

Programmazione con i Socket TCP/IP

ciclo di lezioni tenute da Vittorio Ghini

e-mail: ghini@cs.unibo.it - tel: 0547.642826

ricevimento: nei giorni di lezione e su appuntamento

Scopo del corso è la familiarizzazione con:

- protocolli di rete a vario livello,
- tecniche di programmazione per sistemi distribuiti,
- strumenti per testing di algoritmi distribuiti,
- processo di progettazione di software sistemi distribuiti

Lecture consigliate:

- W. Richard. Stevens, ‘**UNIX Network Programming**’, volume 1
‘Networking APIs: Socket and XTI’, second edition, ed. Prentice Hall

Appunti dalle Lezioni, comunicazioni, progetti ecc. reperibili presso:

- <http://www.cs.unibo.it/~ghini/didattica/lab3/lab3.html>

Modalità d’esame:

Sviluppo di un Progetto Software, concordato col docente.

Tipologie degli Elaborati proposti

- elaborati “di studio”

Prevedono essenzialmente lo studio di alcuni strumenti di ausilio alla validazione di protocolli in rete, ed una parte progettuale non eccessiva.

- elaborati “di implementazione”

Implementazione di alcuni servizi standard (telnetd, ftpd, ...), o non (proxy con particolari caratteristiche). Non difficili, qualche volta lunghi.

- elaborati essenzialmente “di progetto”

Prevedono specifiche più ‘lasche’, richiedono una profonda fase di studio dei requisiti, ma un minore sforzo implementativo.

- elaborati “di ricerca e di progetto”

Richiedono in particolare uno sforzo per individuare politiche “ad hoc”, ma mantengono una consistente fase di progetto.

Criteri di Valutazione degli Elaborati

I principali criteri di valutazione dei progetti presentati saranno:

- **rispondenza ai requisiti,**
- **efficienza (teorica e sperimentale),**
- **modularità,**
- **leggibilità del codice,**
- **chiarezza e completezza di manualistica d'uso e relazione allegate**
- **innovatività delle soluzioni adottate**

Sintesi Elaborati Proposti:

•Rlogind	3 componenti
•Telnetd	4 componenti
•Ftpd	3 comp.
•Httpd	3 comp.
•Smtpt	4 comp.
•Nntpd	3 comp.
•Concast	4 comp.
•Proxy + prefetching	4 comp.
•DNS +round robin	4 comp.
•Proxy +4 politiche distribuzione	3 comp.
•Proxy + distrib.basata su storico delle performance	4 comp.
•Proxy + distrib.usando performance+prefetching	6 comp.
•Distribuzione di Carico Web usando REDIRECT e ..	4 comp.
•Uso Simulatore di rete cnet	5 comp.
•Uso Simulatore di rete ns	8 comp.
•multicast a livello applicazione (*)	6 comp.
•WAP (*)	10 comp.

(*) particolarmente difficili

Elaborati di Implementazione

Elaborati proposti:

implementazione dei servers per i seguenti servizi standard

rlogind	(RFCs 1123+1282), (http://lib.nevalink.ru/tcp_stevens/telnet.htm)	medio	(3 comp.)
telnetd	(RFCs 1123+854 856 857 858 859 860 1091 1073 1079 1372 1184 1408), (http://lib.nevalink.ru/tcp_stevens/telnet.htm)	lungo	(4 comp.)
ftpd	(RFCs 1123+959), (http://lib.nevalink.ru/tcp_stevens/ftp_file.htm)	medio	(4 comp.)
httpd	(RFC 2616)	medio	(3 comp.)
smtp	(RFCs 1123+821 822)	lungo	(4 comp.)
nntpd	(RFC 977)	medio	(3 comp.)
Internet Concast Service	(Sperimentale) (seminario, proposta implementazione, event. implementazione), (internet draft, draft-calvert-concast-svc-01.txt)	difficile	(4 comp.)

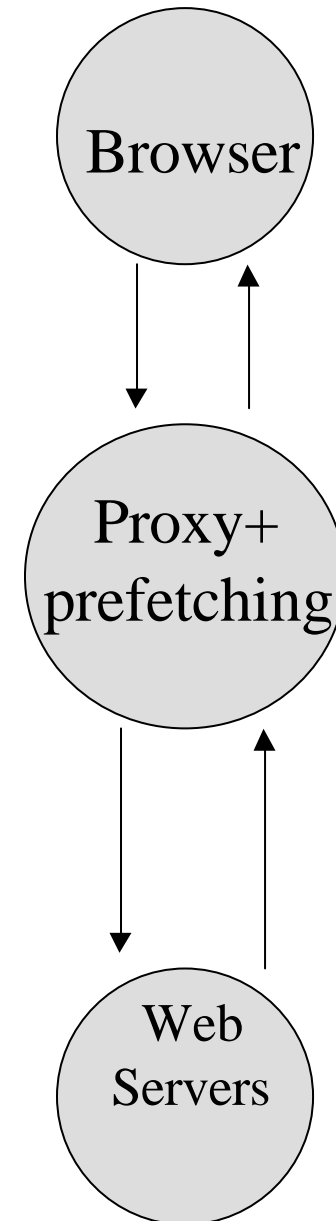
Prefetching di risorse Web

- il browser chiede al Proxy un certo URL,
- il Proxy chiede al Web server il documento
- se il documento e' di tipo HTML (o altri linguaggi con riferimenti), il proxy ne fa il parsing, ed estrae tutti i riferimenti a risorse web.
- Il proxy precarica (una porzione de) i files indicati dai riferimenti, secondo una politica da definire, allo scopo di minimizzare i tempi di risposta quando (se) il browser richiederà' quei files.

Elaborato proposto:

definire la politica di **prefetching**, e implementare il **proxy** che la effettua. Considerare soluzioni già presenti in letteratura.

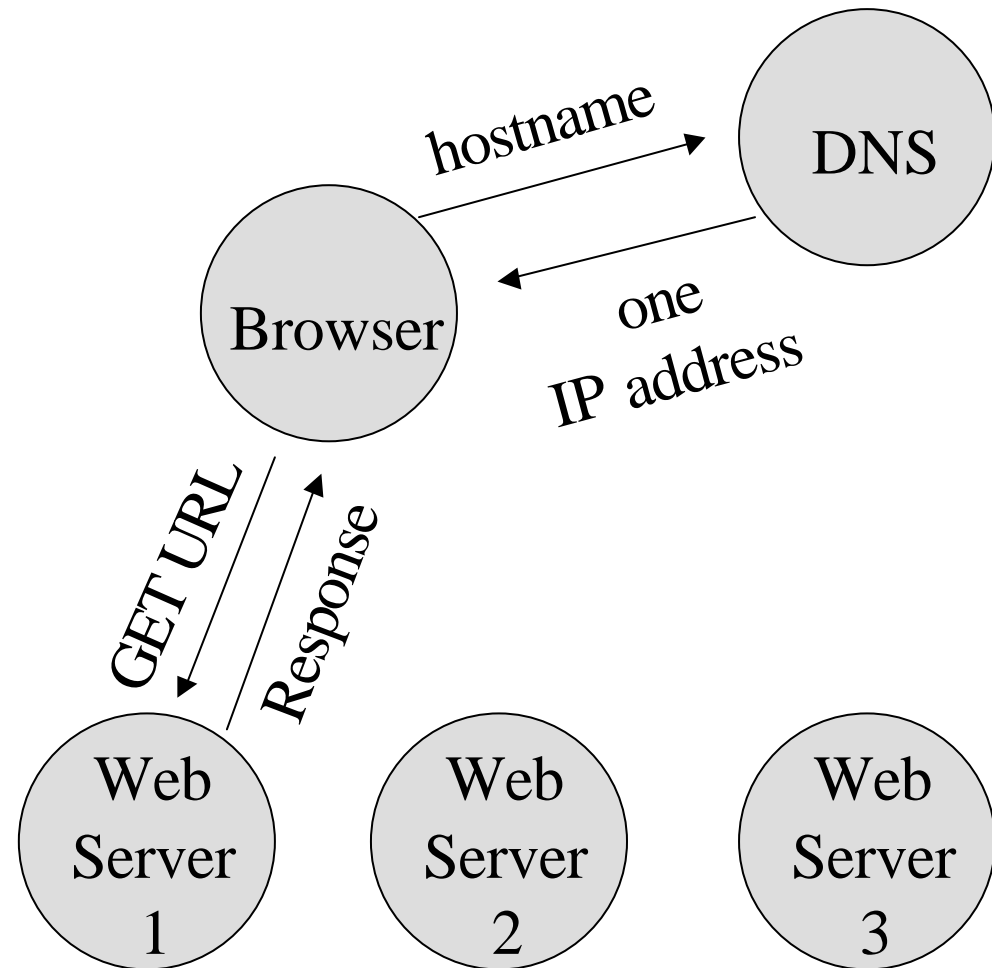
(medio-difficile) (3-4 comp.)



Distribuzione di Carico tra Web Server Replicati usando il DNS

Elaborato proposto:

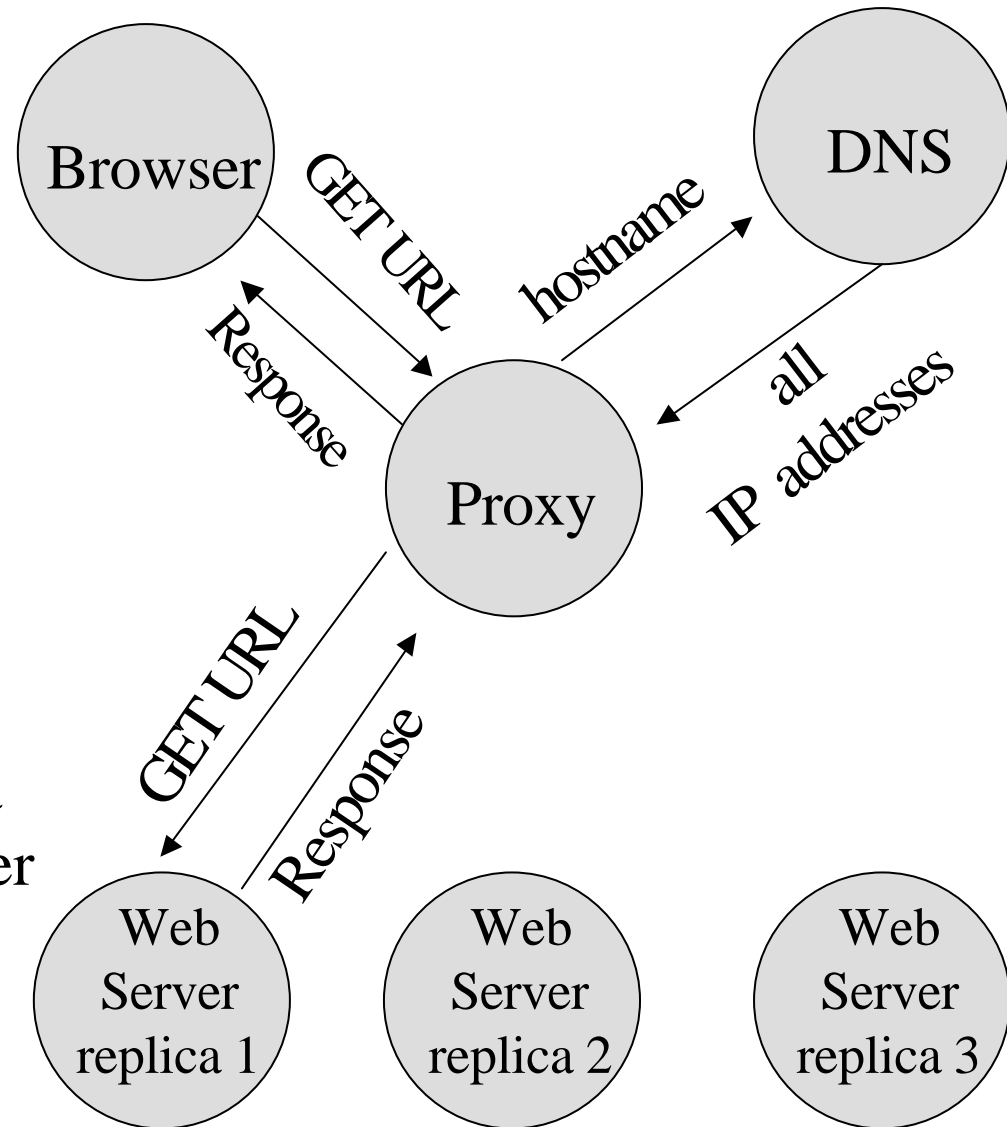
implementazione **DNS**
con round robin e TTL=0
(RFCs 1034-1035)
(medio-difficile)
(4 comp.)



web servers con stesso hostname,
stessi files, indirizzi IP diversi

Modello di Distribuzione di Carico dal lato Client

- il browser chiede al Proxy un certo URL,
- il Proxy ottiene dal DNS tutti gli indirizzi IP del Web server indicato nell'URL (le repliche),
- decide a quale replica chiedere il documento,
- effettua la richiesta alla replica
- restituisce il risultato al browser



web servers con stesso hostname,
stessi files, indirizzi IP diversi

Distribuzione di Carico tra Web Server Replicati sul lato Client

Elaborati proposti:

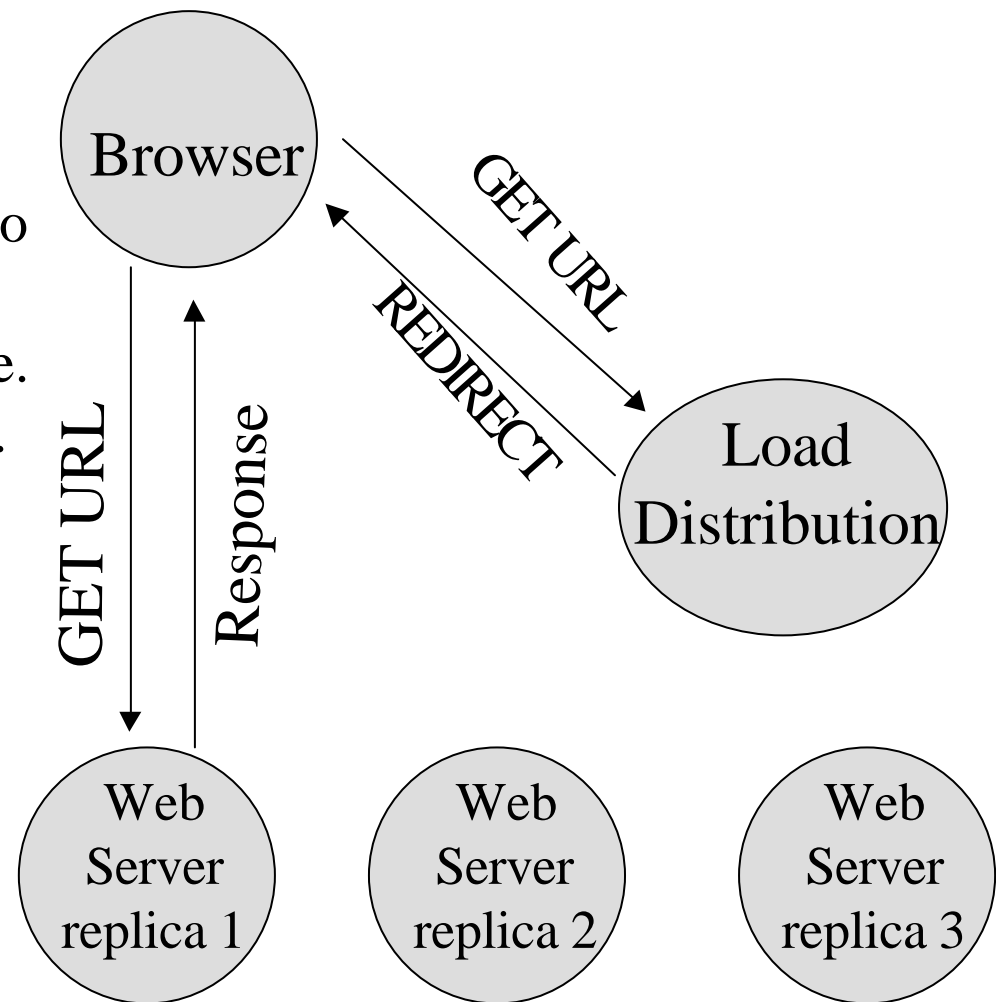
implementazione e valutazione sperimentale di :

- **Proxy server** (senza caching) che usa DNS e implementa **4 politiche** di distribuzione di carico: round robin, probing usando dummy HTTP requests, probing usando ping, probing usando hop distance. (medio) (3 comp.)
- **Proxy server** (senza caching) che usa DNS e implementa **politica mista** di distribuzione di carico: mantenimento di uno **storico delle prestazioni** delle diverse repliche in base a cui decidere a quale replica affidare la richiesta, con probing (ogni tanto) per le repliche non più usate. (medio-difficile) (3-4 comp.)
- **Proxy server** (per caching parziale) che integra distribuzione e **prefetching**: implementa politica mista di distribuzione di carico (vedi sopra) ed inoltre effettua prefetching dei riferimenti contenuti nei files ottenuti dai servers. (difficile) (fino a 6 comp.)

Distribuzione di Carico dal lato Server basato su HTTP REDIRECT

- il browser chiede un certo URL,
- il Distributore di Carico decide a quale replica assegnare la richiesta e utilizza il codice HTTP REDIRECT o TEMPORARILY MOVED per comunicare al Browser a chi chiedere.
- il browser chiede 'URL alla replica.

Elaborato proposto:
distributore di carico, mediante **HTTP REDIRECT** [o **TEMPORARILY MOVED**] che utilizza due politiche, un semplice Round Robin (per confronto) e una politica più intelligente proposta dallo studente (medio-difficile) (4 comp.)



web servers con stesso hostname,
stessi files, indirizzi IP diversi

Valutazione ed Uso di Simulatori di Reti a Pacchetto

Elaborati proposti:

- Studio delle features di un simulatore di rete a pacchetto tra quelli indicati,
- seminario in aula entro fine aprile,
- progetto di rete simulata (molti nodi) per sperimentazione algoritmi di routing a livello applicazione:

tools disponibili:

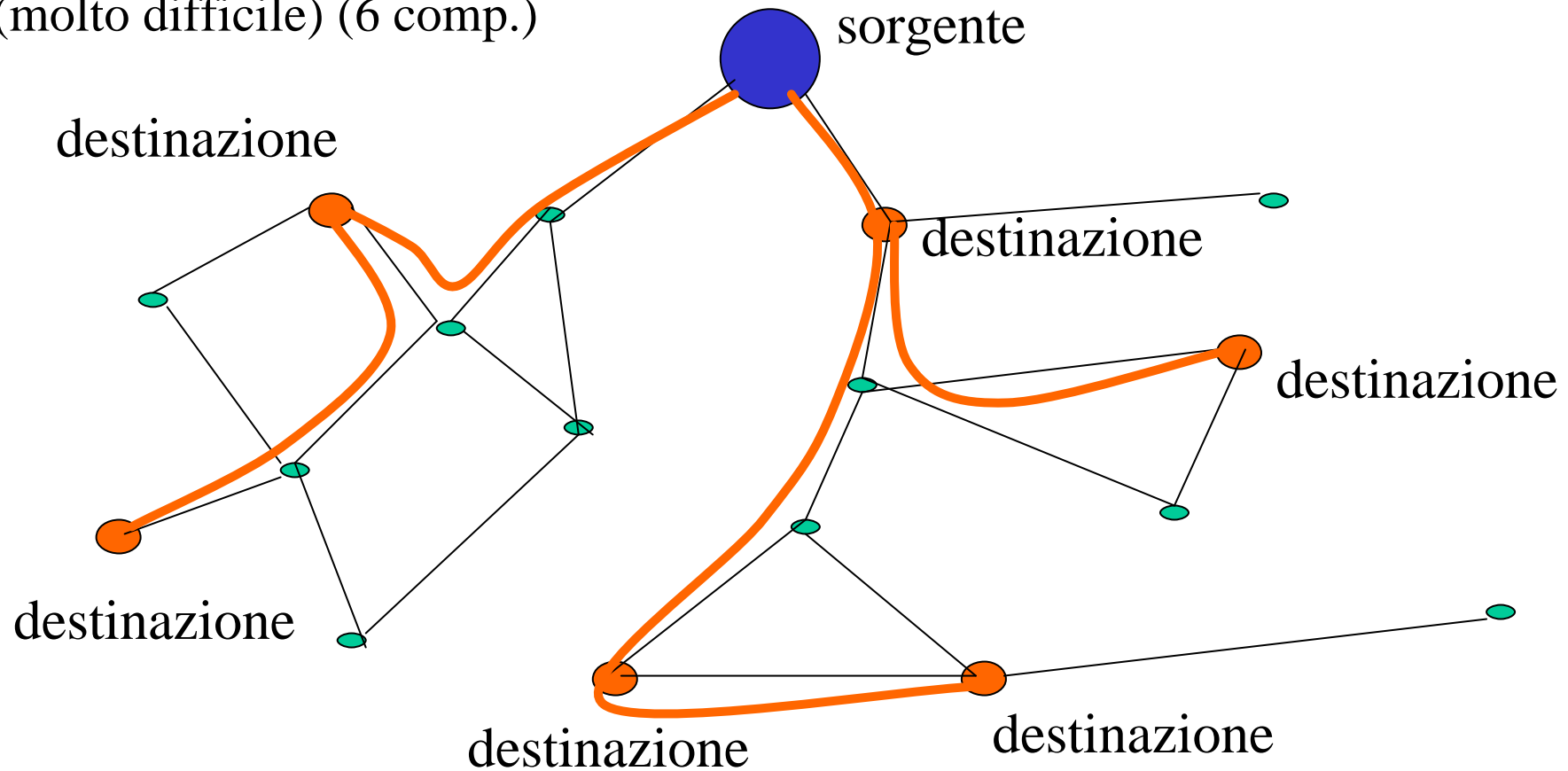
- **cnet** (<http://www.cs.uwa.edu.au/cnet/>)
(media difficoltà, lungo)
(5 comp.)
- **ns** (<http://www.isi.edu/nsnam/ns/>)
(difficile, lungo)
(fino 8 comp.)

Multicast a Livello Applicazione

Elaborato proposto:

Progetto, implementazione e valutazione sperimentale-simulativa di un meccanismo di **instaurazione percorsi per multicast a livello applicazione**, dedicato a streams TCP per trasmissione di flussi molto grandi, allo scopo di risparmiare bandwidth.

(molto difficile) (6 comp.)



Stack WAP

- Studio delle caratteristiche dello stack **WAP**,
- seminario in aula,
- proposta per un'implementazione di un sottoinsieme dello stack WAP (basato su UDP?)

<http://www.wapforum.org/>

(molto difficile e vasto)

(fino a 10 componenti)