La distribution de contenu dans l'Internet
3 – Content Delivery Networks (CDN)

Christophe Deleuze

Christophe.Deleuze@free.fr

ENST Paris

janvier 2004

Contexte

- infrastructure de caches
  - ✓ indispensable
  - ✓ gérés par les opérateurs
  - ✓ peu soucieux des fournisseurs
- fournisseurs de contenu
  - ✓ utilisent le web
  - ✓ gros besoins de distribution

Les fournisseurs de contenu

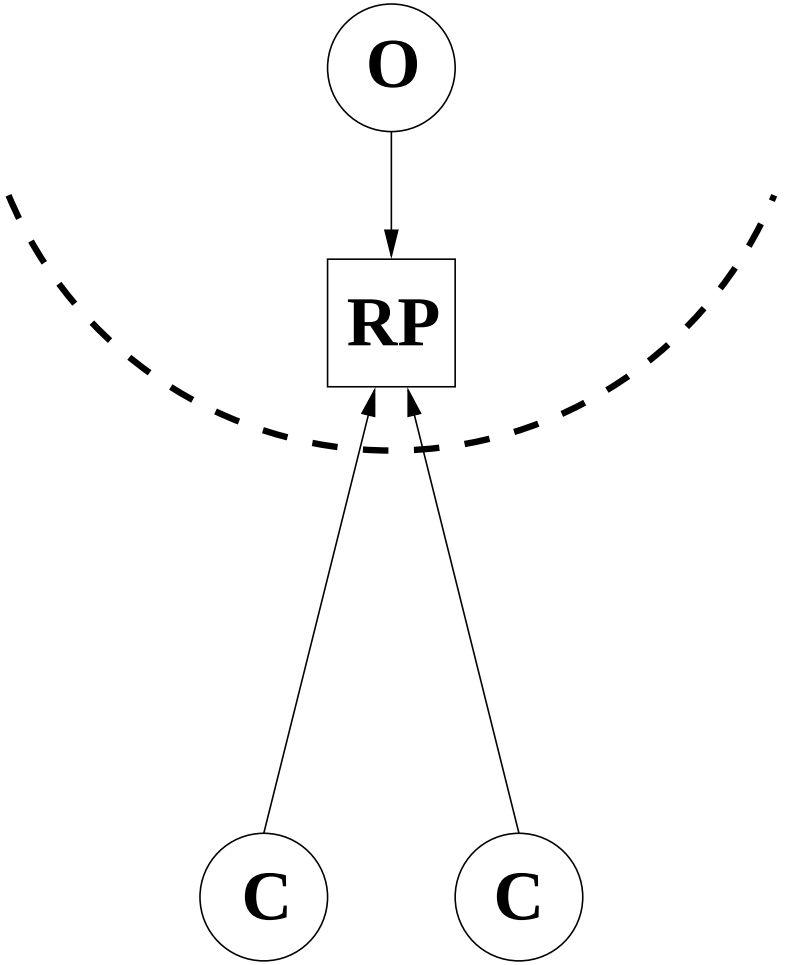
veulent :

- rendre contenu accessible
  - ✓ capacité serveur
  - ✓ capacité réseau
- garder le contrôle
  - ✓ fraîcheur
  - ✓ comptage etc
  - ✓ web dynamique
  - ✓ gestion de site

Systèmes pour les fournisseurs

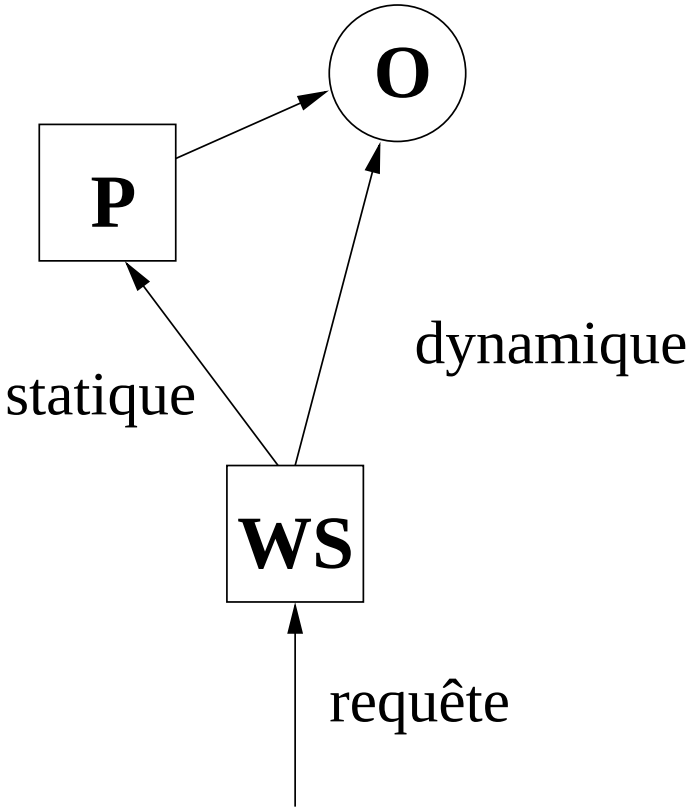
Accélérateur web (reverse proxy)

- origine caché derrière un proxy
- géré par le fournisseur de contenu
  - ✓ + charge origine
- variante : *web switch/server farm*



Équilibrage de charge

- L4 switch (IP src)
- L5-7 switch (web switch)
  - ✓ ex. routage req stat/dyn
  - ✓ monitoring
- Cisco Distributed Director
- Radware Cache Server Director



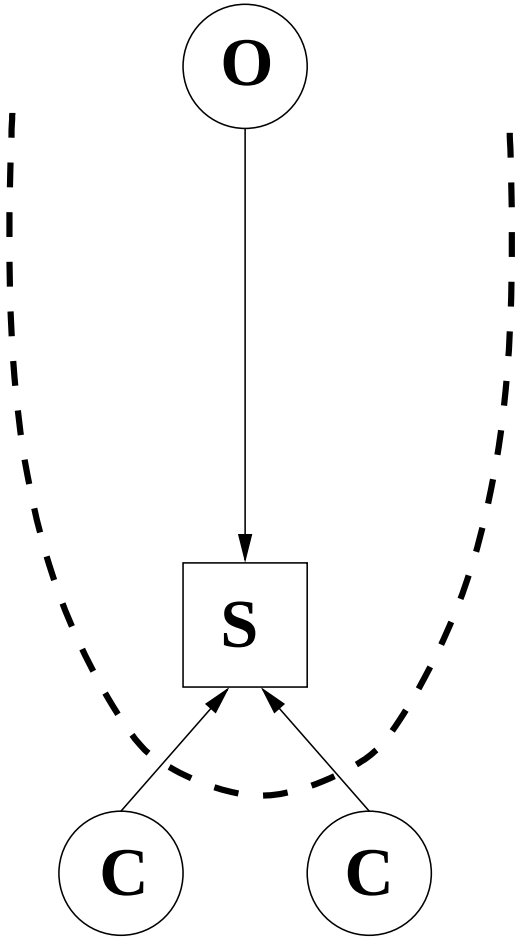
Miroirs

- un site principal
- des sites miroirs
- inconvénients
  - ✓ synchronisation généralement périodique
  - ✓ sélection du miroir + ou – manuelle
    - ☞ CPAN sauve le choix dans un cookie puis HTTP redirect
  - ✓ infrastructure
  - ✓ statique

Content Delivery Network

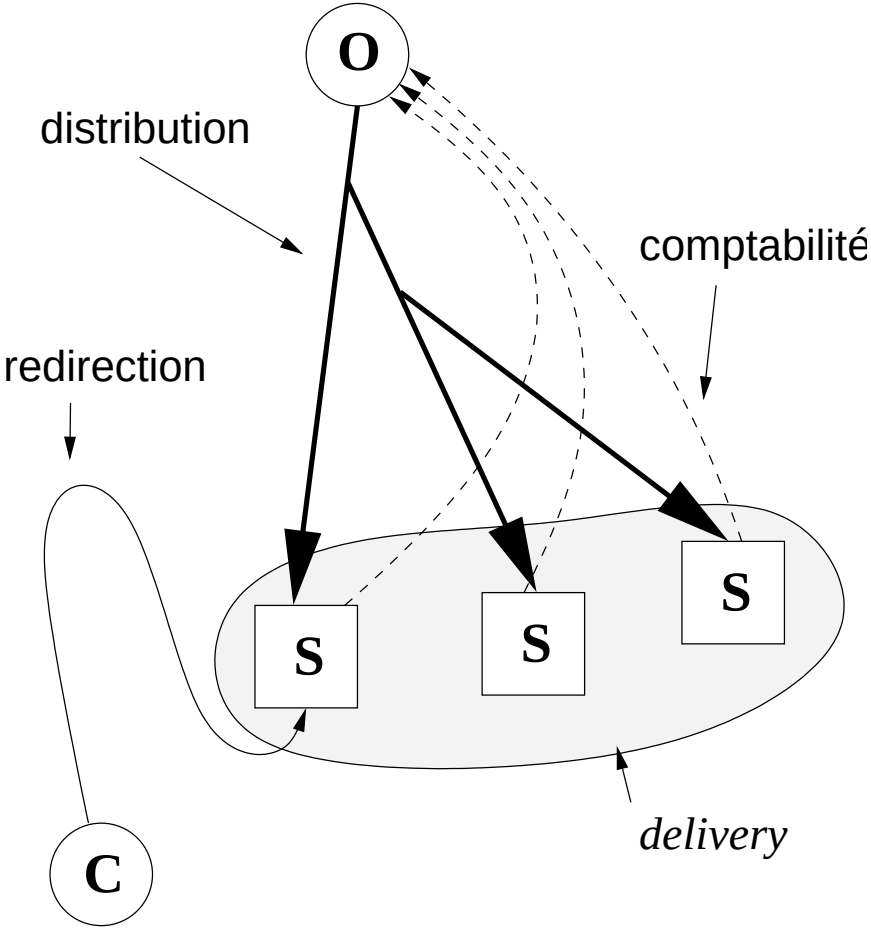
surrogate placé près des clients géré par le fournisseur de contenu

- + charge origine
- + débit réseau
- + délai client
- + stats origine
- + fraicheur



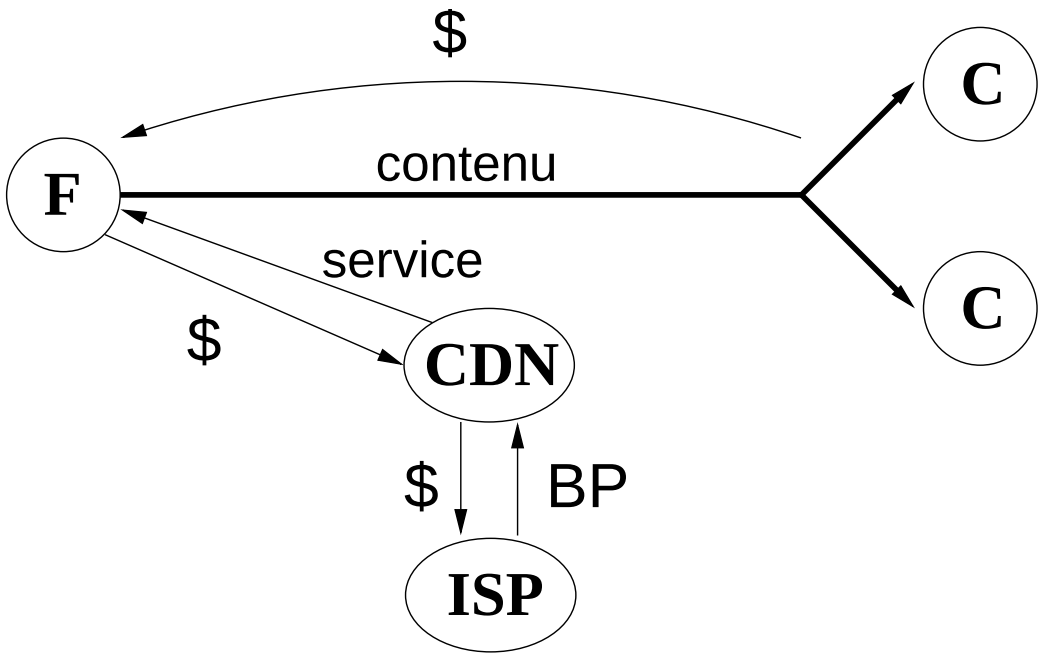
CDN - systèmes

- rapprocher le contenu des clients
- quatre systèmes
  - ✓ *delivery*
    - ☞ serveurs *surrogates*
  - ✓ redirection
  - ✓ distribution
    - ☞ réplication
    - ☞ routage du *live*
  - ✓ comptabilité



Qu'est-ce qu'un CDN ?

- fournisseur de contenu
- consommateurs
- opérateur réseau
- opérateur CDN



Opérateurs de CDN

- Akamai : 13500 serveurs
- Mirror Image
- Digital Island
- ...
- services “connexes” de gestion de contenu
  - ✓ monitoring
  - ✓ reporting
  - ✓ DRM
  - ✓ identification des utilisateurs
    - ☞ spécialiser/segmenter les contenus

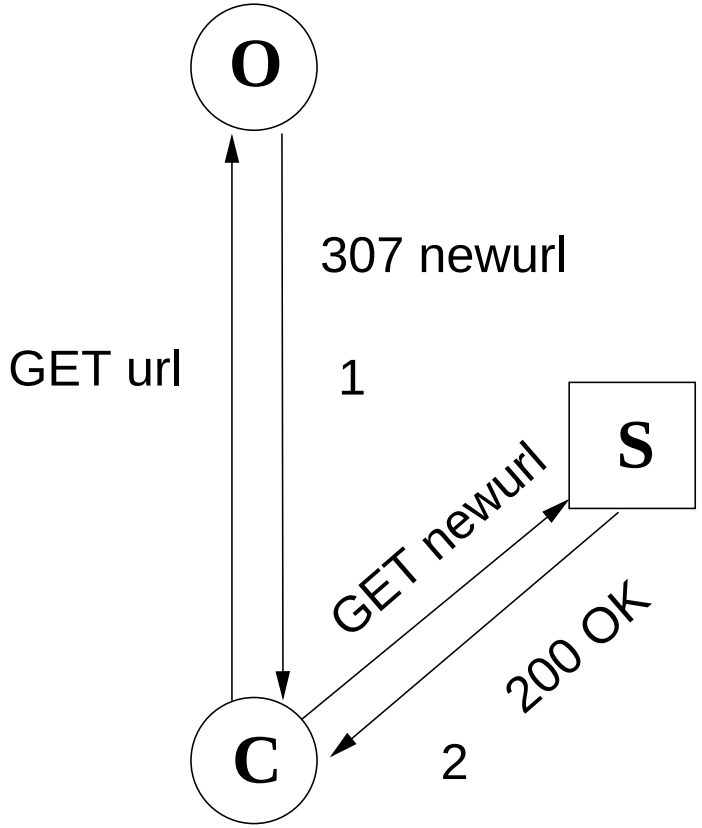
Redirection

Redirection (request routing)

- trouver le “meilleur” surrogate (équilibrage de charge + distance au client)
- surrogate = f(consommateur, contenu, état réseau, états surrogates)
- routage de requête
  - ✓ routage (construction des tables)
  - ✓ relayage *(forwarding)*
- transparent pour le client...
  - ✓ différentes méthodes pour le relayage

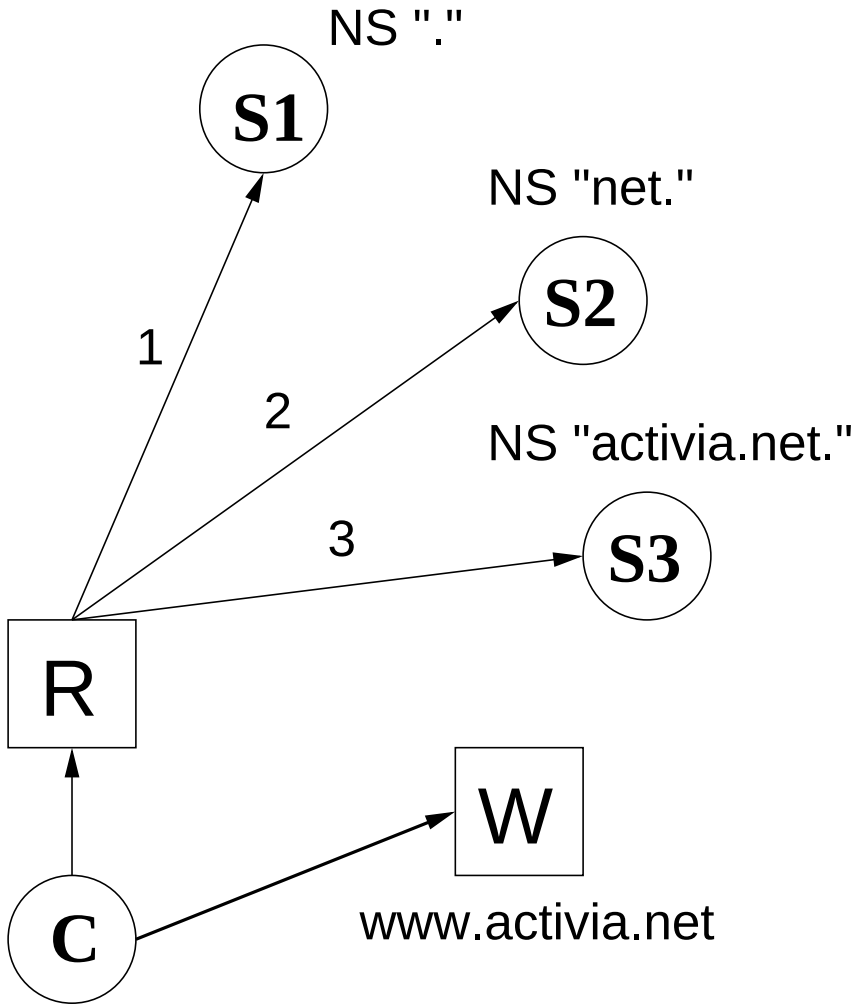
Redirection : application redirect

- ex HTTP : 307 temporary redirect
- \+ précis
  - ✓ adresse client
  - ✓ URL contenu
- – connexion vers l’origine
- – à implémenter pour chaque type de contenu
- – l’URL du document change !



Rappels sur le DNS

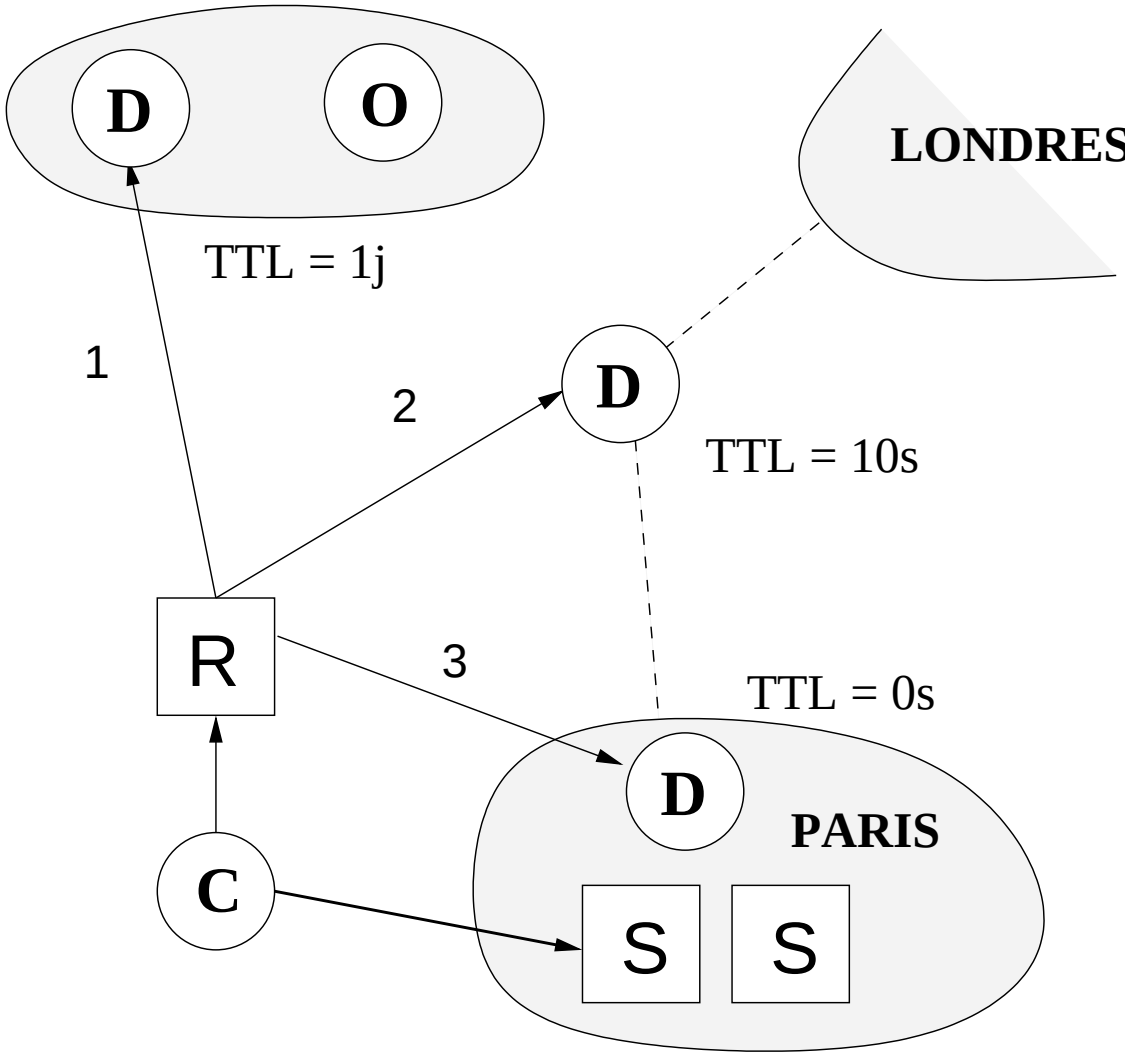
- BD distribuée et redondante
- associations (nom, adr. IP)
- espace de noms hiérarchique
- serveur
- *resolver (cache server)*
- `CNAME` (alias)



Redirection : DNS

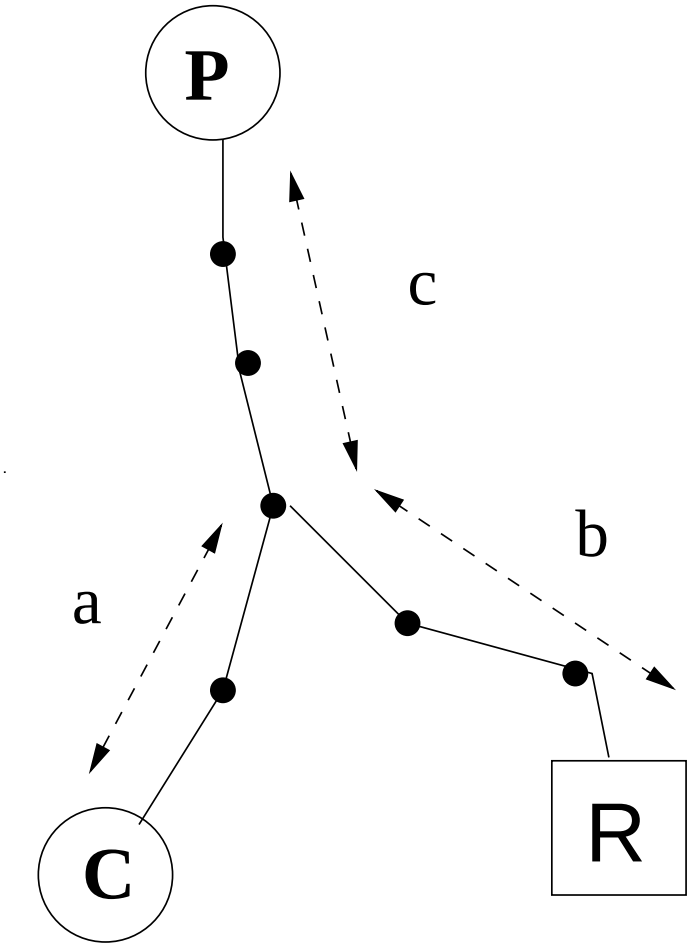
- répondre selon le client
- + élégant
- + scalable, robuste (sécurisé ?)
- – précision
  - ✓ adresse du *resolver* du client
  - ✓ contenu : nom de domaine
    - ☞ `http://www.example.com/pic.gif`
    - ☞ possibilité de conventions (www.* = HTTP port 80)
    - ☞ reécriture des URL
      ➠ `http://gif.example.com/pic.gif`
- – *DNS accelerator*

Redirection DNS hiérarchique



Résolveurs bien placés ?

- parfois NON
- dist. depuis l'intérieur du réseau *(traceroute divergence)*
- $d = max(a, b)$
  - ✓ médiane : 5 (ou 8)
  - ✓ 30 % > 8 !
- $r = c/d$
  - ✓ 35 % ou 5 % < 0.5
  - ✓ moins de 10 % > 2.0



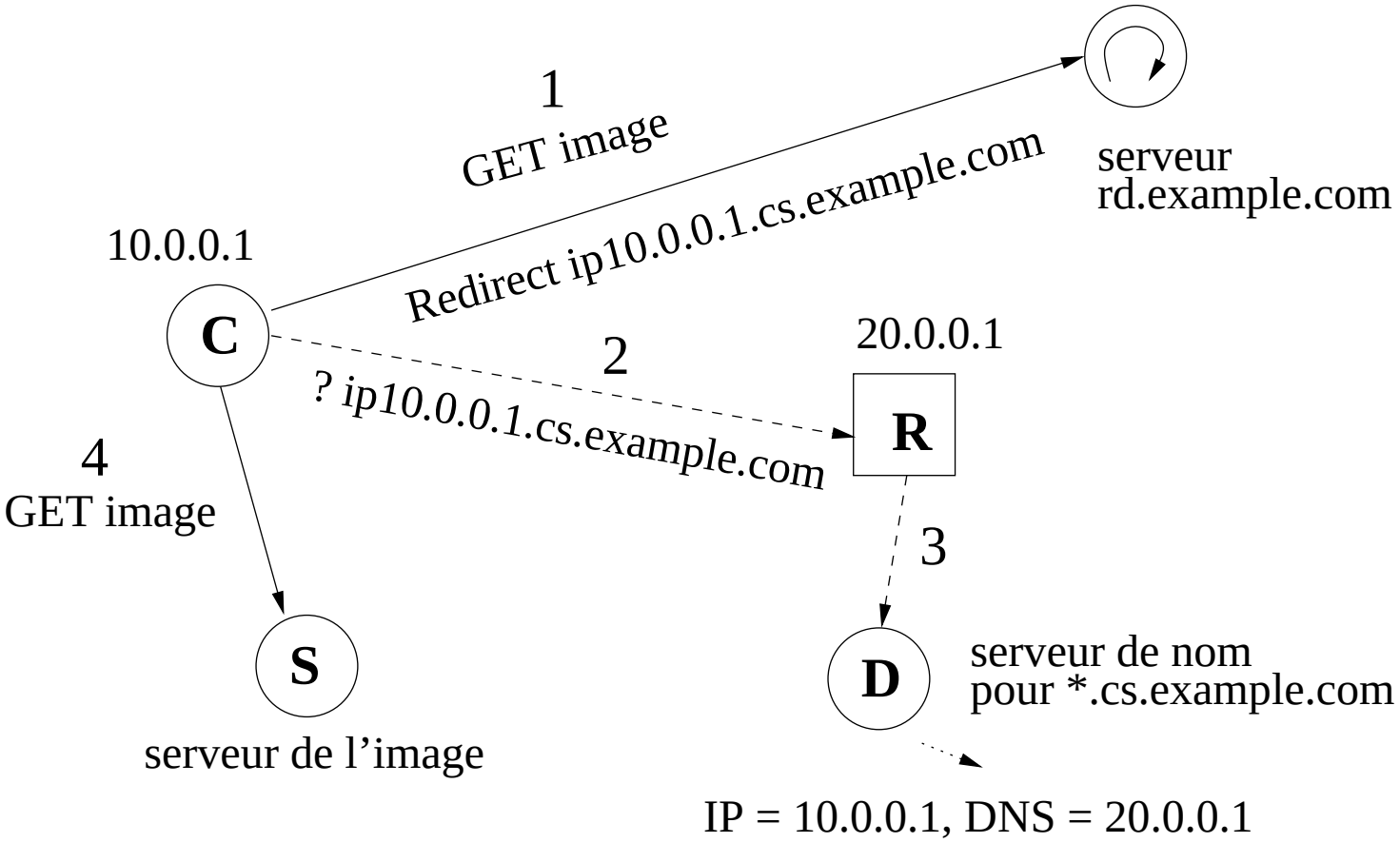
Résolveurs

- définition de la proximité
  1. AS cluster
  2. network cluster (analyse tables BGP)
  3. *traceroute divergence*
  4. corrélation RTT
- 1 à 3 topologie
- 4 performance

Méthode de mesure

- lien sur un GIF de 1 pixel transparent

```
<img src="http://rd.example.com/tr.gif" height=1 width=1>
```



Résultats

mesurés

	req	client
AS	69	64
network	24	16

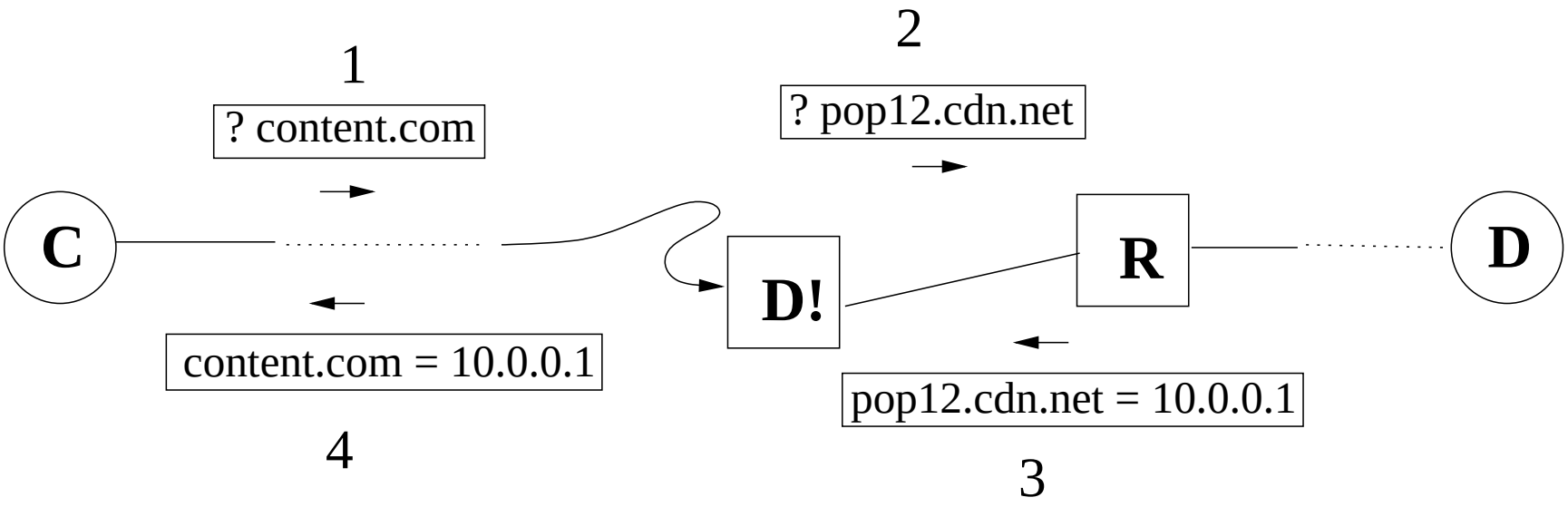
meilleurs possibles

	req	client
AS	92	88
network	70	66

- impact faible pour CDN actuels
- fortes améliorations possibles (et nécessaires) si densification des CDN

Réécriture DNS

- interception DNS réécrit la req.
- pb : très intrusif
- D = DNS standard



Réécriture HTTP

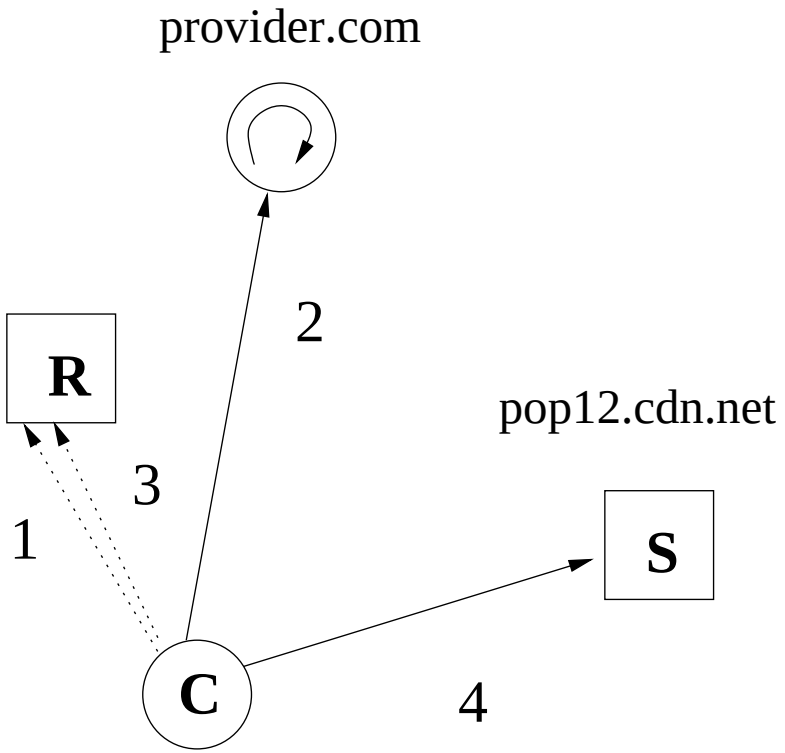
- amorçage par HTTP redir

⇒ `GET /content.html HTTP/1.1`
`Host: provider.com`

⇐ `HTTP/1.1 307 Temp. Redirect`
`Location: http://pop12.cdn.net/provider.com/content.html`

⇒ `GET /provider.com/content.html`
`Host: pop12.cdn.net`

- puis réécriture des liens
  - ✓ type SSI
  - ✓ machine en reverse proxy...



DNS : pbs avec le TTL

- charge serveurs DNS ➠ non
- trafic (5 % en ’92)
- délais

contenu du cache DNS	**délai médian**
réponse	2,3 ms
addr. du serv. DNS	60 ms
adr. root et .com	200 ms

dernier cas : 25 % > 3 s !

DNS : pb des objets embarqués

la “page web moyenne”

tps de charg. total	6,3 s
taille totale	30,9 ko
taille moy. objets	1,22 ko
tps de charg. moy. objets	0,415 s

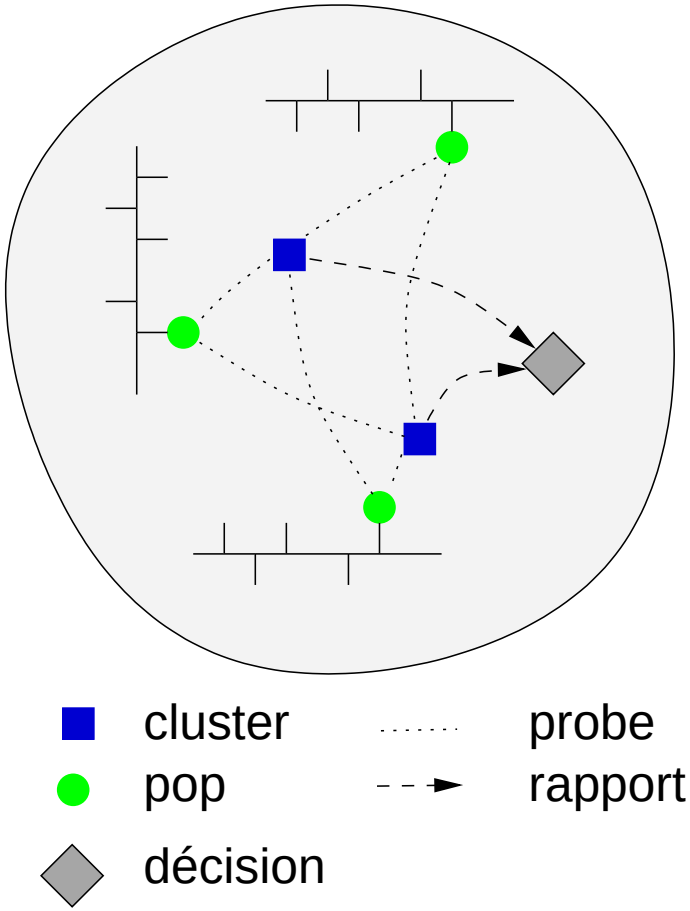
- tt sur le m̂ serveur
  - ✓ 2 ms ➠ 0 %
  - ✓ 200 ms ➠ 3 %
- 1 rés. DNS / obj
  - ✓ 200 ms ➠ 48 %

Redirection : routage

- requête *client* pour *document*
  - ✓ ensemble des Sgtes ayant le document
  - ✓ charge (CPU, nb. connexions...)
  - ✓ distance au client (hops, délai, BP ...)
- réponse rapide + infos à jour
  - ✓ méthode passive (IGP/BGP...)
  - ✓ méthode active *(network probing)*
    - ☞ à la demande
    - ☞ à l'avance
- Akamai ➠ *secret sauce*
- trouver un *surrogate* raisonnable

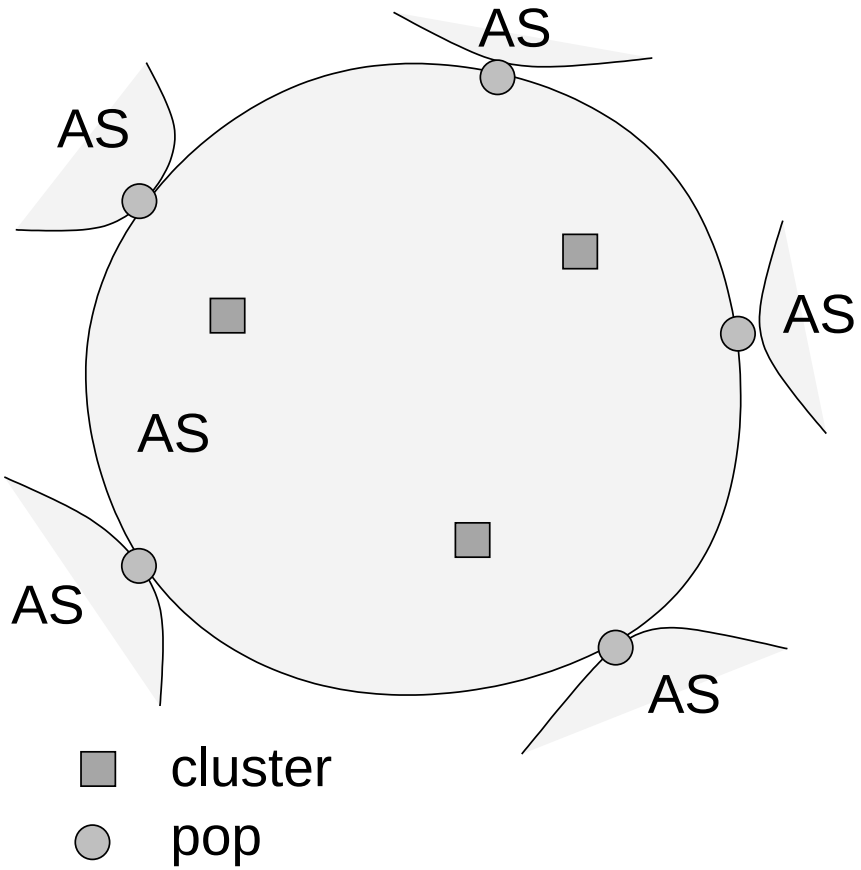
Redirection : routage intra AS

- approche active/à l'avance
- décision centralisée
- clusters de *surrogates*
- pop (agrégats de clients)
- charge des services dans chaque cluster
- chaque cluster probe tous les pops
- association (pop, service) ➠ cluster
- associations pop ➠ { cidr } (ex. IGP)



Redirection : routage extra AS

- point de *peering*
- = pop
- liste des CIDR fournie par BGP (route “descendante”)



Routage à la demande

- si topologie inconnue
- ou très gd nb de pops potentiels
- req. arrive
  - ✓ *probe* le client
  - ✓ maintient un cache de n clients
  - ✓ ➠ découverte topologie dynamique
- ex. créneaux horaires

Mesure des métriques

Passive

- infos des protocoles de routage
  - ✓ IGP *Interior Gateway Protocol*
    - ☞ OSPF ➠ topologie
    - ☞ RIP ➠ distances en nb de nœuds
  - ✓ EGP *Exterior Gateway Protocol* : BGP ➠ AS path length
- traces de trafic

Active

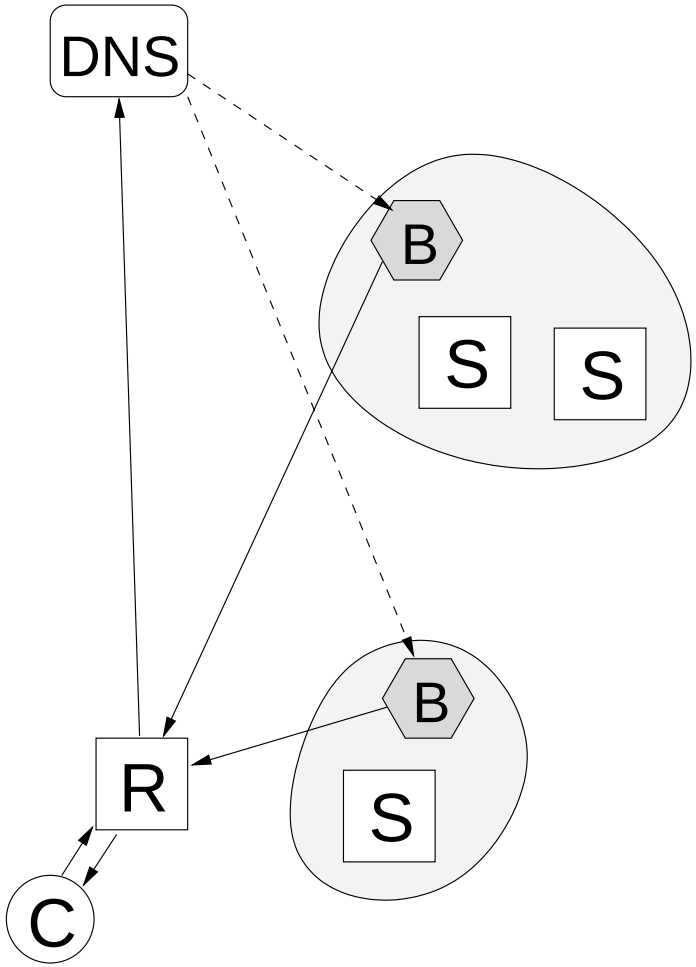
RTT : echo ICMP (ping)

pertes …

BP
- *packet pair* : BP lien goulot
- *packet train* : BP disponible

DNS boomerang

- Boomerang/Flash DNS/DNS flooding
- les clusters répondent
- proximité réseau
- si charge ➠ retarder la réponse
- routage très simple !



Distribution

Distribution : réplication

- Réplication
  - ✓ copie
    - ☞ *push*
    - ☞ *pull*
  - ✓ synchronisation (cohérence)

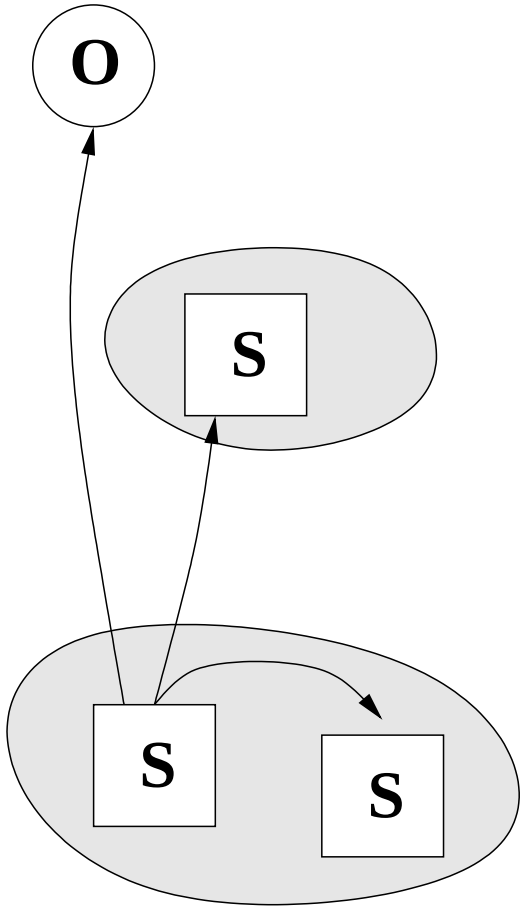
Push

- multicast
  - ✓ problème de la fiabilité
  - ✓ contrôle de flux (hétérogénéité)
  - ✓ interface des *surrogates*

Réplication : copie

Pull

- où chercher le contenu ?
  - ✓ origine
  - ✓ voisins
  - ✓ hiérarchie
  - ✓ ex. ICP
- forme de routage
  - ✓ déterminer les routes possibles
  - ✓ choisir la meilleure (distance réseau + charge serveur)



Réplication : synchronisation

maintenir la cohérence des *surrogates* avec l'origine

- *surrogate* = serveur authoritatif
  - ✓ invalidation de contenu
  - ✓ mises à jour
  - ✓ multicast
- *content signaling*
- tentatives actuelles (abandonnées ?)
  - ✓ WCIP *Web Cache Invalidation Protocol* (IETF/Cisco)
  - ✓ RUP *Resource Update Protocol* (IETF)

Web Content Distribution Protocol – WCDP

- content group (choix du créateur)
  - ✓ `register <cg>`
- object group
  - ✓ scalabilité, atomicité
- “pseudo-push”
  - ✓ invalidation avec rafraich. immédiat
  - ✓ ou rafraich retardé (charge)
- cohérence forte
  - ✓ envoi invalidation
  - ✓ attends les ACK
  - ✓ publie le nouvel objet

WCDP

- cohérence forte avec “push”
  - ✓ proxy “pulle” mais relaie les req.
  - ✓ qd ts les ACK reçus, origine publie et envoie COMMIT
  - ✓ proxy publie à réception du COMMIT
- Δ-cohérence
  - ✓ *heartbeat* et invalidation temporaire

– scalabilité ➠ pour miroirs

Web Cache Invalidation Protocol

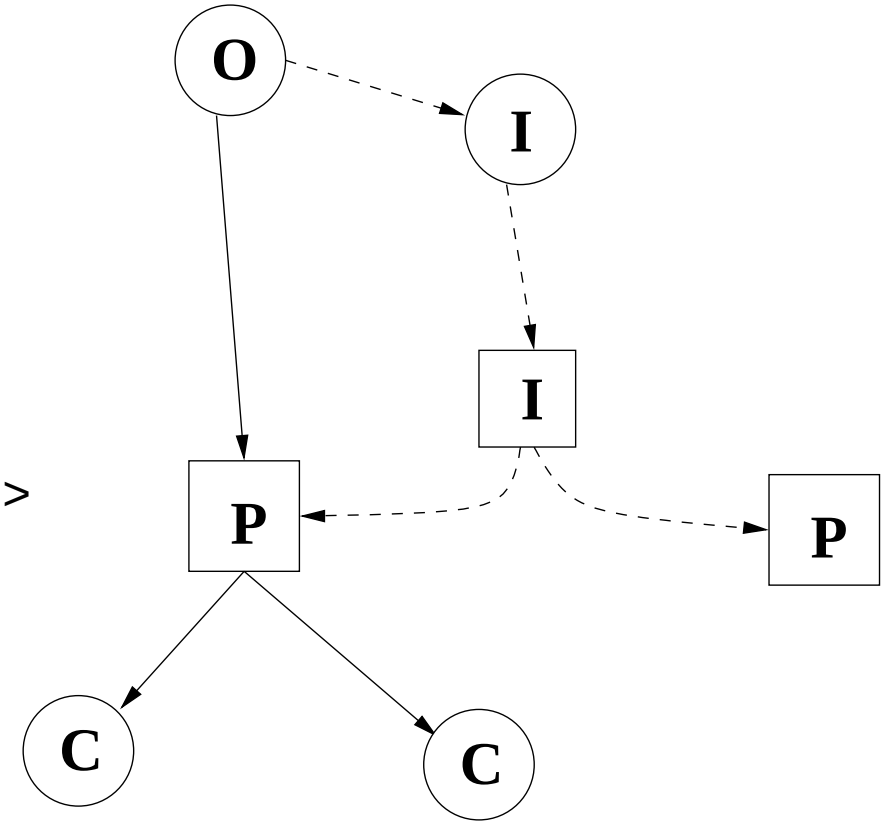
- assure la Δ-cohérence
- serveur d'invalidation ($\neq$ origine)
- canal d'invalidation
- client léger
- volume d'objets
  - ✓ unité de filtrage + unité de cohérence
- deux modes
  - ✓ mené par le proxy
    - ☞ se resynch. regulièrement
  - ✓ mené par le serveur (d'inv.)
    - ☞ invalidations/mises à jour + *heartbeat*

WCIP

- canal `wcip:...` (ex HTTP/XML)
- infos par :

⇐ `Invalidated-By: <wcip URI>`

- I peut être chez ISP !



Leases

- gestion classique
  - ✓ côté serveur (invalidation) ➠ états
  - ✓ côté client (revalidation syst.) ➠ msgs
- compromis : *lease* (bail) {O,P,d}
  - ✓ serv s'engage à notifier P des modifs de O pendant d

⇐ `Lease-Control: Grant-Lease | Renew-Lease ...`

- cohérence forte relachée

⇐ `Invalidate-Lease: ...`
⇒ `Invalidate-Ack: ...`

serv attend les ACK ou fin du bail pour modifier O

Cooperative leases

- *leases* pas scalable (état/cache et objet)
- coop lease {O, G, L, d, Δ }
  - ✓ G groupe, L leader, Δ cohérence
- groupe : $\searrow$ état S, $\searrow$ notifs S
- coopération pour cohérence $\perp$ coopération pour cache

Distribution de stream

- *stream* : non élastique
- on demand/live
- RTSP, RTP/RTCP
- gros fichiers
- longue occupation au serveur

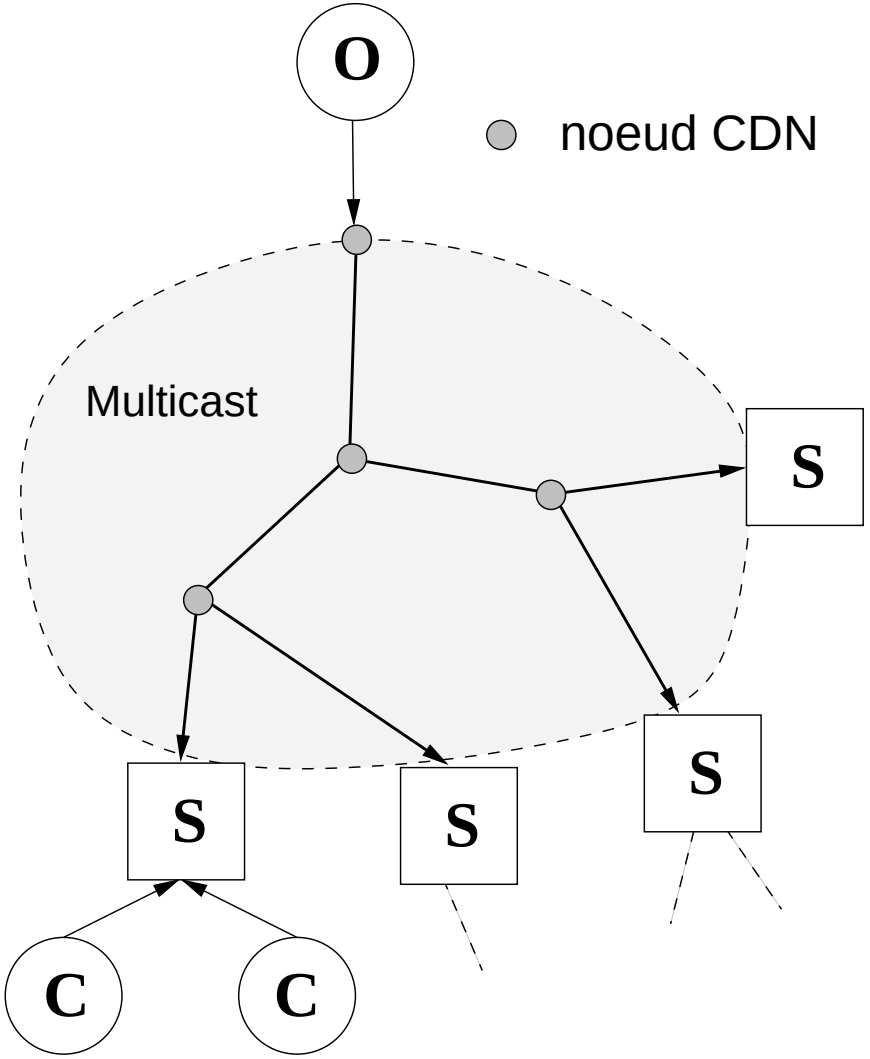
Stream à la demande

- qualité dépend du chemin
- CDN : streamer depuis le bord du réseau
- pb : taille des fichiers
  - ✓ *prefix caching* puis transfert élastique de la suite
  - ✓ segmentation des fichiers et répartition sur les réplicas
- qualité : *path diversity*

Stream live

CDN ➠ réseau *overlay*

- multicast
  - ✓ soulage l'origine
  - ✓ soulage le réseau
- routage ad-hoc
  - ✓ ex min bandwidth
  - ✓ ...
- protection des flots
  - ✓ FEC
  - ✓ flots redondants
- hétérogénéité



Autres

Delivery system

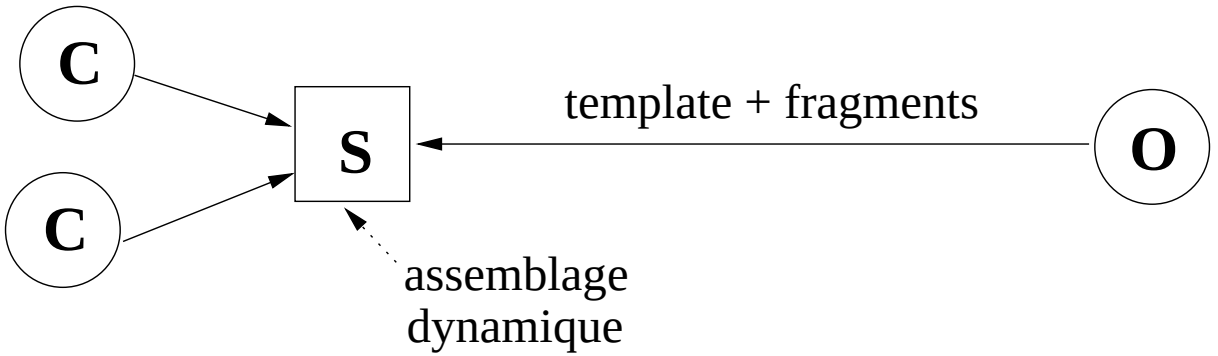
- proxy caches
  - ✓ web (statique, dynamique)
  - ✓ *stream*
    - ☞ real networks
    - ☞ windows media
- machines spécialisées (hardware, OS)
  - ✓ Network Appliances (NetApp), CacheFlow, Inktomi
  - ✓ Cisco, Intel
- besoins spécifiques des CDN ?

Comptabilité

- les logs sont répartis dans tous les *surrogates*
- besoin de générer une vision globale
  - ✓ pour le fournisseur de contenu
  - ✓ pour l'opérateur de CDN
- scalabilité ?
- structure hiérachique (multicast inverse)

Edge Side Includes

- Akamai/Oracle/... ’01
- assemblage dynamique de fragments en bordure
- protocole d’invalidation
- fragment
  - ✓ paramétres de cohérence
  - ✓ partagés par les utilisateurs
- *template*



ESI

- élts du template

```
<esi:include>
<esi:choose>
... style SSI/CGI
```

- Edge Side Includes for Java (JESI)
  - ✓ JSP ➠ ESI
  - ✓ JSP tag library
- protocole

⇒ `Surrogate-Capabilities: ...`
⇐ `Surrogate-Control: surr-id, action`

ESI

protocole d'invalidation

- req. inv. HTTP POST avec doc XML (port 4001)

⇐ `POST ...`
`msg-inv-XML`

⇒ `200 OK`
`msg-result-XML`

- invalidation par
  - ✓ URI
  - ✓ préfixe
  - ✓ regexp URI et en-tête
- ne propose pas de solution au pbs classiques
  - ✓ ➠ petit CDN

Placement des réplicas

théorie des graphes...

- facility location pb
  - ✓ i emplacements
  - ✓ construire à i coûte f_i
  - ✓ client j affecté à i ➠ coût $d_j \cdot c_{ij}$ (d_j demande de j)
  - ✓ solution ? (nb serv + empl. pour coût minimum)
  - ✓ NP-dur...
- Minimum K-median pb
  - ✓ n pts donnés
  - ✓ en choisir K, affecter clt j au plus proche ➠ $d_j \cdot c_{ij}$
  - ✓ NP-dur
- versions avec limite de capacité sur les serveurs

Algorithmes

algos approximation polynomiaux

Greedy choisir M réplicas sur N sites

- 1^{er} = coût min avec tous les clients sur lui
- ajouter le $2^{ième}$...
- itérer

Hot-Spot trier les N sites par trafic généré dans le voisinage

Random …

Enterprise CDN (ECDN)

Internet ➠ intranet
ICDN ➠ ECDN

- réseaux d’entreprise
- *e-learning*, *e-commerce*, *e-...*
- offre CISCO
  - ✓ *streaming*
  - ✓ réplication multicast

Interconnexion de CDNs

IETF CDI *Content Distribution Internetworking*

- qqes CDN mondiaux (Akamai...)
- intérêt des ISP
  - ✓ service ➠ revenu
  - ✓ infrastructure
- mais pb de couverture
- ➠ interconnecter les CDN
- projet abandonné ?

CDN / caches web

qu’est-ce qui les différencie ?

- redirection ? mais caches transparents
- push ?
- systèmes hybrides

Content Networks

Qu'est-ce qu'un CDN ?

Les CDN...

1. partagent dynamiquement l'infrastructure de réplication
2. fournissent un espace de noms adapté
3. utilisent les URL classiques

2 et 3 paraissent contradictoires

Un CDN est :

- une infrastructure de réplication
- un changement de sémantique de l'espace de noms URL