université : `www.univ-rennes1.fr`) ou encore :

```
echo -ne 'GET /index.html HTTP/1.0\r\n\r\n' | nc TPRES.istic.univ-rennes1.fr 80
```

Si la commande de marche pas (i.e., avec un message de type "bad request"), vous pourrez obtenir le résultat escompté avec la commande suivante :

```
printf ``GET / HTTP/1.0\r\n\r\n'' | nc  TPRES.istic.univ-rennes1.fr  80
```

Si vous voulez utiliser HTTP 1.1, l'ajout du champs "Host" est obligatoire :

```
printf ``GET / HTTP/1.1\r\nHost: localhost\r\n\r\n'' | nc  TPRES.istic.univ-rennes1.fr  80
```

Question 1 *La réponse contient quelques en-têtes. Quelles métadonnées contiennent-ils ?*

1.4 Négociation de contenu

Envoyez une requête pour consulter la page `http://tpres.istic.univ-rennes1.fr/holy/bridge`. Ensuite, consultez cette même page dans votre navigateur web.

Question 2 *Quelles différences constatez-vous ?*

HTTP permet au client et serveur de négocier sous quelle forme le contenu demandé est transmis. Pour indiquer ses préférences le client peut inclure des en-têtes `Accept*`, le serveur indique la valeur retenue dans les en-têtes `Content*`. Par exemple :

- `Accept:` pour indiquer le format de contenu préféré. Ce format doit être spécifié sous la forme d'un type MIME[3], par exemple `text/plain` pour du texte brut et `text/html` pour du texte formaté en HTML.
- `Accept-Language:` pour indiquer la langue préférée sous la forme d'un code ISO 639-1, par exemple `fr` pour le français et `en` pour l'anglais
- `Accept-Encoding:` pour indiquer quels formats compression sont supportés par le client, par exemple `gzip` et `deflate`. Cela permet d'économiser de la bande passante.

Refaites à nouveau des requêtes pour la page `http://tpres.istic.univ-rennes1.fr/holy/bridge`, en faisant varier les contenu des champs `Accept, Accept-Language et Accept-Encoding`.

Dans un en-tête `Accept*`, il est possible de spécifier plusieurs valeurs séparées par des virgules (par exemple si l'utilisateur maîtrise plusieurs langues). Dans ce cas on peut indiquer l'ordre de préférence en rajoutant `;q=`*pref* où *pref* est une valeur comprise entre 0 et 1 (1 étant la valeur par défaut). Pour marquer une préférence pour le français, on peut indiquer :

```
Accept-Language: fr,en;q=0.9
```

Refaites à nouveau des requêtes pour la page `http://tpres.istic.univ-rennes1.fr/holy/bridge`, en testant différentes valeurs dans les champs `Accept*`.

Question 3 *En vous basant sur les observations que vous venez de faire, que contiennent les champs* `Accept*` *dans les requêtes envoyées par votre navigateur web ?*

2 Fin du TP

Si vous avez terminé en avance, vous pouvez prendre un petit peu de temps pour parcourir le RFC 5321 qui spécifie le protocole SMTP (notamment la section 3 qui décrit le comportement attendu d'un client et d'un serveur lors de l'envoi d'un mail) : `https://tools.ietf.org/html/rfc5321#section-3`

3. La liste officielle des types MIME est tenue par l'IANA à l'adresse : http ://www.iana.org/assignments/media-types