Le multimédia dans l'Internet

———————

Architecture de l'Internet

Christophe Deleuze
ActiVia Networks
Christophe.Deleuze@activia.net

DESS Télécom

8 janvier 2002

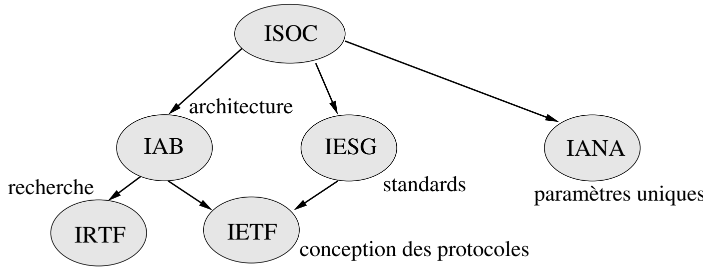
Des processus de standardisation

Qu'est-ce que l'Internet ?

Plusieurs axes de réponse

- des processus de standardisation
- une pile de protocoles
- une interconnexion de réseaux
- un espace de nommage
- un ensemble de technos en constante évolution
- . . .

Les organisations



- Internet Society
- Internet Assigned Numbers Authority
- Internet Architecture Board
- Internet Engineering Steering Group
- Internet Engineering Task Force
- Internet Research Task Force

L'IETF

- ouverte à tous les individus
- travail en *Working Groups* groupés en 8 domaines *(areas)*
- discussions par *mailing-lists*
- réunions *(IETF meetings)* 3 fois par an
- propriété intellectuelle
  - ✓ pas de confidentialité
  - ✓ indication des droits
  - ✓ si possible, "termes raisonnables et non discriminatoires"

Les documents de l'IETF

- *Internet Drafts* (ID)
- *Request For Comments* (RFC)
  - ✓ *Informational*
  - ✓ *Experimental*
  - ✓ *Historic*
  - ✓ *Standards Track*
    - ☞ *Proposed Standard*
    - ☞ *Draft Standard*
    - ☞ *Internet Standard*

Les standards Internet

- *The Internet Standards Process* - RFC 2026
- une spécification
  - ✓ stable et bien comprise
  - ✓ techniquement satisfaisante
  - ✓ plusieurs implémentations indépendantes et interopérables
  - ✓ soutien public significatif
  - ✓ utile dans (une partie de) l'Internet

"rough consensus and running code"

Le W3C

- *World Wide Web Consortium*
- fondé en 1994 (Tim Berners-Lee)
- *"promoting interoperability and encouraging an open forum for discussion..."*
- *"... leading the technical evolution of the web"*
- *recommandations*
  - ✓ working draft
  - ✓ last call WD
  - ✓ candidate recommandation
  - ✓ proposed recommandation
  - ✓ recommandation

L'esprit Internet

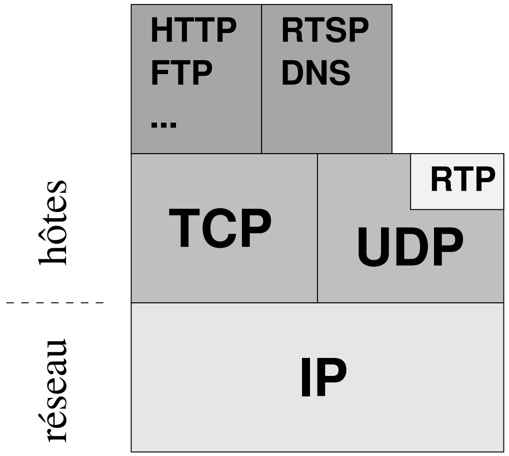
- ouverture
  - ✓ des protocoles
  - ✓ des processus
- logiciels libres
  - ✓ pour l'Internet
  - ✓ par l'Internet

Une pile de protocoles

La pile de protocoles de l'Internet

Couches :

- application
- transport
- réseau



Couche réseau : IP

- commutation de datagrammes
- *best effort*
  - ✓ pertes
  - ✓ déséquencement
  - ✓ altération
- ...
- IP multicast

Couche transport : UDP

transport non fiable de messages

- (dé)multiplexage
- somme de contrôle (optionnelle)

➠ très rudimentaire

Couche transport : TCP

transport fiable d'un flux d'octets

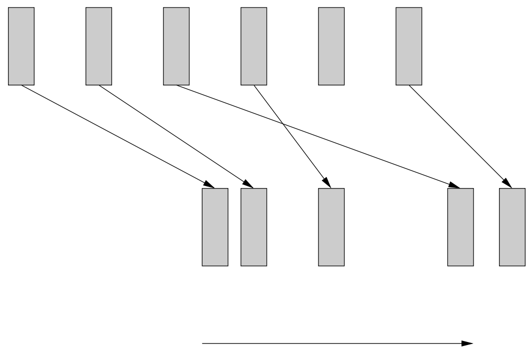
- segments
- n^o de séquences
- ack + timeout + retransmission
- somme de contrôle
- contrôle de flux
- contrôle de congestion *(slow start - congestion avoidance)*

➠ inadapté aux flux multimédias

Besoins des applications multimédias

- le réseau crée :
  - ✓ pertes
  - ✓ gigue *(jitter)*
  - ✓ déséquencements
- ➠ *playout buffer*
- besoin de
  - ✓ numéros de séquence
  - ✓ estampilles
  - ✓ synchronisation
- pour les pertes
  - ✓ *Forward error correction* (FEC)

✓ applications adaptatives



Couche transport : RTP/RTCP

- *Real-time Transport Protocol* ('96)
- modèle = conférence
- services
  - ✓ identification de contenu
  - ✓ numérotation en séquence
  - ✓ estampilles
  - ✓ *delivery monitoring* (RTCP)
- RTP + UDP = service de transport
  - ✓ UDP fournit multiplexage
- n'inclut rien pour
  - ✓ *timely delivery*
  - ✓ garanties de QoS

RTCP

- *RTP Control Protocol*
- une session RTCP associée à chaque flot RTP
- émis par chaque participant à destination de tous les autres
  - ✓ synchro inter flux
  - ✓ statistiques de réception
  - ✓ *membership*
- fréquence d'émission

$$f = \frac{y}{nb.participants}$$

  - ✓ bp RTCP $<$ x % bp RTP

RTCP suite...

paquets RTCP

SR *sender reports*

RR *receiver reports*

SDES *source description*

BYE départ explicite

APP extensions spécifiques à une application

RTP/RTCP suite...

- RTP est en fait un protocole générique *("protocol framework")*
  - ✓ *RTP profiles* ➠ AVP
  - ✓ *RTP payloads*
- mixeur *(mixer)*
  - ✓ combine plusieurs flots RTP
  - ✓ peut changer l'encodage
- translateur *(translator)*
  - ✓ encodage peut être changé
  - ✓ l'identité des flots persiste
  - ✓ ex. traversée de *firewall*

Implémentation RTP

bibliothèque RTPlib [Lab01]

- RTPCreate() – RTPDestroy()
- RTPOpenConnection() – RTPCloseConnection()
- RTPSend() – RTPReceive()
- RTPGetRTPPacket()
- ...
- fcts de paramétrage
- accès aux stats RTCP/aux infos de participants

- simple à utiliser
- 81 fonctions...
- 13000 lignes de code

Couche application : HTTP

- *...distributed collaborative hypermedia information systems*
- URI = URL ou URN
- requête/réponse
  - ✓ requête
    - ☞ méthode + URI + version protocole
    - ☞ message type MIME (mod. requête + info sur le client [+ contenu])
  - ✓ réponse
    - ☞ ligne de status
    - ☞ message type MIME (info sur le serveur + information sur l'entité (en-tête) [+ corps de l'entité])

Methodes HTTP

OPTIONS capacités du serveur/contraintes associées à une ressource

* **GET** récupère une entité

* **HEAD** GET sans le corps de l'entité

POST associe un objet à une entité

PUT ajoute une entité

DELETE retire une entité

TRACE équivalent de `ping -R`

CONNECT réservé

HTTP : éléments intermédiaires

- importance des éléments intermédiaires
  - ✓ proxy
    - ☞ transparent ou pas
  - ✓ gateway
    - ☞ prétend être le serveur
  - ✓ tunnel
    - ☞ relai “aveugle”
- directives de *cache control*
  - ✓ en requête
  - ✓ en réponse

HTTP : exemple

```
GET /%7Edeleuze/ HTTP/1.0
User-Agent: Wget/1.5.3
Host: rp.lip6.fr:80
Accept: */*


HTTP/1.0 200 OK
Date: Thu, 20 Dec 2001 10:07:24 GMT
Server: Apache/1.3.19 (Unix) mod_ssl/2.8.1 OpenSSL/0.9.5a PHP/4.0.5
Last-Modified: Mon, 09 Jul 2001 11:20:30 GMT
ETag: "c11b2-520-3b49937e"
Accept-Ranges: bytes
Content-Length: 1312
Content-Type: text/html
X-Cache: MISS from firebolt.local
Connection: close

Data (1123 bytes)
```

Couche application : RTSP

Real-Time Streaming Protocol
"télécommande" pour des serveurs multimédia

- variation sur HTTP
  - ✓ code client et serveur existants
  - ✓ techno de proxy et authentification
- différent de HTTP
  - ✓ état dans le serveur
  - ✓ canaux de données et de contrôle séparés
  - ✓ serveur peut émettre des requêtes

RTSP : présentations

notion de **présentation**

- URI `rtsp://media.example.com:554/twister`
- ensemble de composants
  `rtsp://media.example.com:554/twister/audiotrack`
- description
  - ✓ *Session Description Protocol* (SDP)
  - ✓ information sur les composants
  - ✓ encodages
  - ✓ ...

RTSP : gestion du temps

- NPT *(Normal play time)*
  - ✓ `h:m:s.ss` `10:07:00-10:07:33`
  - ✓ `s.ss` `121.45-125`
  - ✓ affiché par le lecteur
- SMPTE
  - ✓ `h:m:s:trame.sstr` `10:07:00-10:07:33:05.01`
  - ✓ édition du média
- temps absolu
  - ✓ ISO 8601 UTC/GMT
  - ✓ 20010829T114501.00Z
  - ✓ synchronisation de serveurs

Méthodes RTSP

method	direction	object	requirement
DESCRIBE	C->S	P,S	recommended
ANNOUNCE	C->S, S->C	P,S	optional
SETUP	C->S	S	required
PLAY	C->S	P,S	required
PAUSE	C->S	P,S	recommended
RECORD	C->S	P,S	optional
TEARDOWN	C->S	P,S	required
OPTIONS	C->S, S->C	P,S	required (S->C: optional)
REDIRECT	S->C	P,S	optional
GET_PARAMETER	C->S, S->C	P,S	optional
SET_PARAMETER	C->S, S->C	P,S	optional

Exemple RTSP 1 – description

```
GET /twister.sdp HTTP/1.1
Host: www.example.com
Accept: application/sdp


HTTP/1.0 200 OK
Content-Type: application/sdp

v=0
o=-2890844526 2890842807 IN IP4 192.16.24.202
s=RTSP Session
m=audio 0 RTP/AVP 0
a=control:rstp://audio.example.com/twister/audio.en
m=video 0 RTP/AVP 31
a=control:rtsp://video.example.com/twister/video
```


Exemple RTSP 2 – prépare

```
SETUP rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 1
Transport: RTP/AVP/UDP;unicast;client_port=3056-3057

RTSP/1.0 200 OK
CSeq: 1
Session: 12345678
Transport: RTP/AVP/UDP;unicast;client_port=3056-3057;
   server_port=5000-5001

SETUP rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 1
Transport: RTP/AVP/UDP;unicast;client_port=3058-3059

RTSP/1.0 200 OK
CSeq: 1
Session: 23456789
```


Exemple RTSP 3 – joue

```
PLAY rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 2
Session: 23456789
Range: smpte=0:10:00-

RTSP/1.0 200 OK
CSeq: 2
Session: 23456789
Range: smpte=0:10:00-0:20:00
RTP-Info: url=rtsp://video.example.com/twister/video;
 seq=12312232;rtptime=78712811
```


Exemple RTSP 4 – joue (suite)

```
PLAY rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 2
Session: 12345678
Range: smpte=0:10:00-

RTSP/1.0 200 OK
CSeq: 2
Session: 12345678
Range: smpte=0:10:00-0:20:00
RTP-Info: url=rtsp://audio.example.com/twister/audio.en;
 seq=876655;rtptime=1032181
```

Exemple RTSP 5 – termine

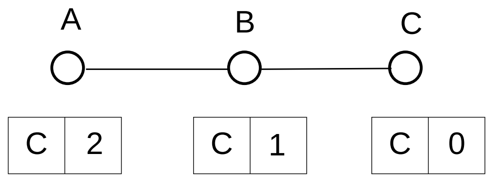
```
TEARDOWN rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 3
Session: 12345678

RTSP/1.0 200 OK
CSeq: 3

TEARDOWN rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 3
Session: 23456789

RTSP/1.0 200 OK
CSeq: 3
```


Une interconnexion de réseaux

Routage et relayage

- relayage *(forwarding)*
  - ✓ envoyer au prochain nœud vers la destination
  - ✓ circuit virtuel : id connection + table de commutation
  - ✓ datagramme : adr. dest + table de relayage
  - ✓ routage à la source : liste des nœuds contenue dans le paquet
- routage
  - ✓ rassembler l'info nécessaire au calcul des routes
    - ☞ statique
    - ☞ vecteurs de distance
    - ☞ état des liens

Routage : vecteurs de distance



- vect. de dist. vers toutes les destinations
- informe ses voisins
- et voilà !
- RIP
- + simple à implémenter
- – convergence lente (comptage à l'infini)

Routage : état des liens

- chaque routeur connait l'état (le coût) des liens vers ses voisins
- il *inonde* cette information
- tous les routeurs construisent une carte du réseau
- algorithme du plus court chemin (Dijkstra)
- IS-IS, OSPF

- + convergence rapide
- + souplesse
- – procédure d'inondation complexe

Routage hiérarchique

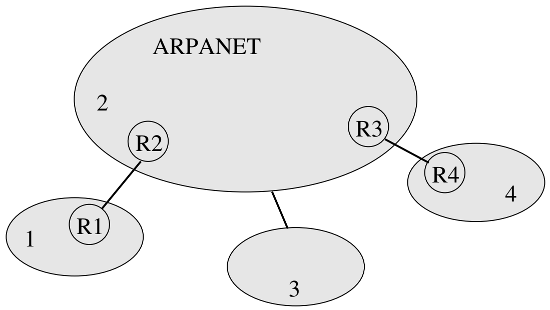
- croissance de l'Internet
  - ✓ nb de routeurs
  - ✓ nb de réseaux
  - ✓ fréquence des changements
  - ✓ déployer de nouvelles versions
- hiérarchiser ➠ systèmes autonomes
  - ✓ intra-domaine : dans un AS
  - ✓ inter-domaine : entre les AS

Systèmes autonomes (AS)

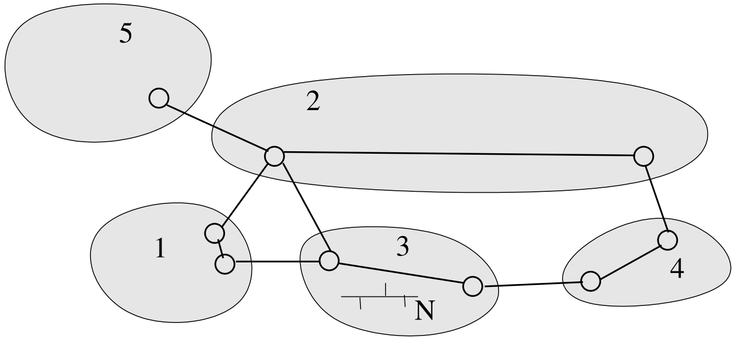
- *"un ensemble de routeurs et de réseaux sous la même administration"*
- techniquement ➠ complètement connecté
- *Interior Gateway Protocol* (IGP)
- numéro d'AS sur 16 bits
- routage inter-domaine
  - ✓ statique
  - ✓ EGP
  - ✓ BGP

EGP

- *Exterior Gateway Protocol*
- EGP = distance vector
- métrique = accessibilité
- hypothèse : cœur avec hiérarchie à deux niveaux



Topologie plus complexe



- EGP pas possible
- ➠ BGP

BGP

- *Border Gateway Protocol*
- vecteurs de chemins ➠ pas de comptage à l'infini
- *policies* ➠ f(destination, annonceur) = rang
- confédération = groupe d'AS
  - ✓ hiérarchie
  - ✓ potiliques communes
- –
  - ✓ bcp de configuration
  - ✓ incohérence des politiques

Un espace de nommage

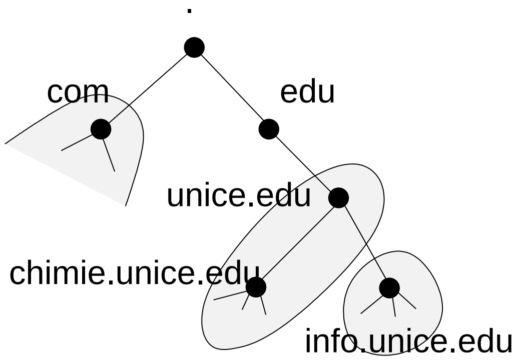
Avant le DNS...

- fichier `hosts.txt`
- copie authoritative gérée par le NIC au SRI
- copié par `ftp` toutes les nuits
- problèmes d'échelle
  - ✓ distribution
  - ✓ mise à jour
  - ✓ coordination

- besoin d'une architecture décentralisée

Base de données du DNS

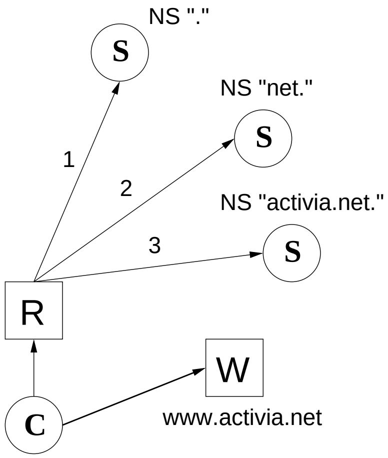
- espace de nommage hiérarchique
- noms = clés de la BD
  - ✓ `nom TTL A adresse`
  - ✓ `nom TTL NS nomdeserveur`
  - ✓ …
- zones
  - ✓ délégation



Composants du DNS

- serveurs
- résolveurs
- résolveurs “stub”
  - ✓ `libresolv` ➠ `gethostbyname()`
- réplication
  - ✓ serveurs primaire et secondaires
  - ✓ transfert de zone
- protocole applicatif (sur UDP ou TCP)

Algorithme de résolution



Problèmes

- utilisation non standard
  - ✓ dans les CDN
  - ✓ pour l’équilibrage de charge
- nouveaux RR difficiles à déployer
- sécurité
- espaces DNS alternatifs
  - ✓ superoot
  - ✓ …
- complexité (50 RFC)

…

Un ensemble de technologies en constante évolution

Évolution “voulue”

conçue et planifiée par l’IETF

- *classless inter-domain routing (CIDR)* 1993
- IPv6
  - ✓ adresses 128 bits
  - ✓ autoconfiguration
  - ✓ ...
- QoS
- multicast
- mobilité
- ...

Évolution “de fait”

conçue et planifiée par... personne !

- perte de la transparence
  - ✓ *Network address translation* (NAT)
  - ✓ *firewalls*
  - ✓ proxies transparents
- applis *peer to peer*
- les CDN

Références

[Bra96] S. Bradner. *The Internet Standards Process – Revision 3.* RFC 2026, IETF, octobre 1996.

[Har01] S. Harris. *The Tao of the IETF - A Novice Guide to the Internet Engineering Task Force.* RFC 3160, IETF, août 2001.

[HPC95] C. Huitema, J. Postel et S. Crocker. *Not All RFCs are Standards.* RFC 1796, IETF, avril 1995.

[IET01] IETF. *Audio Video Transport (AVT) working group.* Web page, 2001. http://www.ietf.org/html.charters/avt-charter.html.

[Lab01] Internet Real-Time Lab. *RTP Library.* Web page, 2001. http://www.cs.columbia.edu/IRT/software/rtplib/.

[Sal01] Andras Salamon. *DNS Resource Directory.* Web page, 2001. http://www.dns.net/dnsrd.

[Sch01a] Henning Schulzrinne. *RTP home page.* Web page, 2001. http://www.cs.columbia.edu/~hgs/rtp/.

[Sch01b] Henning Schulzrinne. *RTSP home page.* Web page, 2001. http://www.cs.columbia.edu/~hgs/rtsp/.

Fin

Document réalisé avec LaTeX.
Généré le 8 janvier 2002.
Style de document `foils`.
Dessins réalisés avec `xfig`.

Christophe Deleuze, `Christophe.Deleuze@activia.net`
`http://www.activia.net`

Ce document est disponible en postscript compressé avec gzip à
`http://rp.lip6.fr/~deleuze/activia.html`