

06/10/25 Cours 2:

L'internet est cassé car la couche de convergence (IP) ne fonctionne pas.

Pour rétablir cela on utilise ~~HTTP~~ UDP

HTTP

Web: se trouve dans la couche application (bas) le formatage est en HTML.

Le web a été créé au début des années 90

Avant y avait ARPANET, FTP

Le web est un Hypertexte distribué

↓
texte avec des liens

il peut mener sur d'autres textes.

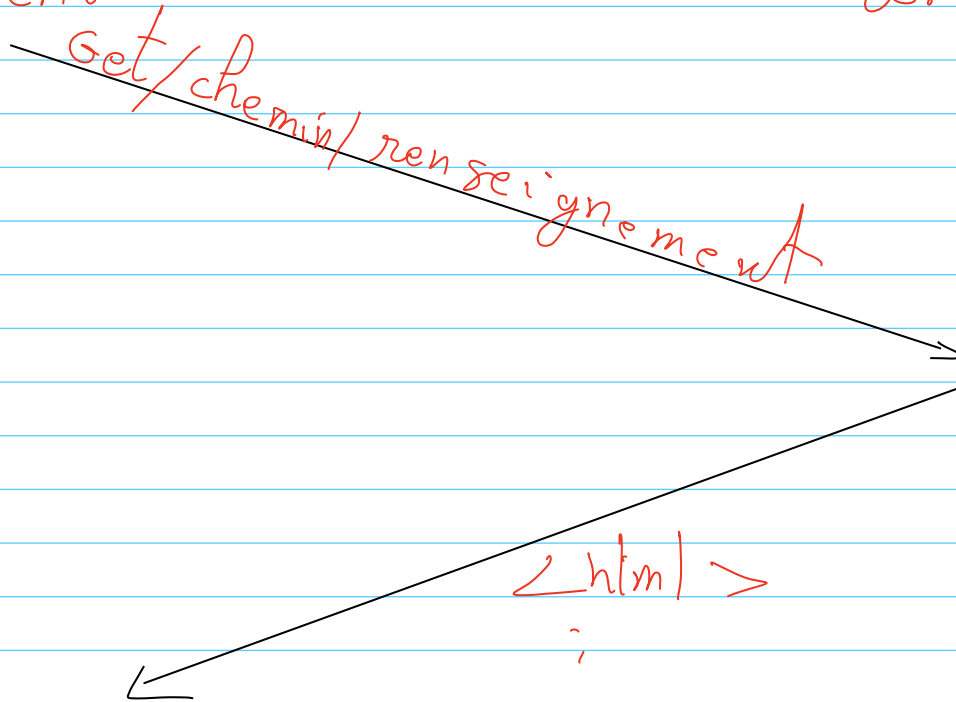
Ce qui rend possible la distribution du web est l'erreur 404

HTTP et HTML

HTTP/0.9 (c'est la version avant la version numérotée)

Client

Serveur



Problème avec ce protocole.

Pb1: il transfère uniquement du HTML

Pb2: la fin du transfert est marquée par le client qui ferme la connexion

L'API socket ne permet pas de distinguer quel host s'est planté. Donc ça rend le transfert non

fiable

HTTP/1.0

peut
contenir
un nbre
important
d'entête

GET HTTP/1.0

⋮

Ligne blanche: la fin

- n° = version
- entête request
- entête response
- " Content-Type
- " " Length
- optionnel

Après on y a introduit plusieurs méthodes
permet de continuer la fiabilité

GET

PUT

POST

corps de requête

idempotentes

non-idempotente

HTTP/1.1: il introduit de nombreux
changement comme:

- streaming fiable (chunked encoding)
- connexions persistantes → plusieurs requêtes sur la même connexion

NB

Contrôle flux: pas envoyer les données

plus vite que serveur ne peut les recevoir

Contrôle de congestion : contrôle flux en remplaçant serveur par le réseau
= d'entête cache-contrôle → dans 3 cons

à seule
diff 1.2,
2 et 3 et 4
le transfert
des requêtes

HTTP/2 : Permet de multiplexer les requêtes sur TCP

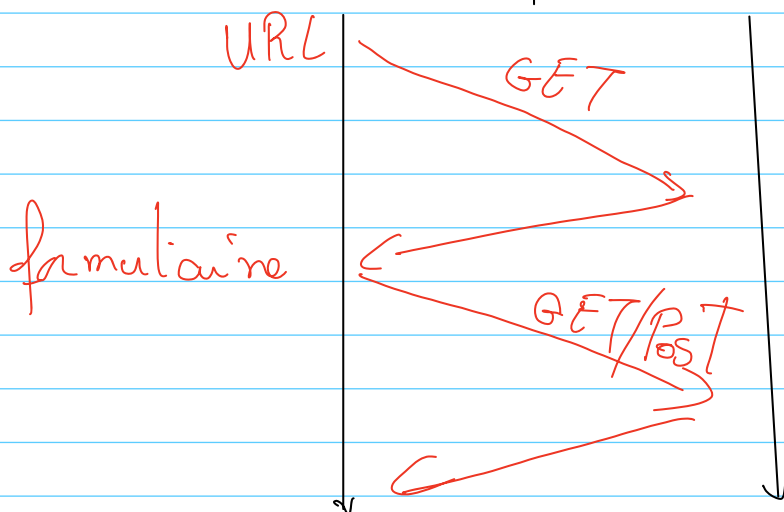
HTTP/3 : → QUIC

Une application ne peut pas être distribuée

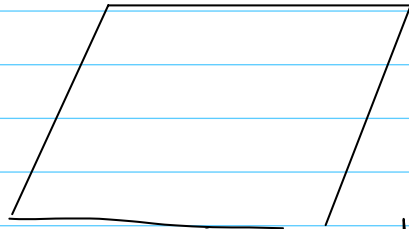
Applications web

C'est une app qui s'exécute sur un navigateur web.

Web "1.0" parfois

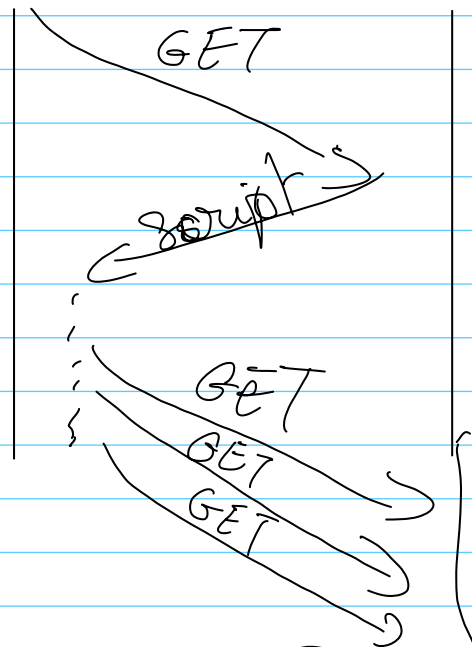


JavaScript pour que le navigateur
soit scriptable.



JS the good bits

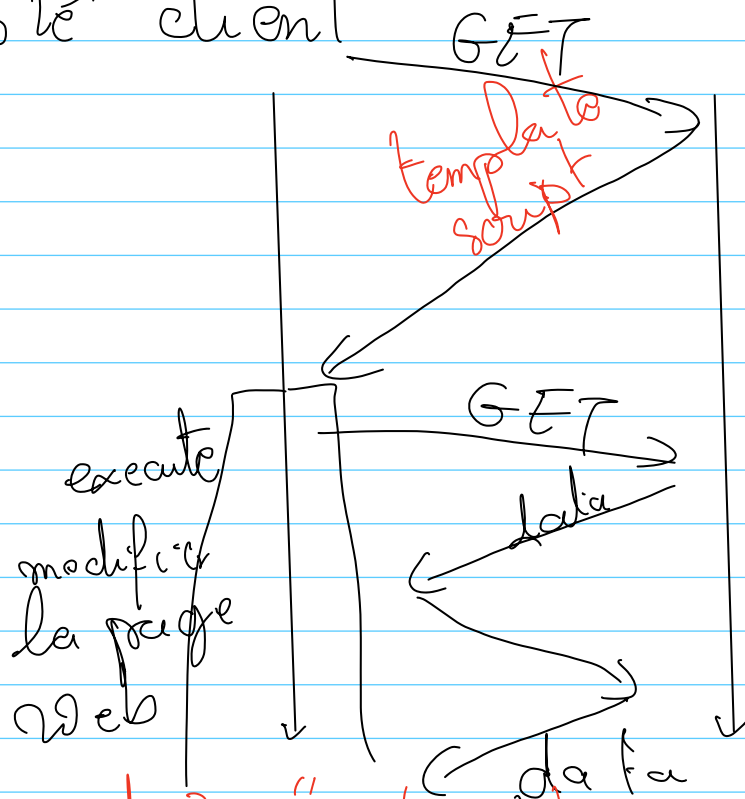
I.E 6 : XMLHttpRequest (Microsoft)
(obsolete)
permet au code javascript de faire
des requêtes HTTP



Un nouveau API a été remplacé
par **fetch**

"Web 2.0" AJAX ← XML
 Asynchrone Javascript

technique de programmation web qui change complètement, qui consiste à faire toute la partie intelligente côté client



"Web 3.0" n'existe pas
"API" web

Protocole { client web → "API" web
 serveur

Donc besoin:

Positive

- serialisation / deserialisation
- stabilité et évolutivité

Negative ou limitation

- structure imposée par HTTP

→ transport ^{request} requête - ^{response} réponse
client - serveur

on peut pas
le faire au
dessus de HTTP

→ ?

TP: app web