Web caching

23/05/2007

Federico Maggi
fmaggi@elet.polimi.it
http://www.elet.polimi.it/upload/fmaggi

Agenda

1. Problemi da risolvere: soluzioni
2. Web caching vs. Web replication
3. **Web caching**
   1. **Tipologie e (s)vantaggi**
   2. **Architetture**
   3. **Tipologie di deployment**
   4. **Dimensionamento: analisi costi vs. benefici**
   5. **Popolarità degli oggetti nel Web**
4. Web replication

Il problema della scalabilità

Incremento generalizzato del traffico di Internet:

- Diffusione delle tecnologie a banda larga
- Web browsing: aumentano le esigenze degli utenti
  - Maggior numero di oggetti scaricati
  - Disponibilità al download di oggetti di grandi dimensioni
  - Pagine composte da numerosi oggetti multimediali
- Applicazioni emergenti (VoIP, P2P, Web services, ...)
  - Crescente richiesta di banda per applicazioni diverse dalla semplice navigazione

Come affrontare l'aumento del carico?

Soluzioni empiriche

Approccio brute-force al problema della scalabilitá: si sfrutta il basso costo dell'hardware aggiungendo nuovi router, link, server e banda

Soluzione accettabile nel breve periodo, ma contribuisce a fare crescere i costi e la difficoltà di manutenzione dell'impianto.

Il problema fondamentale è che non si riesce a contrastare l'aumento del traffico circolante nel sistema: probabile che a breve il sistema sia nuovamente congestionato!

Approcci sistematici – Web caching

Web caching: si sfrutta la ripetitività nel comportamento degli utenti per velocizzare il download degli oggetti piú popolari. Questi sono salvati su una macchina locale (solitamente il proxy).

Esempio di Web caching

Un team di sviluppo accede frequentemente su Internet alla documentazione tecnica di alcune librerie Java.

Scaricando e salvando su una macchina condivisa la documentazione si conseguono due vantaggi:

1. maggiore velocitá di accesso ai documenti
2. minor traffico in uscita verso Internet

Approcci sistematici – Web replication

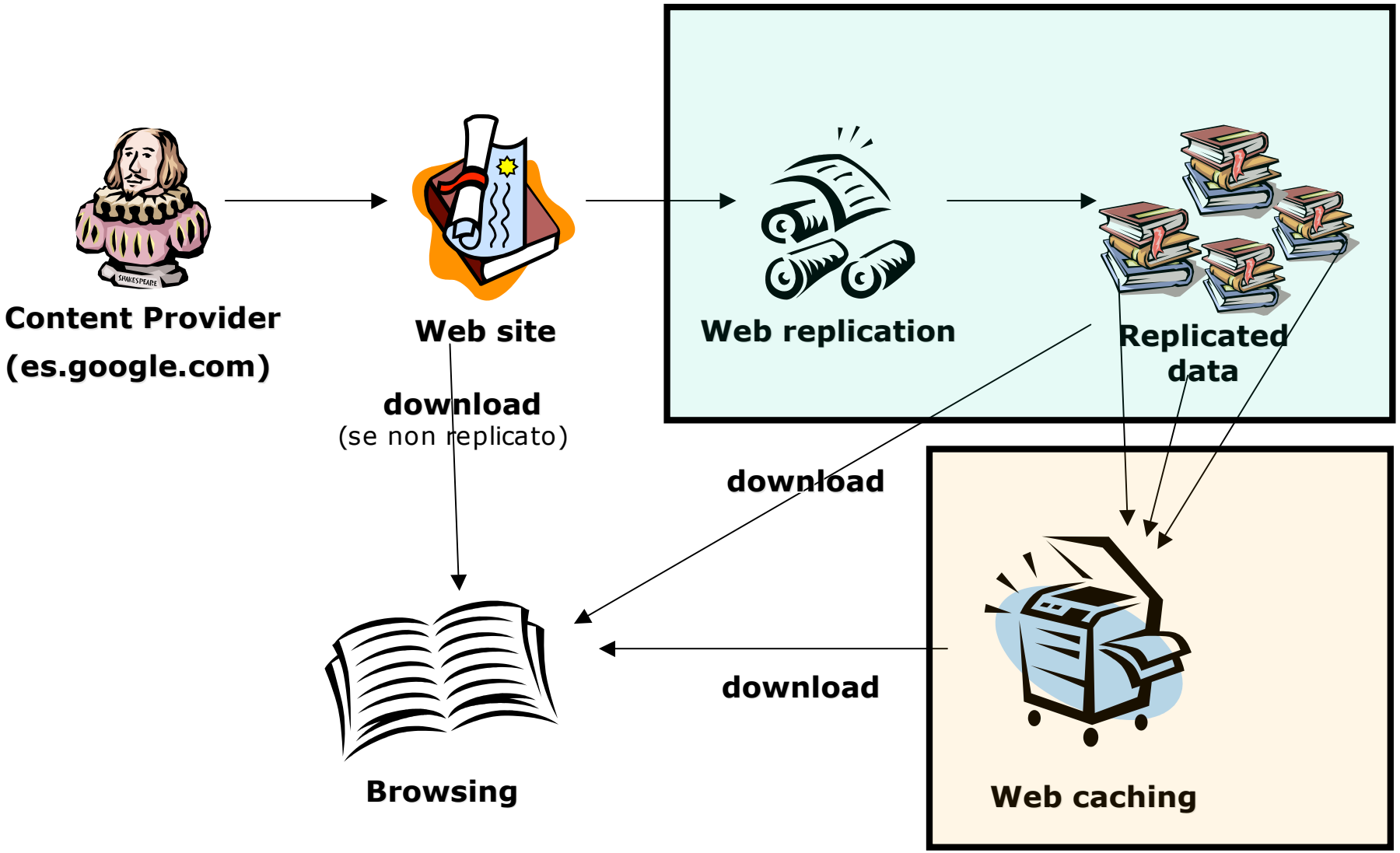
Web replication: esistono diversi server replicati da cui è possibile scaricare uno stesso oggetto Web.

Esempio di Web replication

Il sito di e-commerce di una multinazionale processa ordini provenienti da diversi continenti. Acquisendo server in aree geografiche strategiche e replicando il sito Web su tali server è possibile:

1. bilanciare il carico tra i diversi server
2. evitare che i picchi di richieste del periodo (es., natalizio) congestionino il sistema
3. migliorare i tempi di risposta **percepiti** dai clienti

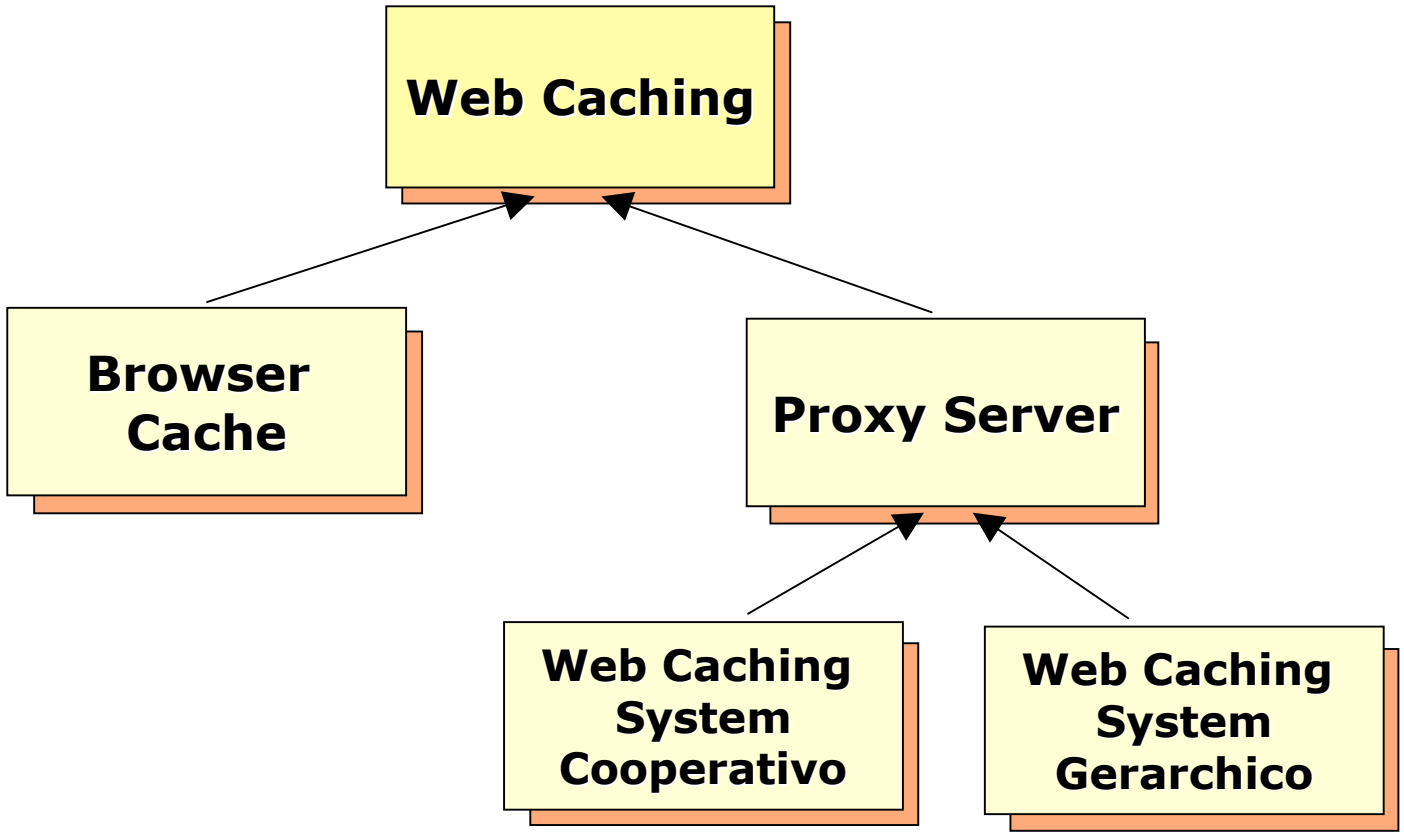
The big picture



Web caching

Tipologie dei sistemi di Web caching

La classificazione distingue posizionamento (lato client/lato ISP) e architettura del sistema di Web caching.



Browser cache: web-caching "locale"

Quando l'utente richiede il download di un oggetto Web il browser compie una serie di operazioni:

- Il **browser** controlla se l'oggetto è stato scaricato recentemente
- In caso affermativo l'oggetto é salvato temporaneamente su disco (o in RAM) nella browser cache:
  - se si rileva che la copia in cache dell'oggetto non é aggiornata lo si scarica nuovamente
  - se la copia in cache é aggiornata, viene ricaricata dal browser (cache hit)
- Nel caso l'oggetto non sia invece disponibile (cache miss), esso viene scaricato normalmente e una sua copia é salvata nella browser cache

Proxy server: web-caching “condiviso”

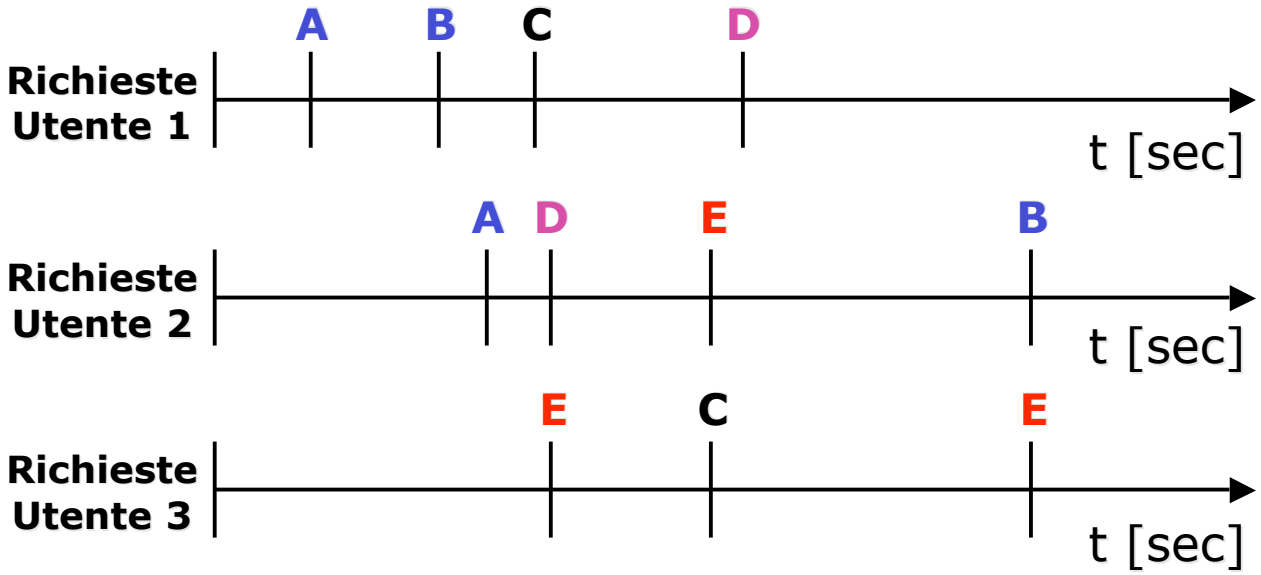
I proxy server sono server di rete che (svolgono funzioni di controllo di accesso e) implementano meccanismi di Web caching.

Nel ambito del Web caching trattiamo i proxy esclusivamente come Web caching proxy, ovvero come erogatori di un servizio di Web caching.

Rispetto alla browser cache, il proxy ha vantaggio di essere un sistema di caching condiviso, ovvero di poter sfruttare la ripetitivitá dei download di una comunitá di utenti.

Vantaggi di una Web cache **condivisa**

Proxy condiviso da tre utenti:



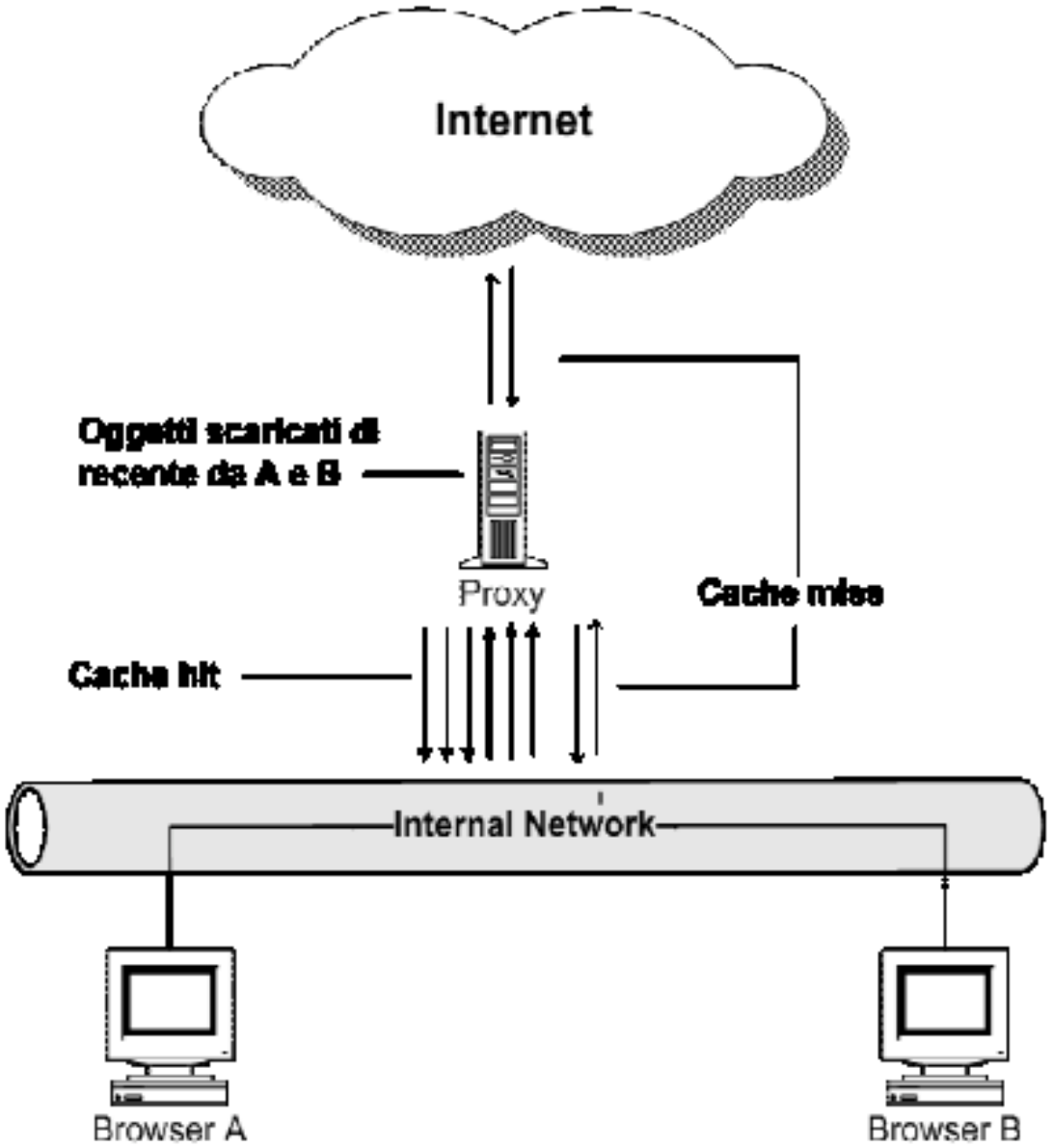
Grazie al Web caching si **risparmiano** 6 download:

L'utente 1 scarica dal proxy la pagina D.

L'utente 2 scarica dal proxy le pagine A,E,B.

L'utente 3 scarica dal proxy la pagina C e la pagina E (solo alla seconda richiesta).

Rete aziendale con **singolo** proxy



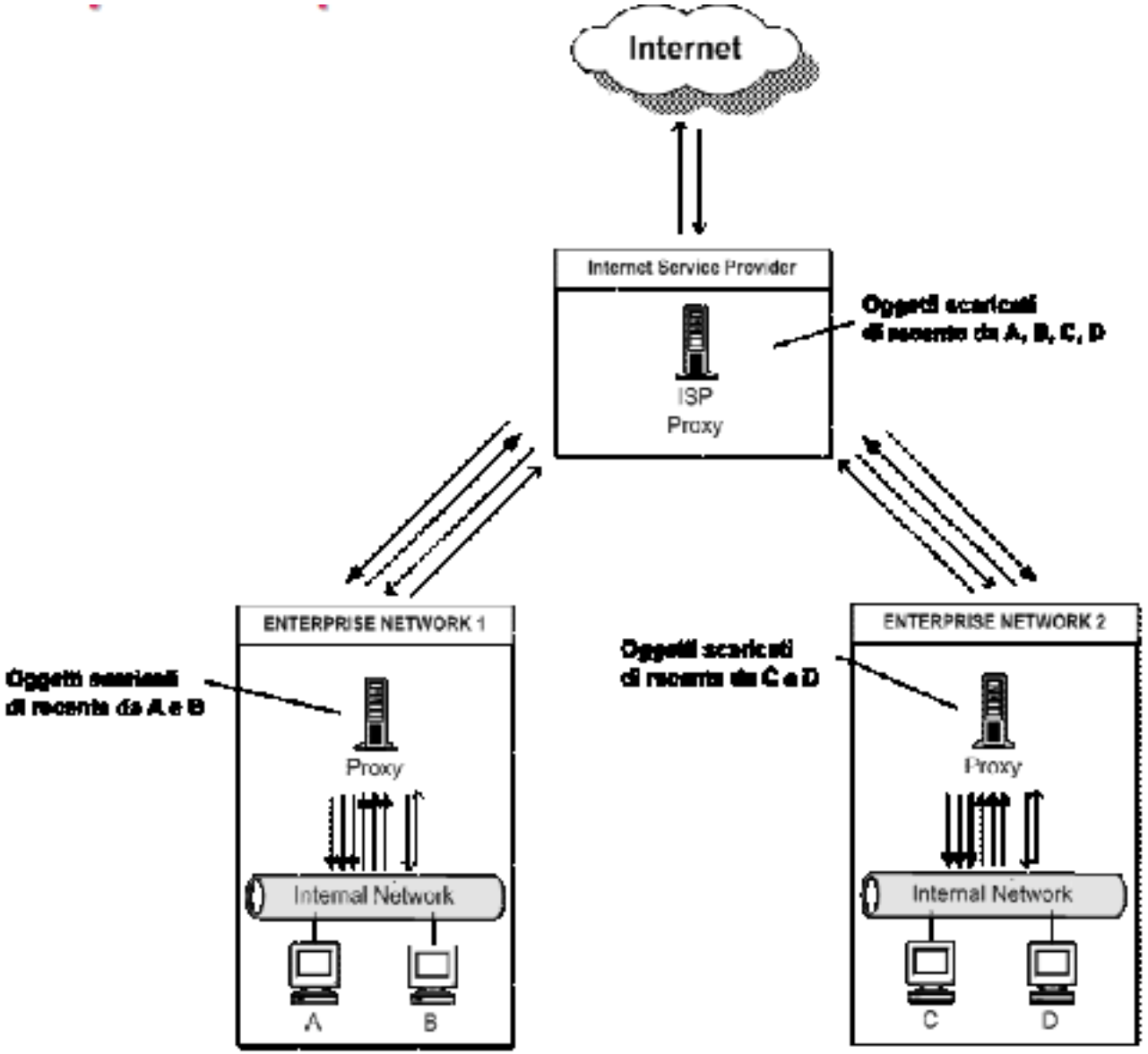
Rete aziendale con **singolo** proxy

Nella configurazione più semplice, il proxy intercetta tutte le richieste uscenti dalla rete. In generale é collocato nella rete interna:

- Soluzione poco realistica per reti geografiche di grandi dimensioni e con molti link verso l'esterno
- Soluzione particolarmente inefficiente in reti con elevato traffico

Il principale vantaggio di questa configurazione é la semplicitá di installazione, configurazione e manutenzione.

Proxy **multipli**



Organizzazione a **livelli**

Nei sistemi di Web caching piú complessi esiste una gerarchia di proxy. Le richieste risalgono la gerarchia fino ad incontrare un proxy che le possa servire o uscire verso Internet.

Ad esempio, la rete di un'universitá potrebbe contenere proxy all'interno dei laboratori, proxy dipartimentali, proxy a livello campus e proxy di ateneo.

L'organizzazione a livelli non coinvolge solamente i proxy di una stessa azienda, ma anche la catena di proxy "esterni" attraversati dalla richiesta nel suo cammino end-to-end.

Architetture, dimensionamento e configurazione

Proxy: dimensionamento e configurazione

Per evitare di creare colli di bottiglia prestazionali, il proxy server é spesso una macchina **dedicata**.

Generalmente le richieste hardware sono modeste, almeno per reti di medie dimensioni: Squid, uno dei software di Web caching piú diffusi, raccomanda 128MB di RAM e una quota disco tra 512MB (rete con pochi utenti) e 24GB (reti su larga scala).

Diverse societá vendono **proxy preconfigurati** per reti con un numero di utenti variabile tra 10 e 250.000. I prezzi attuali sono compresi tra poche centinaia a decine di migliaia di dollari.

Complessitá architetturale

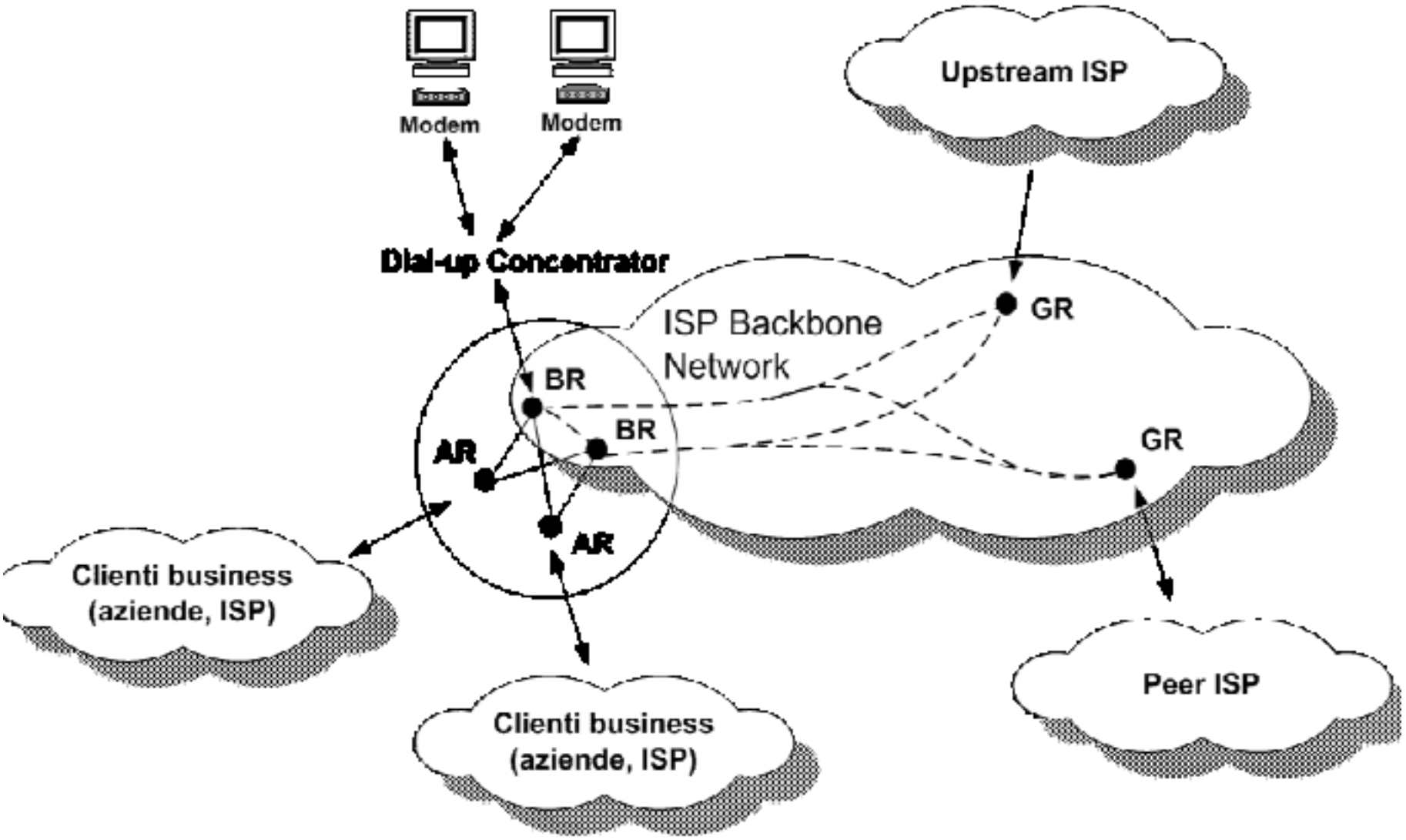
Nonostante la semplicitá del Web caching, la sua implementazione é resa difficile dalla complessitá delle architetture in cui esso si inserisce.

In particolare, le maggiori difficoltá si presentano in sistemi distribuiti su scala geografica.

Come caso di studio, consideriamo un Internet Service Provider (ISP), ovvero una societá che fornisce su scala **geografica** servizi di connettivitá a utenti privati e aziende.

- Come collocare i proxy nella **rete dell'ISP**?
- Come collocare i proxy nelle **aziende**?

Architettura di un ISP



Elementi della rete di un ISP

ISP backbone network: dorsale di rete che permette di smistare il traffico verso Internet

Access router (AR): router che consente alle aziende di instradare il proprio traffico verso la ISP backbone

Dial-up concentrator: sistema per spartire la banda di accesso alla ISP backbone tra utenti privati che dispongono di connessioni lente

Backbone router (BR): router di servizio per l'instradamento all'interno della ISP backbone

Elementi della rete di un ISP (2)

Gateway router: router per instradamento dalla backbone verso reti gestite da altre societá

L'ISP tipicamente stipula contratti di peering che consentono il mutuo instradamento di traffico con altri reti (peer ISP).

Il contratto puó anche prevedere flussi di traffico unidirezionali (downstream/upstream ISP).

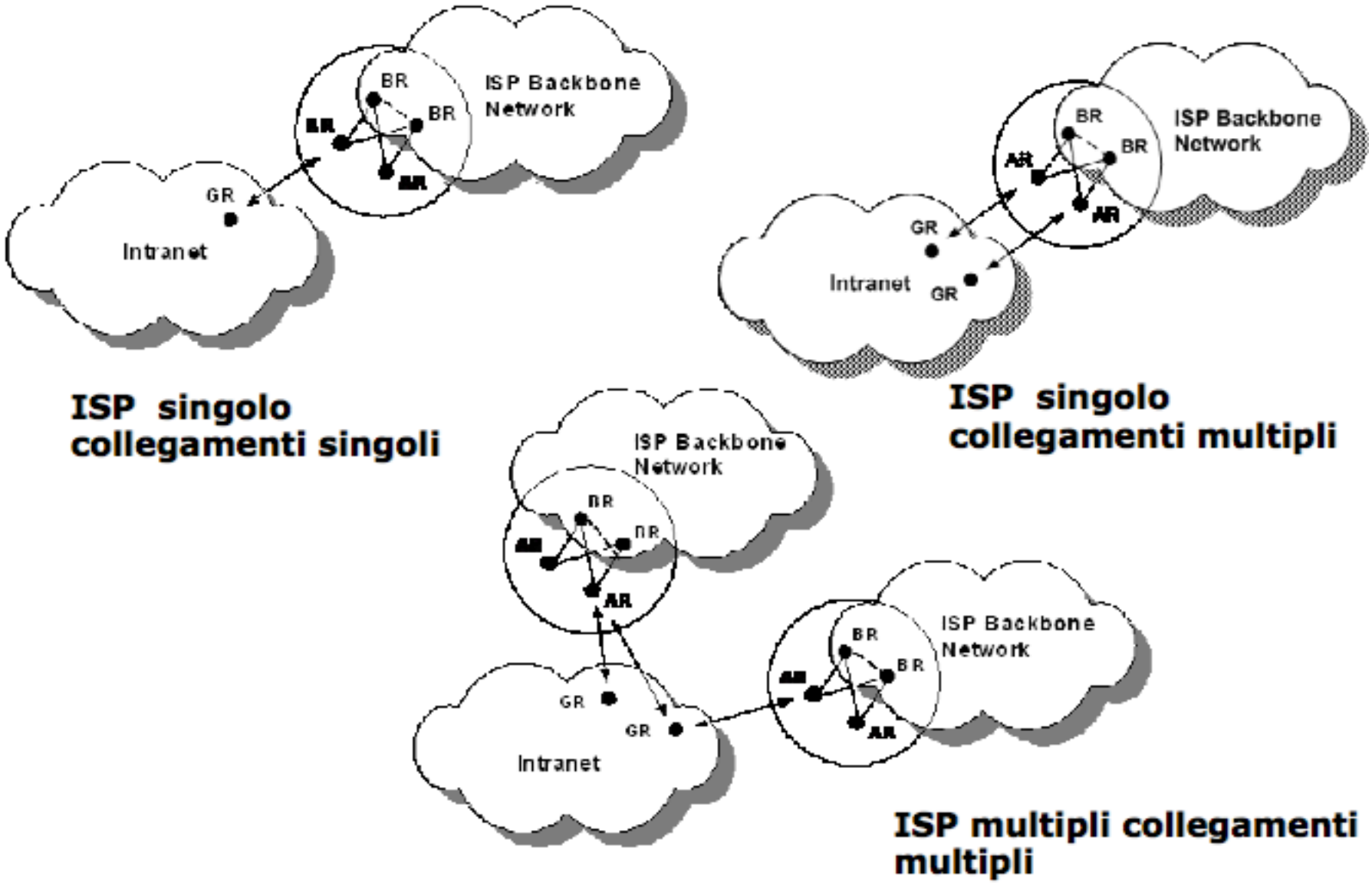
Multihoming

Alcune delle difficoltá nella collocazione del proxy nella backbone sono legate alla molteplicitá dei link che collegano azienda e ISP (multihoming).

Collegamenti multipli permettono di mantenere la connettività anche in presenza di guasti e possono bilanciare il traffico uscente su AR diversi o su ISP diversi.

Dal punto di vista dell'azienda, collegamenti multipli rendono difficile la collocazione del proxy aziendale (non esiste **un** punto dove transitano **tutte** le richieste in uscita). Anche per l'ISP non é semplice decidere dove collocare i propri proxy.

Tipologie di connessione azienda-ISP



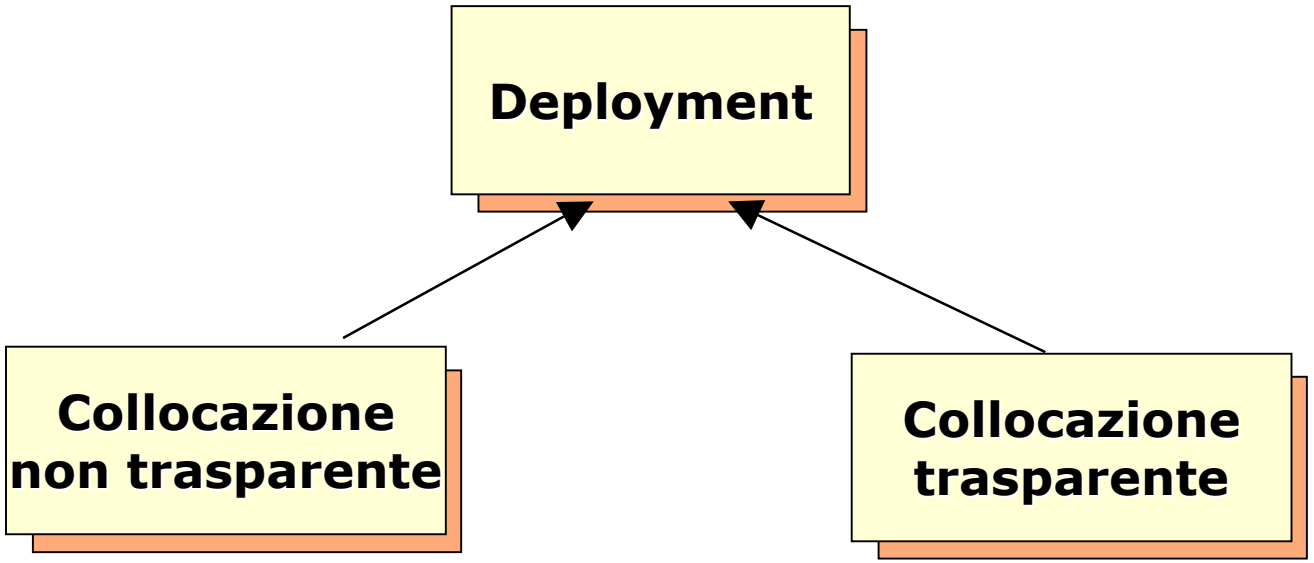
Politiche di collocazione dei proxy

Principalmente si distinguono due famiglie di politiche di collocazione: collocazione trasparente e collocazione non trasparente.

La collocazione non trasparente richiede che i browser degli utenti vengano configurati esplicitamente per sfruttare il proxy. E' una soluzione semplice, ma richiede la collaborazione degli utenti. Inoltre é complessa da realizzare per un ISP.

Una collocazione trasparente permette invece di mascherare l'esistenza dei proxy, ma puó richiedere hardware dedicato o una configurazione decisamente complessa.

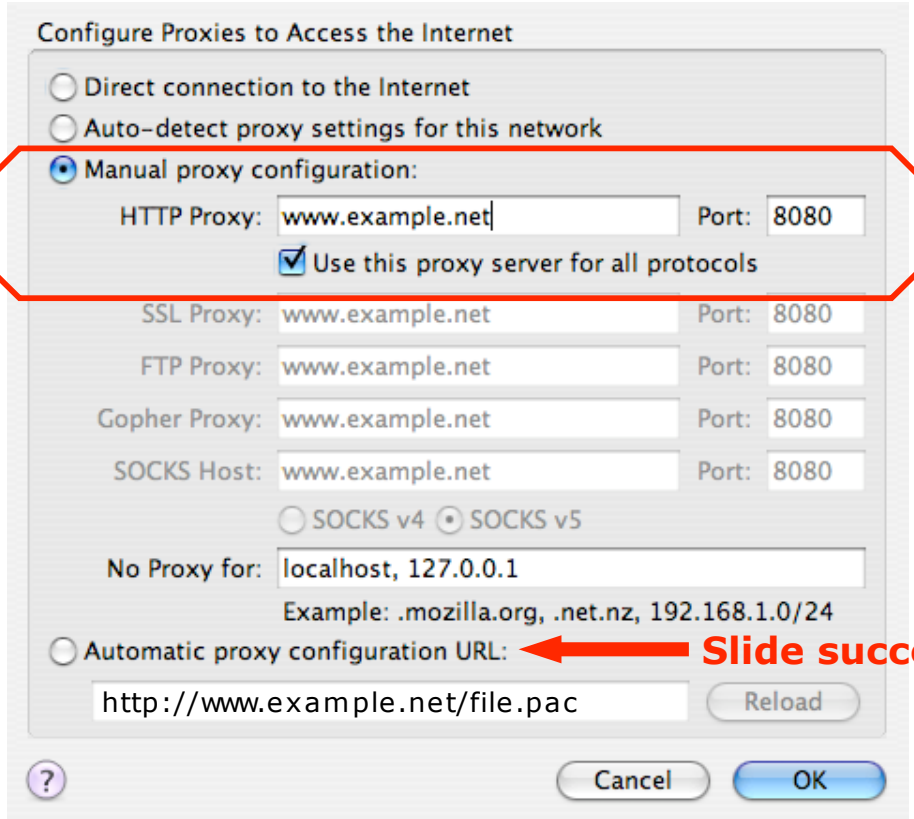
Tipologie di deployment



- **Configurazione esplicita**
- **Browser auconfiguration**
- **Proxy autodiscovery**

- **Interception proxy**
- **Switch L4**
- **Switch L7**
- **Intercepting link**

Configurazione esplicita del browser



Vantaggi: flessibilità nella collocazione del proxy

Svantaggi:

- E' possibile che l'utente possa evitare il proxy andando a riconfigurare il proprio browser
- Scarsa tolleranza ai guasti

Browser autoconfiguration

Periodicamente il browser scarica un file .pac (Proxy Auto Configuration) che descrive le politiche di instradamento delle richieste.

- In caso di caduta di un proxy la rete è bloccata solo fino a quando il file .PAC non è modificato con informazioni che permettano di aggirare il proxy caduto.

Vantaggi: flessibilità nelle politiche di gestione dei proxy e buona tolleranza ai guasti
Svantaggi: il server da cui é scaricato il .PAC diventa il nuovo single point of failure della rete

Esempio di file .PAC

Un file .PAC é un file Javascript che implementa le funzioni richieste dal browser per determinare l'URL a cui inviare le richieste.

```
function FindProxyForURL(url, host)
  {
     if (shExpMatch(url, "http://www.polimi.it*"))
     {
           return "DIRECT";
     }
     if (isInNet(myIpAddress(), "192.168.1.0",
                 "255.255.255.0"))
           return "PROXY 192.168.1.1:8080";
     else
           return "DIRECT";
  }
```

Proxy Autodiscovery

Permette la convergenza di collocazione trasparente e non trasparente.

La posizione del file .PAC nella rete è indicata direttamente o dal server DHCP o dal server DNS.

E' generalmente usato in reti aziendali di grandi dimensioni.

Vantaggi: trasparente all'utente finale e piú tollerante ai guasti della browser autconfiguration

Svantaggi: richiede l'adozione di un protocollo specifico (WPAD – Web Proxy Auto-Discovery) non ancora standardizzato

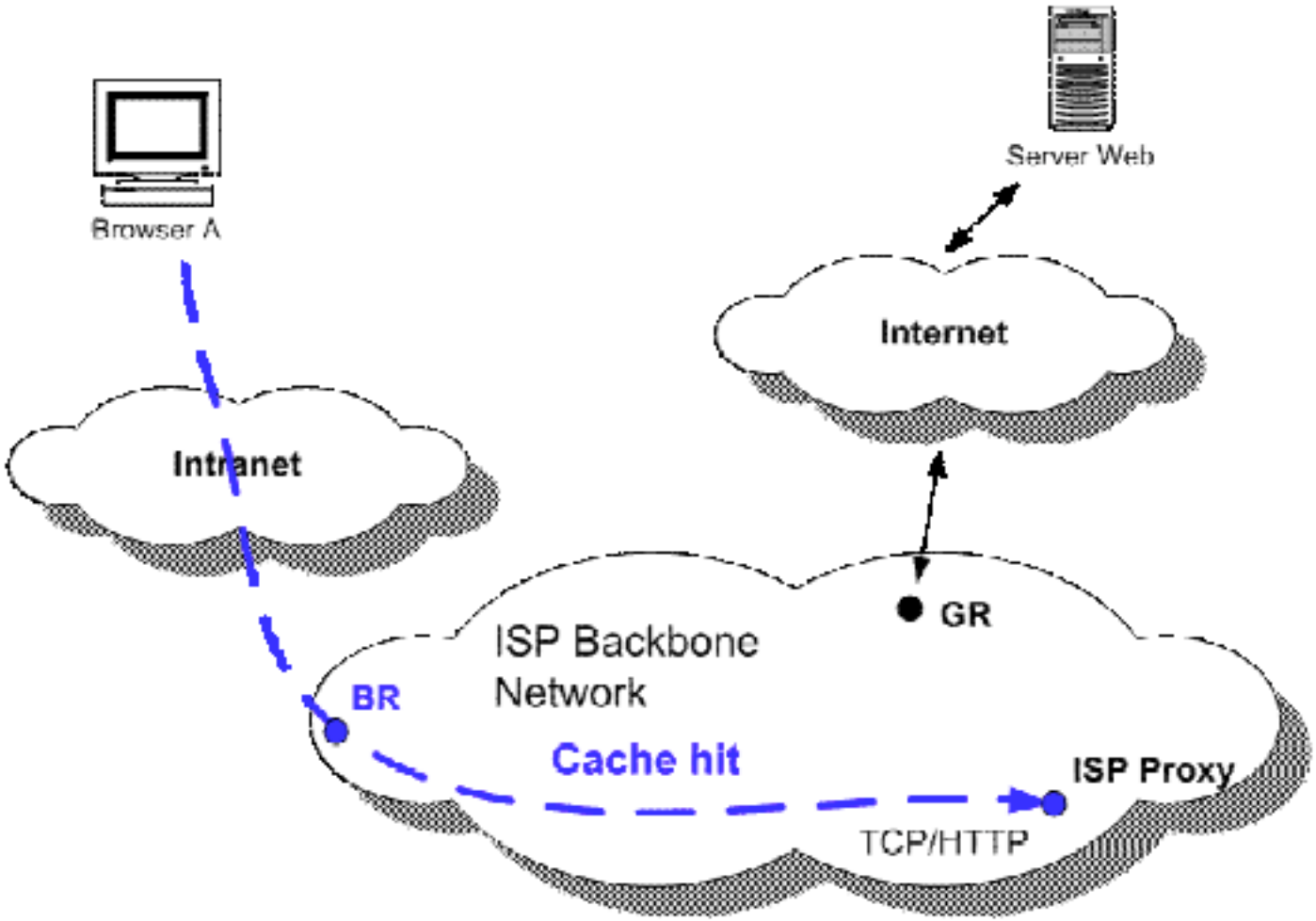
Interception proxy

I router (BR) redirezionano il traffico uscente verso il proxy (interception proxy).

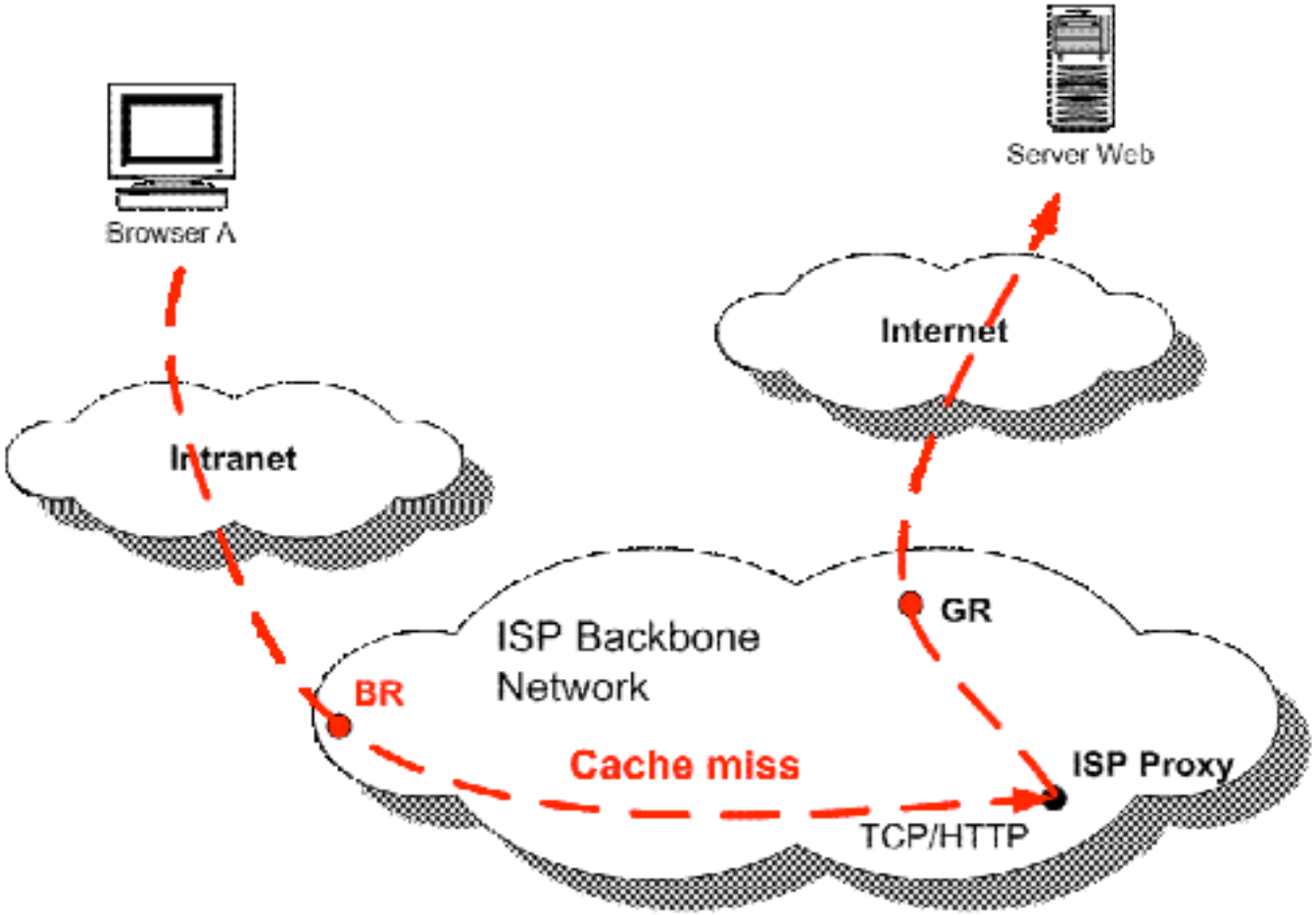
Il pacchetto originale é inserito nel pacchetto destinato al proxy (IP-in-IP Encapsulation).

Alla ricezione il proxy estrae il pacchetto interno e, conoscendo l'IP di destinazione originale, si finge il server di destinazione della richiesta (Connection Hijacking).

Interception proxy (cache hit)



Interception proxy (cache miss)



Vantaggi e svantaggi dell'Hijacking

Vantaggi: in caso di guasto del proxy, é sufficiente che i router rilevino la caduta della macchina per ripristinare correttamente il servizio.

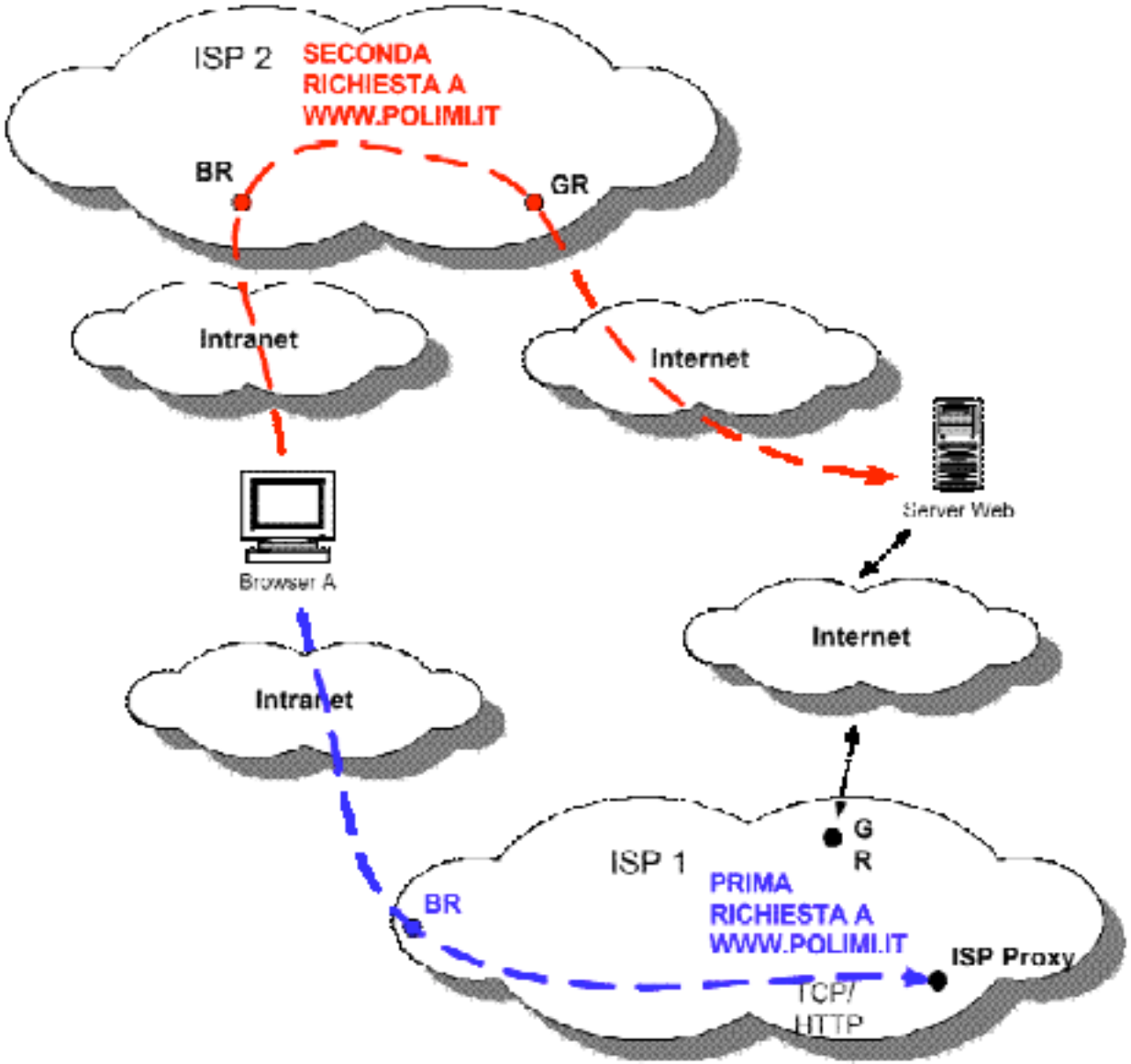
Svantaggi:

- il connection hijacking viola il principio end-to-end alla base di Internet (problema minore):

"*An application can completely and correctly be implemented only with the knowledge and help of the application standing at the points of the communication system*" (Saltzer, 1984)

- possono sorgere problemi connessi al routing delle richieste da reti che sfruttino connessioni multiple (multipath problem)

Multipath!


ISP 2
SECONDA
RICHIESTA A
WWW.POLIMI.IT
BR
GR
Intranet
Internet
Server Web
Browser A
Internet
Intranet
G
R
ISP 1
PRIMA
RICHIESTA A
WWW.POLIMI.IT
BR
ISP Proxy
TCP/
HTTP

Soluzioni al multipath problem: Switch L4

Dispositivo di rete che smista i pacchetti in transito **anche** sulla base del contenuto dell'header **TCP** (non solo sull'IP).

Grazie agli switch L4 si può imporre che i pacchetti di una **stessa connessione** siano **sempre** instradati verso lo stesso ISP.

Se connesso ad una batteria di proxy lo switch L4 può svolgere anche funzioni di load balancer.

Consente di **evitare** l'IP-in-IP **encapsulation**.

Soluzioni al multipath problem: Switch L7

Estensione dello switch L4. Può interpretare le richieste sia a livello TCP che applicativo (es., HTTP).

Rispetto allo switch L4, lo switch L7 permette di **non redirezionare** le richieste a oggetti non cacheabili (es. oggetti serviti in streaming). Perché?

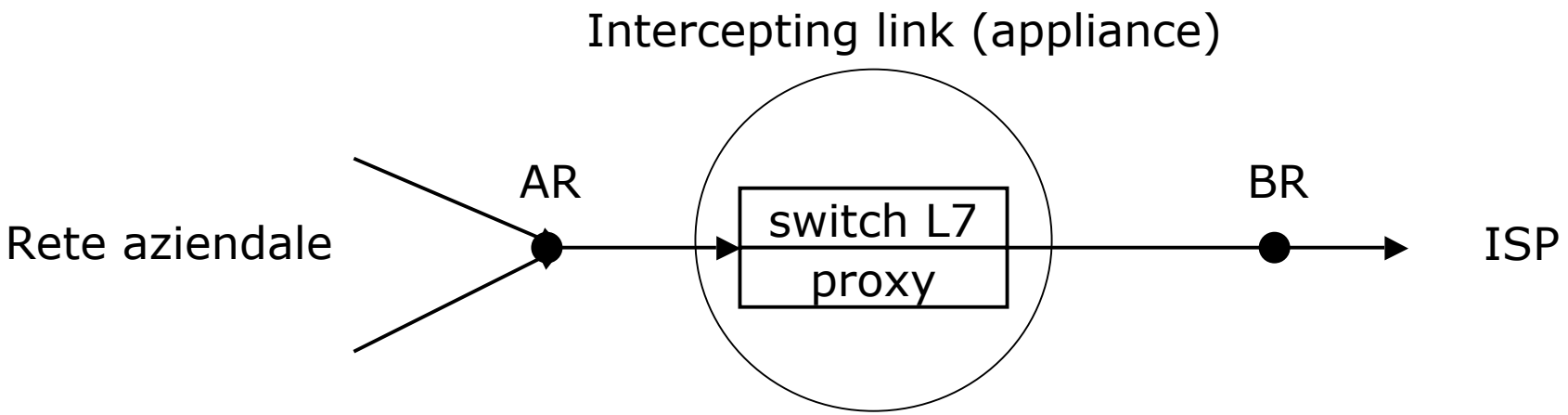
Inoltre consente di separare i proxy in base al contenuto cacheato (HTML, video, immagini).

Se usato senza criterio può ridurre sensibilmente il throughput della rete.

Intercepting link

Estensione dello switch L7. La stessa macchina è al contempo uno switch L7 e un proxy.
La caratteristica distintiva è di essere un dispositivo integrato e autoconfigurante.

- approccio plug'n'play al Web caching



Prestazioni

Web Data caching

Prima di ricorrere ad un meccanismo di Web caching, è necessario effettuare un'analisi costi-benefici al fine di prevedere **l'effettivo** incremento delle prestazioni.

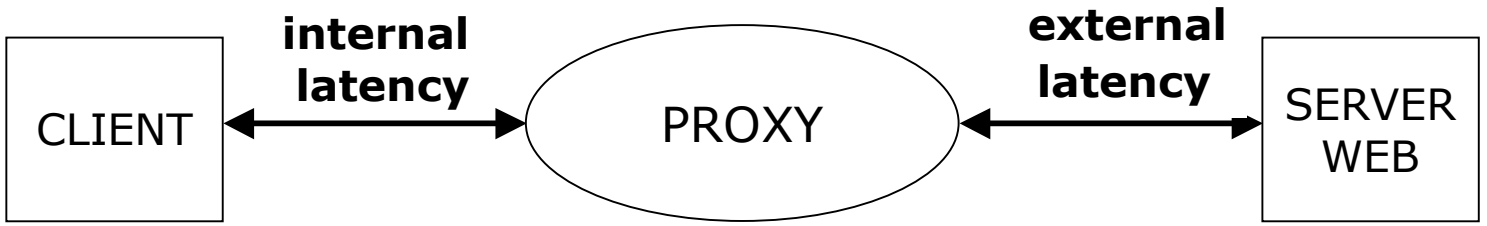
I cache hit dovuti all'attivitá degli **altri** utenti (shared hit) sono considerati la maggiore **motivazione** per l'uso del Web caching:

- Shared hit spesso compresi tra il 45% e il 50%
- Alcuni studi riportano valori prossimi all'85%
- Il byte hit rate è attorno al 50%, ovvero il 50% dei (K)B scaricati provengono **dal proxy**

Nel seguito **assumeremo** sempre per prudenza un guadagno da shared hit del 45%.

Riduzione effettiva della **latenza**

Trascurando l'overhead del proxy, il tempo di download può essere scomposto in due componenti:



- L'external latency impatta per il 77-88% del totale
- Assumendo uno shared hit rate del 45%, il guadagno di banda massimo è del

  45 % x 88% = 40%
- Purtroppo un'analisi di questo tipo è scorretta perchè non tiene conto delle dimensioni dei file
  - il guadagno nel caso ottimo è attorno al 22-26%
  - **i proxy riducono i trasferimenti (già brevi) di file piccoli**

Guadagno nel caso pessimo

Andiamo a rimuovere ulteriori ipotesi semplificative.

Teniamo conto di ulteriori aspetti che non abbiamo ancora considerato (overhead del proxy):

- –E' necessaria una fase di connessione tra proxy e server Web in caso di cache miss
- –Il proxy introduce un suo overhead di gestione del TCP, ovvero introduce un ritardo dovuto alla apertura e analisi dei pacchetti in transito

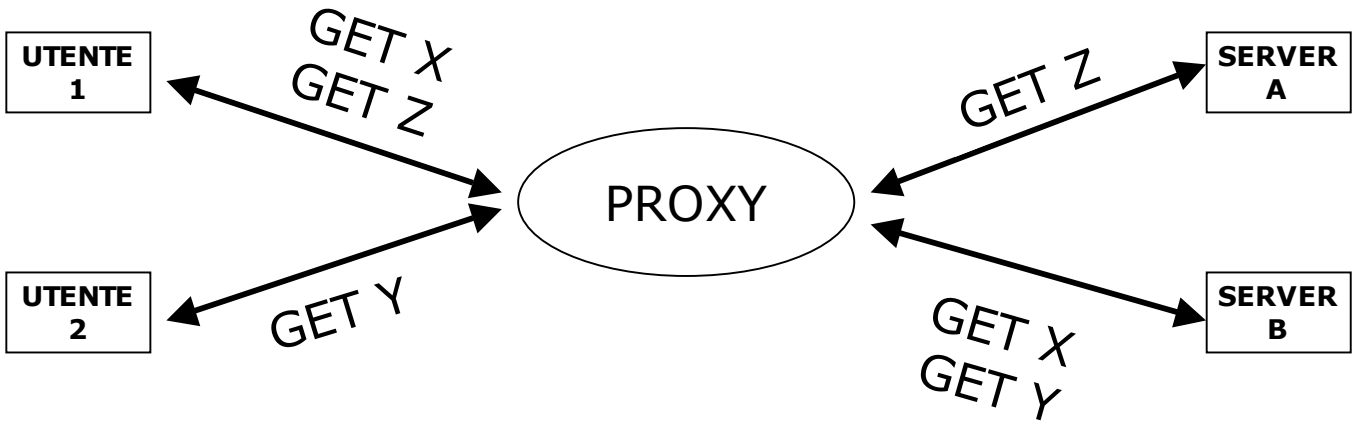
•Studi hanno dimostrato che tenendo conto anche di questi ritardi il guadagno nel caso pessimo è soltanto del 3%!

MA ALLORA IL CACHING CONVIENE DAVVERO?

➢In realtà questa stima è severa perchè non tiene conto della congestioni di rete che rallentano i download che potrebbero essere serviti da un proxy locale. Inoltre…

TCP Connection Caching

Il proxy permette anche il connection caching:



Vengono mantenute attive le **connessioni** giá aperte per i cache miss

Questo permette di risparmiare agli utenti i tempi di connessione. Tuttavia lo stesso canale non può però essere usato simultaneamente da più client (no HTTP pipelining)

- La pipeline viene servita in ordine FIFO e gli utenti dovrebbero aspettarsi fra loro

Ethernet caching vs Web caching

Riduzione della latenza di download	Media	Mediana
Web Connection caching	21%	40%
Web Connection caching + Web Data caching	**24%**	**48%**
Ethernet Connection caching	2%	20%
Ethernet Data caching	**47%**	40%

TCP Connection Splitting

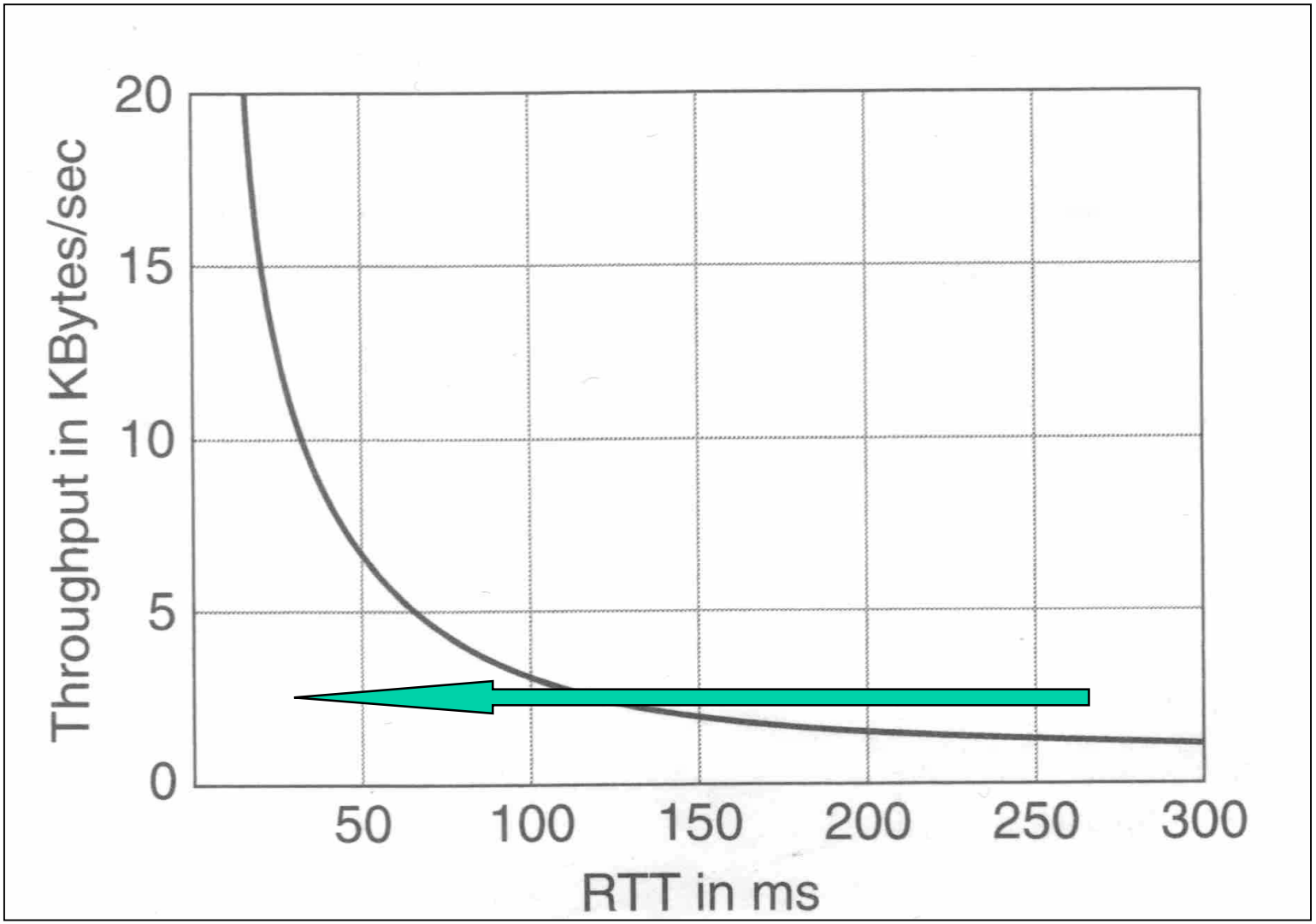
Round Trip Time (RTT): tempo richiesto da un pacchetto per viaggiare da una certa sorgente ad una certa destinazione e fare ritorno

Un RTT troppo alto può limitare notevolmente il throughput del TCP. Analiticamente si vede che questo dipende anche dalle dimensioni della finestra del TCP (slide successiva).

La presenza di un proxy spezza il viaggio dei pacchetti su due RTT più **piccoli**:

- L'incremento massimo **teorico** del throughput è di tre volte, quello **reale** è circa del doppio
- Il vantaggi maggiori si hanno per oggetti di **grandi** dimensioni

RTT vs Throughput massimo teorico



Riduzione del traffico di rete

La riduzione di **utilizzo** della rete consentita dai proxy è stimata al 25%.

Tuttavia senza alcune cautele il traffico può aumentare: ad esempio, in presenza di **abort** del client dopo un cache miss i proxy devono **cessare** immediatamente il **download** dell'oggetto

- Studi in letteratura mostrano che **continuare** a scaricare l'oggetto per salvarlo in cache è una politica **sbagliata**
- Si può mantenere sul proxy la porzione di oggetto scaricato usando all'occorrenza una HTTP partial GET per concludere il download

Caratterizzazione degli oggetti Web

Dimensione dei file

	Aggregato	HTML	Immagini	ZIP/RAR/...
Numero oggetti	**20728**	**1141**	**7025**	**67**
Media (kB)	**15.1**	**7.1**	**8.7**	**1501.8**
Mediana (kB)	**4.6**	**4.5**	**4.4**	**38.1**

Distribuzione fortemente asimmetrica: generalmente oltre il 75% degli oggetti ha piccole dimensioni rispetto alla media.

Tra i rimanenti, una percentuale consistente è di oggetti di grandi dimensioni.

Impatto sul Web caching

L'asimmetria nella distribuzione degli oggetti Web implica requisiti contrastanti per il Web caching:

- Cacheare oggetti piccoli migliora l'hit rate
- Cacheare oggetti grandi riduce i tempi di download piú lunghi

Quali tipi di oggetti Web mantenere in **cache**?

Il mix ottimo é quello che consente di minimizzare il tempo medio di download percepito dagli utenti.

Questo dipende dalle caratteristiche del traffico in esame e, in particolare, dalla **popolaritá** dei diversi tipi di oggetti Web.

Popolaritá degli oggetti Web

Alcune statistiche comuni:

- 90% dei download diretto a file HTML/immagini
- Gli oggetti più popolari di un sito sono immagini
- Ogni pagina HTML contiene in media 10 immagini

Poiché il download di una pagina HTML implica il download di un certo numero di immagini, possiamo assumere che gli accessi ai singoli oggetti abbiano un certo grado di dipendenza.

Quale legge esprime la **dipendenza** tra i diversi accessi?

Legge di Zipf

E’ una legge empirica, osservata per la prima volta dal prof. G.K. Zipf di Harvard nel 1949.

Applicata per la prima volta in linguistica allo studio della frequenza d’impiego delle parole negli articoli di giornale.

Legge di Zipf:

> “la frequenza d’uso della k-esima parola piú popolare è inversamente proporzionale a k”

Applicata all’ambito Web, se A e B sono il primo e il secondo oggetto per popolaritá, allora la frequenza di accesso a B è metá di quella ad A.

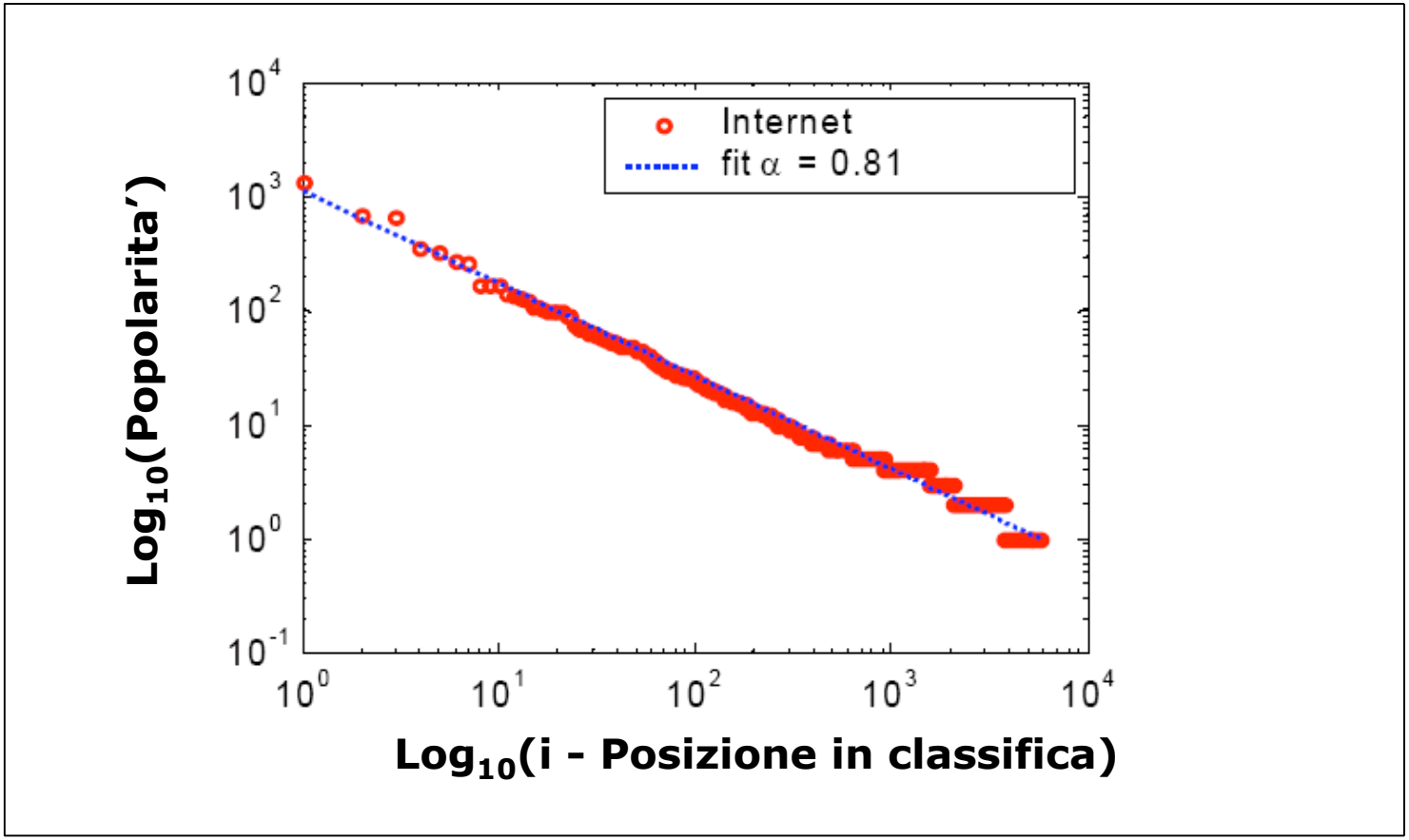
Legge di Zipf (definizione)

Nella sua veste piú generale, la Legge di Zipf afferma che l'i-esimo oggetto più popolare è proporzionale a $1/i^{\alpha}$, con $1 \geq \alpha > 0$.

Se $\alpha=1$ (parametro di scala) si ha effettivamente che l'oggetto più popolare è scaricato il doppio rispetto al secondo più popolare, il triplo rispetto al terzo più popolare, ecc…

Il parametro α é ricavato **empiricamente** dai file di **log** del traffico Web.

Fitting di α come coefficiente angolare



Popolaritá ~ $1/i^{\alpha}$ $\rightarrow$ $\text{Log}_{10}(\text{Popolaritá}) \sim -\alpha\ \text{Log}_{10}(i)$

Performance a regime

Esistono formule analitiche che consentono di **stimare** da α le performance a **regime** (t→+∞) di un sistema di Web caching.

Nel caso piú semplice in cui α=1, la probabilitá che un oggetto sia referenziato entro k richieste dall'ultimo download è data da:

$$d(k) \approx \frac{1}{k \ln N}\left(\left(1-\frac{1}{N \ln N}\right)^{k}-\left(1-\frac{1}{\ln N}\right)^{k}\right)$$

dove N è il numero di oggetti Web di cui è composto il sito.

Previsione dell'hit rate – Stack Distance

La metrica della Stack Distance (SD) consente di stimare l'hit rate previsto per un proxy.

Partendo dai dati nei file di log, si va a vedere con quale frequenza si é fatto accesso a pagine visitate di recente.

La SD consente inoltre di **dimensionare** la cache di un proxy, in funzione della hit rate desiderata.

Algoritmo di calcolo della SD

Inizializzazione

Lo stack è inizialmente vuoto.

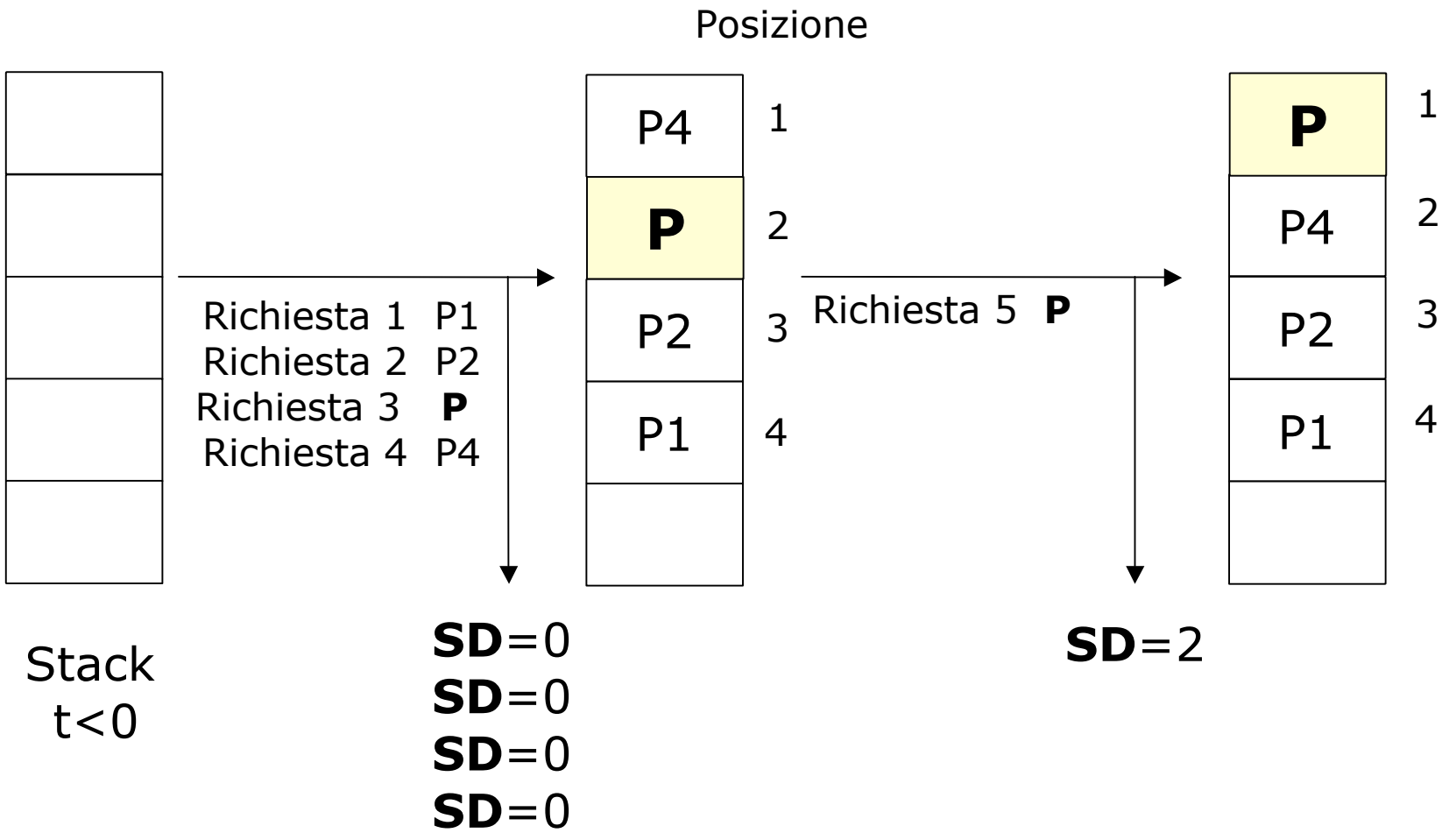
Algoritmo

1. Leggi la prossima richiesta dal file di log
2. Sia **P** la pagina di destinazione della richiesta
   1. se lo stack ***non*** contiene **P**, allora inserisci **P** in cima allo stack e PRINT SD=0
   2. Se **P** è ***presente*** alla posizione **k** dello stack, allora sposta **P** in cima allo stack e PRINT SD=**k**
3. Finché esistono altre richieste, torna al passo 1

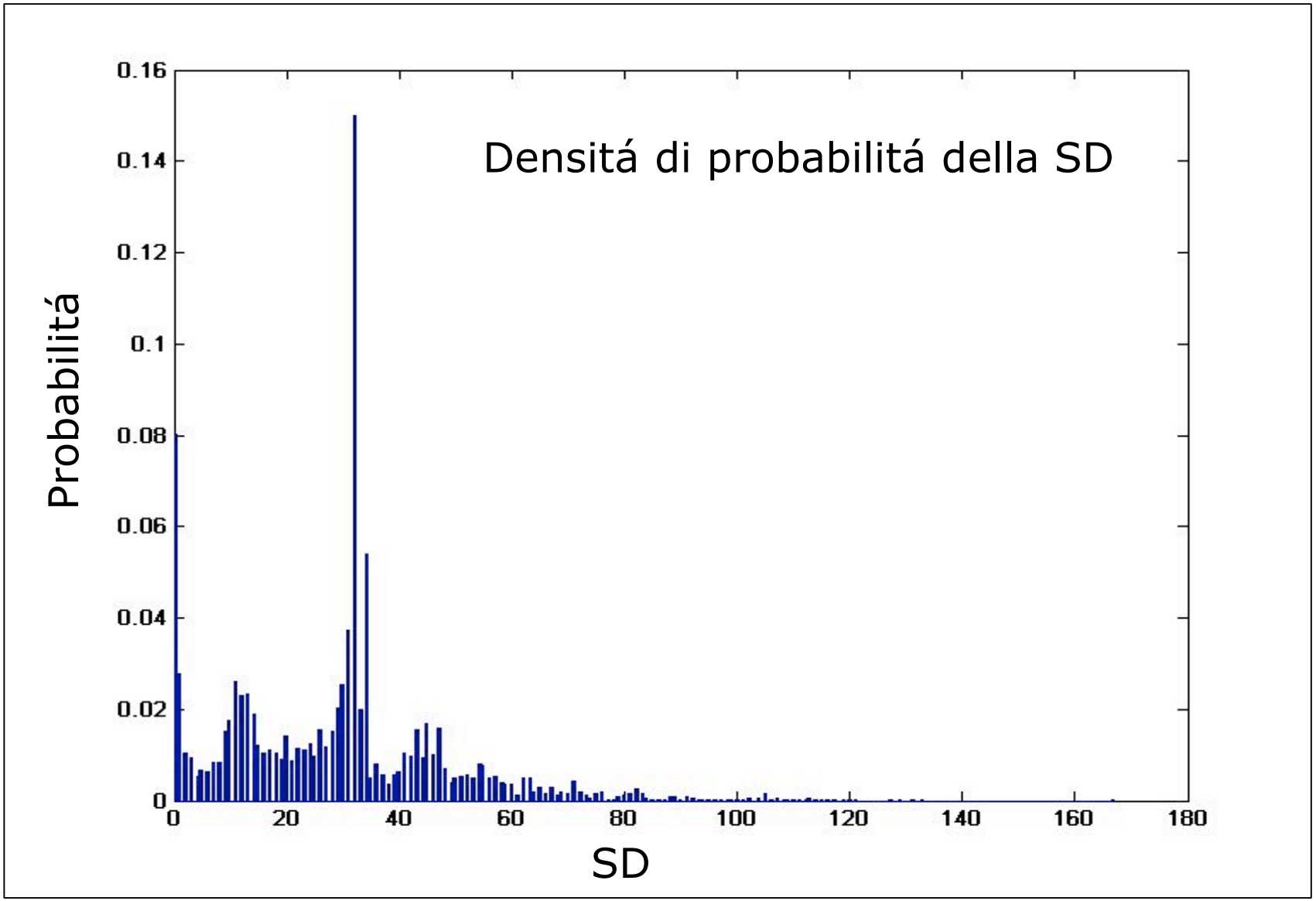
Risultati

I numeri stampati dalla PRINT costituiscono l'insieme dei valori osservati per la SD.

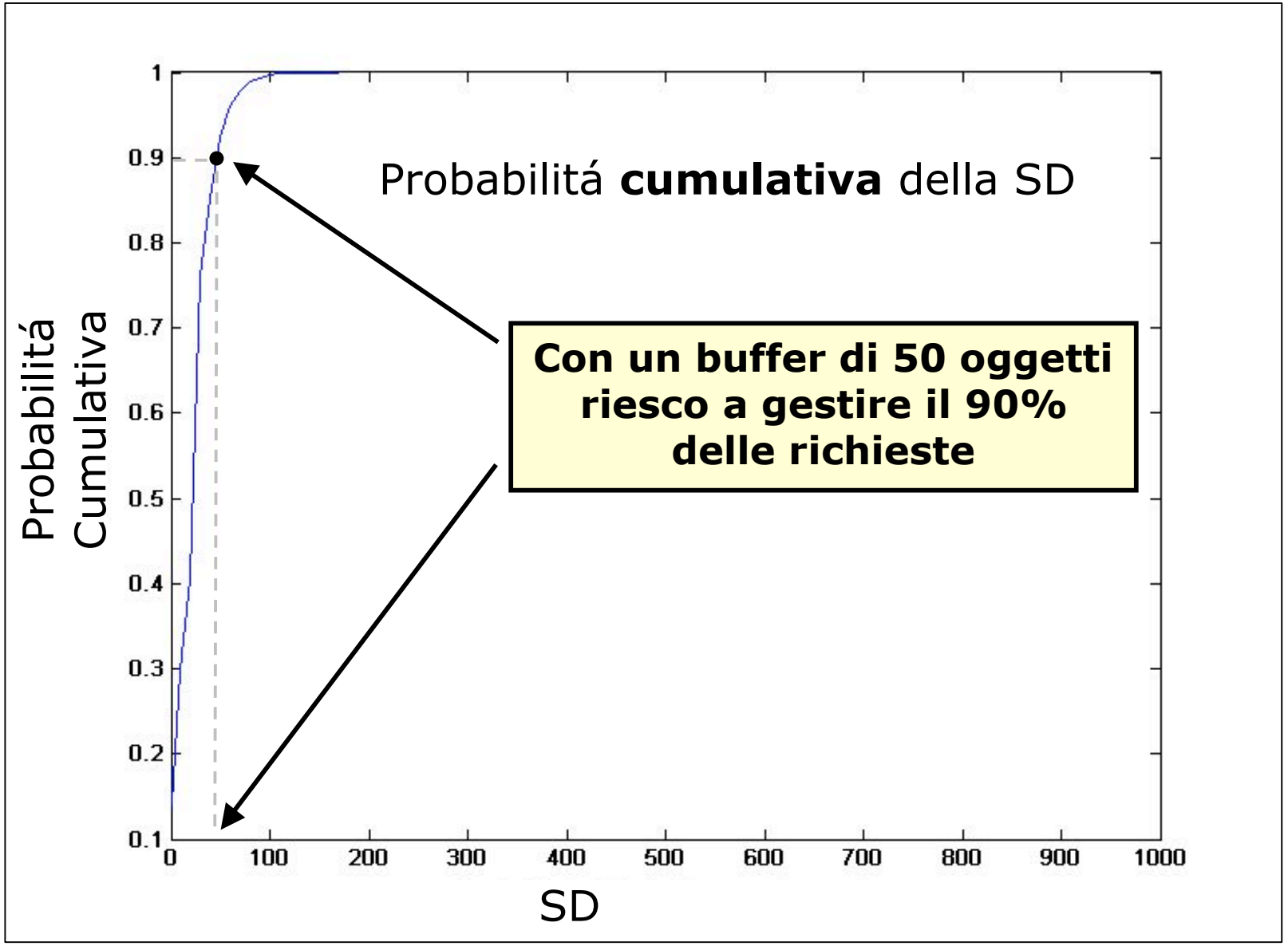
Stack Distance - Esempio



Densitá di probabilitá della SD



Dimensionamento della cache



Altre metriche di utilitá

Localitá spaziale

$$\mathbf{SL=} \frac{\text{Numero di path diversi presenti nei file di log}}{\text{Numero di path possibili}}$$

Change ratio

Alcuni studi indicano che il 15% delle richieste è diretto a pagine aggiornate di recente.

Richieste abortite

Nel calcolo delle prestazioni dei proxy si deve tenere conto di una percentuale tra il 5% e il 10% di richieste abortite.

Tematiche avanzate - Piggybacking

La possibilitá che un oggetto non sia aggiornato ha portato allo sviluppo di meccanismi per la validazione della consistenza (della cache).

Il metodo piú popolare si basa sul piggybacking:

1. Se una richiesta genera un cache miss
2. Il proxy inoltra la richiesta verso il server
3. un thread sul proxy aggancia in coda alla richiesta un treno di richieste di validazione per **altri** oggetti presenti in cache e scaricati in **precedenza** da quel server
4. tra questi ultimi, se un oggetto è stato modificato, allora è scaricato soltanto se è **abbastanza** popolare: la richiesta di validazione è eseguita con il metodo HEAD di HTTP che non obbliga al download dell'oggetto

Bibliografia e link di interesse

M. Arlitt, T. Jin - Workload Characterization of the 1998 World Cup Web Site- HP TR 1999
http://www.hpl.hp.com/techreports/1999/HPL-1999-35R1.pdf

M. Rabinovich, O. Spatscheck – Web Caching e Web Replication - Addison Wesley 2002

NLANR: **http://moat.nlanr.net**

SQUID: **http://www.squid-cache.org**