Outils informatiques

5. HTML et le protocole HTTP

DESS TEXTE

Introduction

- But du cours :
  - récupérer automatiquement des données accessibles sur le web pour en extraire les informations pertinentes.
- Prérequis à cela : maîtriser les notions que consituent HTML et HTTP
- Ce chapitre va donc présenter brièvement les réseaux informatiques, et le fonctionnement du W3 dans ce contexte.

Plan du chapitre

1. HTML et HTTP

2. Cheminement d'une requête :
   a. Analyse d'une requête
   b. Analyse d'une réponse

3. Traitement du corps de la réponse :
   a. Traitement d'une image
   b. Traitement d'un lien hypertexte

4. Les différentes couches réseau :
   a. PC et serveur sur le même sous-réseau
   b. PC et serveur sur deux sous-réseaux différents

1. HTML et HTTP

- **HTML** : langage utilisant des balises pour représenter la mise en forme de documents. Ce langage est interprété par une application appelée *navigateur Web* ou *arpenteur*.

- **HTTP** : protocole de communication pour le transfert de documents. La communication se fait entre un client (machine envoyant des requêtes) et un serveur (machine répondant à ces requêtes). Ce protocole est utilisé par les serveurs Web hébergeant des sites internet, dans le but de permettre de télécharger des documents ainsi que la consultation de pages sur l'écran du client.

- **Remarque** : il faut connaître HTTP pour communiquer avec un serveur Web sans passer par un navigateur (i.e. pour automatiser des requêtes internet).

2.a Analyse d'une requête

Soit la requête suivante :
`http ://hypothetical.ora.com/`

Ce qui provoque l'envoie du message suivant par le navigateur :

```
GET / HTTP/1.0
Connection : Keep Alive
User-Agent : Mozilla/3.0Gold (WinNT;I)
Host : hypothetical.ora.com
Aspect : image/gif,image/x-xbitmap,image/jpeg,text/html
```

2.b Analyse de la réponse

La serveur répond au navigateur par le message suivant :

```
HTTP/1.0 200 OK
Date : Fri, 04 Oct 2002 10:38:29 GMT
Server : Apache/1.1.1
Content-type: text/html
Content-length : 327
Last-Modified : Fri, 04 Oct 2002 09:28:12 GMT

<title>une page web</title>
(...)
```

3.a Traitement d'une image (1)

Lors de l'analyse syntaxique du document HTML ainsi parvenue, le navigateur arrive à la ligne suivante :

```
<img src=''/images/oreilly_mast.gif''>
```

Ce qui provoque l'envoie d'une seconde requête par le navigateur :

```
GET /images/oreilly_mast.gif HTTP/1.0
Connection : Keep Alive
User-Agent : Mozilla/3.0Gold (WinNT;I)
Host : hypothetical.ora.com
Aspect : image/gif,image/x-xbitmap,image/jpeg,text/html
```

3.a Traitement d'une image (2)

Ce à quoi le serveur répond par :

```
HTTP/1.0 200 OK
Date : Fri, 04 Oct 2002 10:38:29 GMT
Server : Apache/1.1.1
Content-type: image/gif
Content-length : 327
Last-Modified : Fri, 04 Oct 2002 09:28:12 GMT

[données du fichier gif]
```

3.b Traitement d'un lien hypertexte (1)

Si l'utilisateur clique sur un lient hypertexte tel que :

```
<a href=''/example2.html''>page intressante</a>
```

Le navigateur va agir de même, et envoyer une requête au serveur :

```
GET /example2.html HTTP/1.0
Connection : Keep Alive
User-Agent : Mozilla/3.0Gold (WinNT;I)
Host : hypothetical.ora.com
Aspect : image/gif,image/x-xbitmap,image/jpeg,text/html
```

3.b Traitement d'un lien hypertexte (2)

Ce à quoi le serveur répond par :

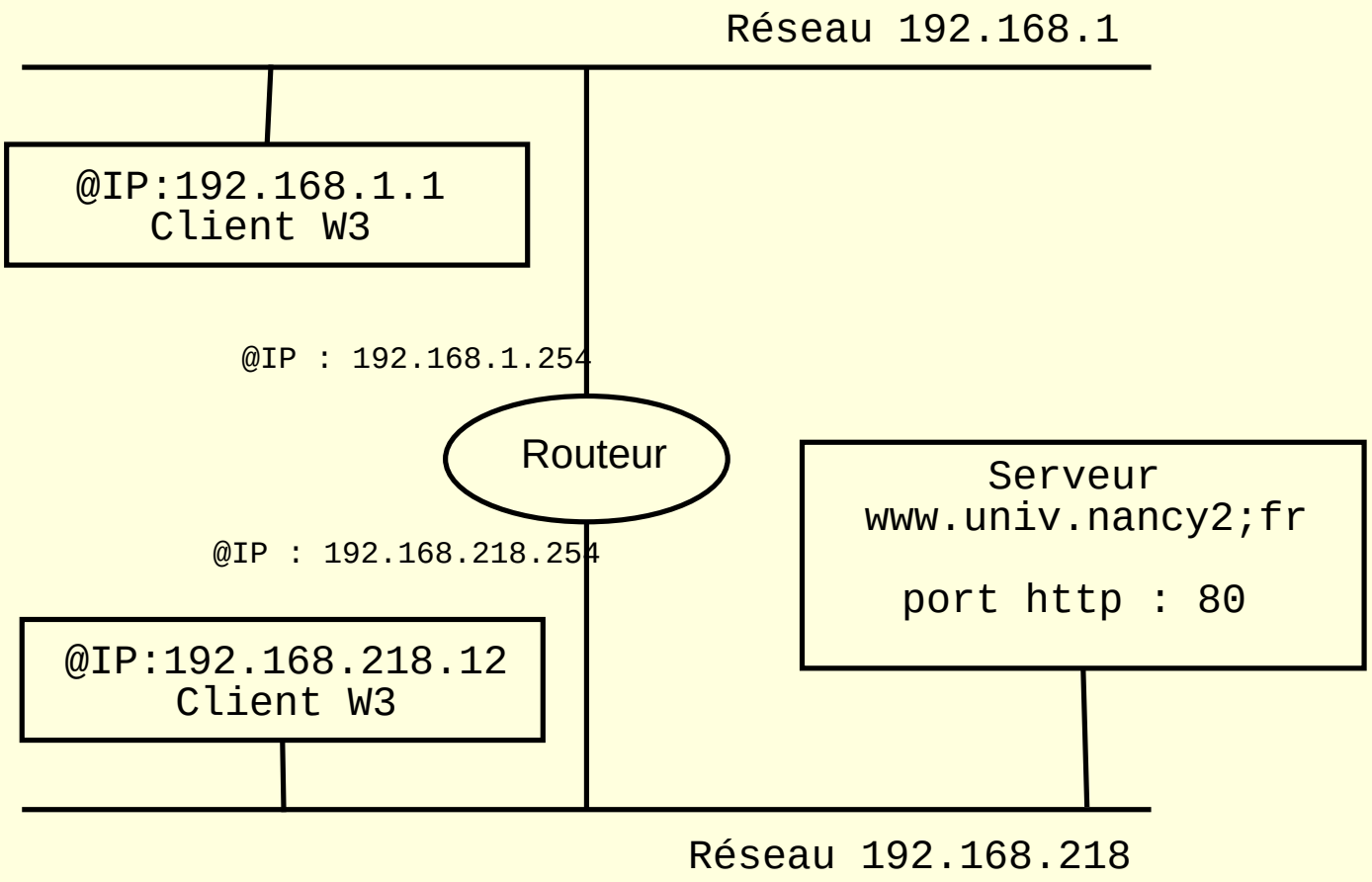
```
HTTP/1.0 200 OK
Date : Fri, 04 Oct 2002 10:38:29 GMT
Server : Apache/1.1.1
Content-type: text/html
Content-length : 327
Last-Modified : Fri, 04 Oct 2002 09:28:12 GMT

[nouvelle page html]
```

4. Les différentes couches réseau

- 1er cas de figure : PC et serveur sur le même sous-réseau :
  - comment navigateur et serveur communiquent - ils via le réseau ?
  - quels sont les couches réseau utilisées ?
  - (pour nous) ce cas se présente si l'on veut afficher la page d'accueil du serveur de l'université : www.univ-nancy2.fr

4.a PC et serveur sur le même sous-réseau (1)



4.a PC et serveur sur le même sous-réseau (2)

- Les machines sur Internet sont repérées par une adresse IP unique.
- Une @IP est un quadruplet de nombres sur 8 bits.
- Les 3 premiers octets réfèrent à l'adresse d'un réseau, et le dernier octet à l'adresse de la machine au sein de ce réseau.
- Pour éviter d'avoir à retenir les adresses IP des machines, on leur associe une chaîne de caractères et le lien entre cette chaîne et l'adresse IP est conservée sur un serveur **DNS**.

4.a PC et serveur sur le même sous-réseau (3)

- La communication entre deux machines via un réseau se fait au travers de différentes ”couches”.

- L’*application* (ici le navigateur) gère un ensemble de couches de niveau supérieur, dans lesquelles un paquet de données P1 correspondant à une requête HTTP est formé, puis passé au système d’exploitation pour que ce dernier l’envoie vers une machine cible.

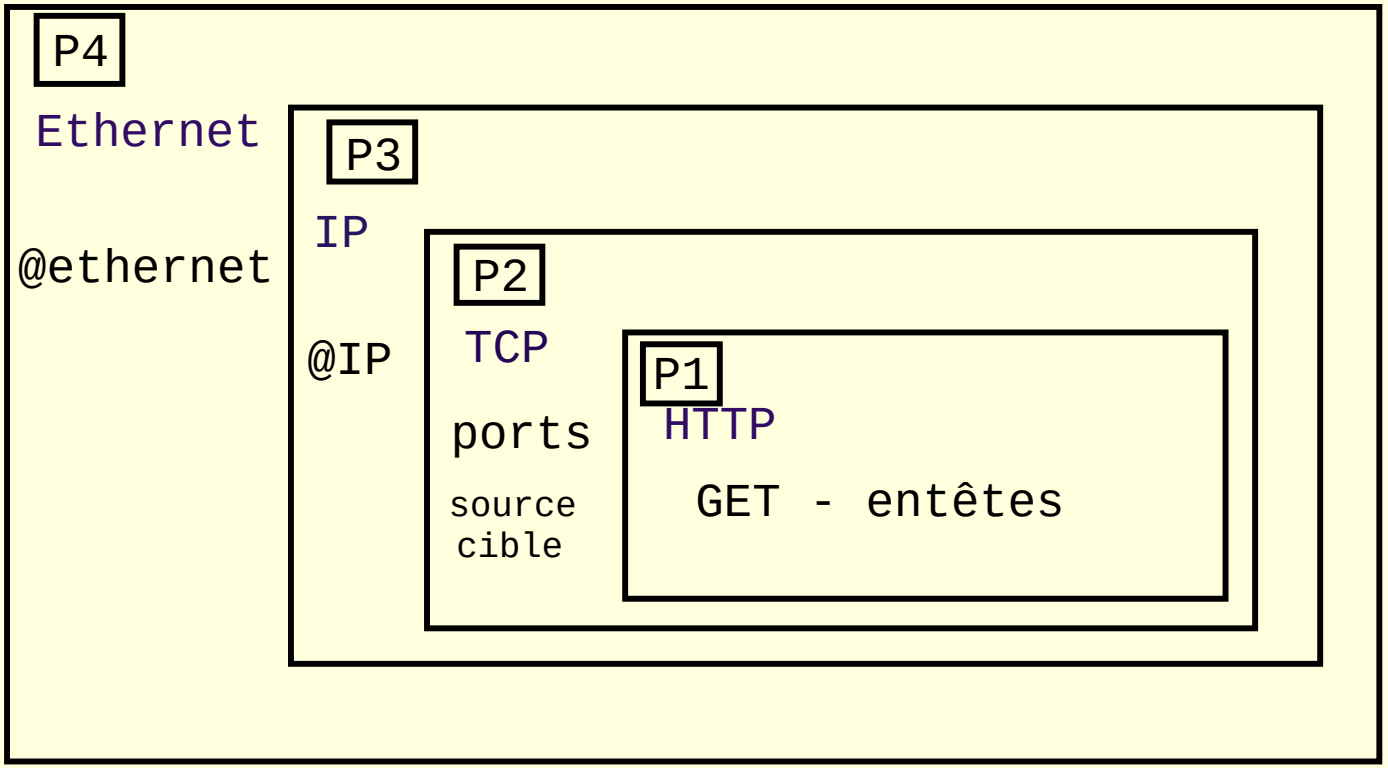
- Le niveau *transport* prenant en charge la gestion des paquets de données (que faire en cas de perte, etc), reçoit ce paquet P1 (pour HTTP, ce niveau correspond au protocole TCP - Transmission Control Protocol -). Pour pouvoir réaliser sa mission ce niveau va ajouter au paquet P1 des informations de contrôle, ce qui donnera le paquet P2.

4.a PC et serveur sur le même sous-réseau (4)

- C’est alors que P2 est transmis à la couche *système*, qui ajoute de nouvelles informations de contrôle, entre autres les adresses IP de la source et de la cible $\rightarrow$ paquet P3.

- Enfin, en bas se trouve le niveau *physique*, assurant l’acheminement du paquet de données via le réseau physique à son destinataire (traitement des adresses Ethernet) $\rightarrow$ paquet P4.

4.a PC et serveur sur le même sous-réseau (5)



4.a PC et serveur sur le même sous-réseau (6)

Que fait la machine cible après réception de P4 ?

- Le processus inverse à lieu, i.e. décapsulations successives des différents paquets par les différentes couches de la machine distante.

- Ainsi, le serveur récupère le paquet originel P1 et l'achemine vers l'application adéquate (Apache dans notre exemple).

- Enfin, le serveur peut traiter la requête qui lui est parvenue, et ensuite envoyer sa réponse via le même procédé.

4.b PC et serveur sur deux sous-réseaux différents (1)

- Le processus de communication est identique, à l'exception de la dernière étape de recherche de l'adresse Ethernet.

- En effet, dans ce cas de figure, on doit passer par un routeur, effectuant la jonction entre deux sous-réseaux.

- Le routeur reçoit un paquet de données, le décapsule et consulte sa table de routage pour réacheminer ce paquet vers le bon sous-réseau.

4.b PC et serveur sur deux sous-réseaux différents (2)

Remarque : Il est possible qu'une machine n'ait pas accès à Internet directement,

- il arrive qu'un pare-feu soit placé à l'entrée d'un sous-réseau pour le protéger d'éventuelles attaques.

- il se peut que l'adresse IP de la machine ne soit pas utilisable sur Internet, car ne faisant pas partie des adresses valides.

On recours alors à un serveur mandataire (proxy), prenant en charge

l'envoie de la requête vers le serveur à la place du client.

4.b PC et serveur sur deux sous-réseaux différents (3)

Les tâches du proxy sont diverses :

- transférer les requêtes des machines du sous-réseau,
- filtrer les accès avec l'extérieur,
- conserver en cache des documents afin d'économiser les requêtes.

Conclusion

Que doit-on savoir faire pour automatiser des requêtes HTTP ?

- savoir programmer une requête simple ou complexe :
  - accéder à un document et à ses liens,
  - télécharger du texte, exclure les images,
- passer par le proxy si besoin, pour envoyer des requêtes à un serveur,
- savoir analyser les entêtes des requêtes et réponses, et le corps des réponses.

Pour cela, nous utiliserons les modules Perl, notamment :

- `LWP : :Simple`
- `HTML : :LinkExtor`