MPLS-based Request Routing

MPLS-based Request Routing

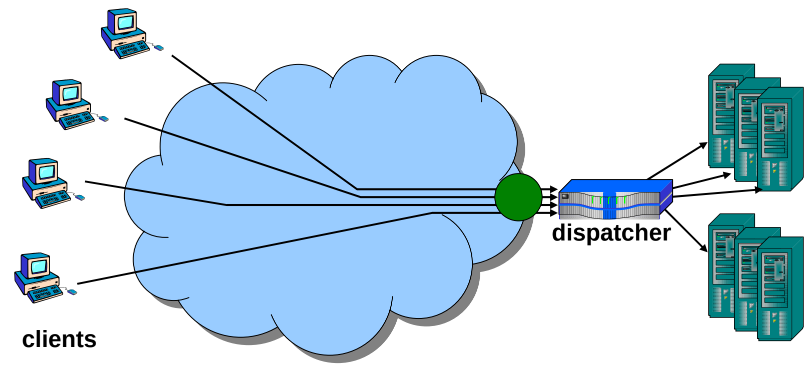
Arup Acharya, Anees Shaikh, Renu Tewari, Dinesh Verma

IBM TJ Watson Research Center

High volume Internet data centers

Web server cluster + front-end dispatcher

- direct requests based on server load, requested content, client identity, etc.



Current dispatcher technology

Layer-4 dispatchers

- route requests based on TCP/IP headers
- high-performance h/w implementations
- functionality limited to load balancing or simple affinity

Layer-7 dispatchers

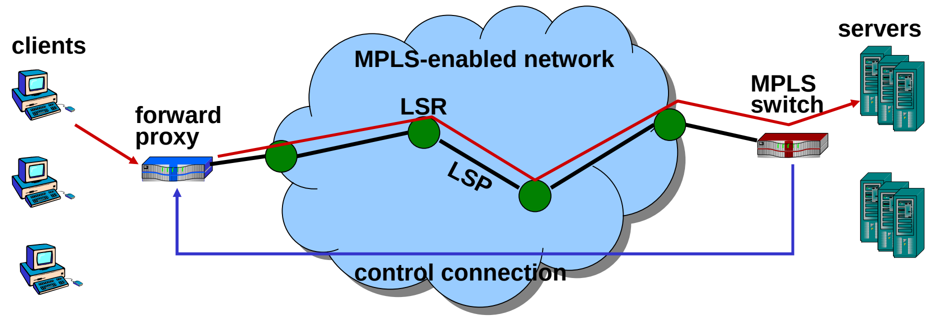
- use application information (e.g., HTTP headers)
- sophisticated functionality – content-based routing, affinity, load-balancing
- scalability and performance limited by TCP connection termination
  - Application-level gateway
  - TCP splicing
  - TCP connection handoff

Desired solution

- sophisticated functions and flexibility
- high-performance

MPLS-based architecture

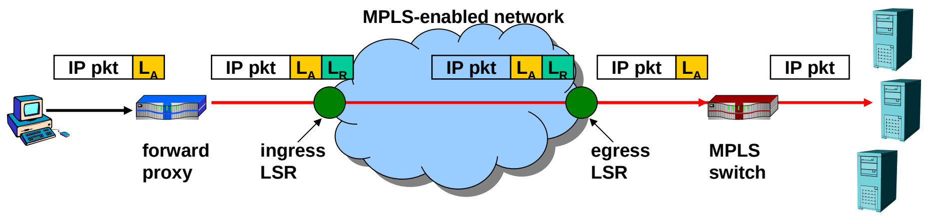
- MPLS provides a circuit-switching service over a hop-by-hop IP network
- Architecture components
  - MPLS network
  - MPLS-enabled client-side proxy
  - MPLS switch acting as dispatcher



MPLS label stacking

label stacking:

- Labels typically used for expressing routing policies
- Use label stacking to push application-layer label
- Outer routing label used for switching in the network
- MPLS-enabled server network further improves performance



Label distribution to proxies

Persistent control connection between dispatcher and proxy

- content-based routing: URL→label mapping
- load balancing: labels, weights, and policy
- client affinity: labels and timeout, start/stop URLs
- service differentiation: per-service class label set (e.g., “gold”, “silver”, “bronze”)

Dispatcher populates layer-2 label table

Deployment issues

Need wide MPLS deployment in core *and* edge

- supported by reports from large ISPs and IP service equipment vendors (e.g., for VPNs)

Why install an MPLS-enabled proxy?

- data center and proxies in same administrative control (ISP with hosting service)
- ASPs with large enterprise customers and SLAs
- Intranet and extranet servers

Scaling to many proxies and web sites

- limit proxy participation to high-volume client sites
- proxies may initiate with selected, popular sites

Summary

Key advantages

- leverage growth of MPLS deployment in core and edge networks
- removes primary bottleneck of TCP termination
- realization in standard off-the-shelf switch hardware
- implements sophisticated request routing functions

Requirements

- assign some request-routing functionality to proxies
- MPLS-aware proxies at the network edges
- implementation of control protocol for label distribution

Literature and acknowledgements

MPLS-based Request Routing (R&D Synopsis)

A. Acharya, A. Shaikh, R. Tewari, and D. Verma, *Proc. Int'l Workshop on Web Caching and Content Distribution (WCW '01)*, June 2001. Extended version published as IBM Research Report RC 22275

http://www.research.ibm.com/people/a/aashaikh/papers/rc22275.pdf

In MPLS World News

http://www.mplsworld.com/archi_drafts/focus/analy-ibm.htm

Thanks to Anees Shaikh and Arup Acharya for providing their presentation!

Content Distribution in the WWW

Motivation & Classification

Web Caching

Content Distribution Networks

- Techniques
- Performance

Content Distribution in the WWW: Motivation

WWW users use HTTP to retrieve web objects from a server

Response time can be slow ("World wide wait"):

- Low-speed path causing low transmission delay
- One or more congested links cause queuing delay and packet drops
- Web server is overloaded

Strategy:

- Replicate server content
- Direct client to "best server"

Content Distribution in the WWW

Content distribution refers to mechanisms for:

1. Replicating content on multiple servers in the Internet
2. Providing end systems means to determine the servers with fastest response

Large industry:

- Cisco, Lucent, Inktomi, CacheFlow etc.: provide hard-and software
- Akamai, AT&T etc.: provide content distribution services to providers such as CNN and Yahoo

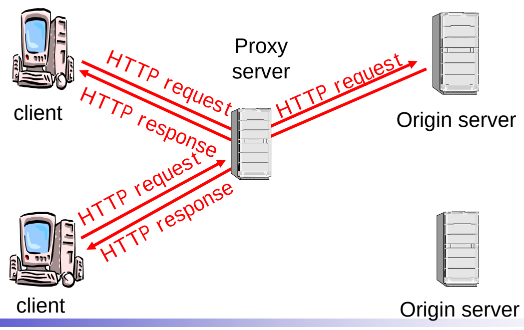
Classification:

- Web caching
- Content distribution networks
- P2p file sharing (extra lectures)

Web caching

A web cache (proxy server) is a network entity that satisfies HTTP requests on the behalf of an origin server

Cache is both a client (to the origin server) and a server (to the clients)



Web caching: Motivation

Reduce latency by avoiding slow links between client and origin server:

- low bandwidth links
- congested links

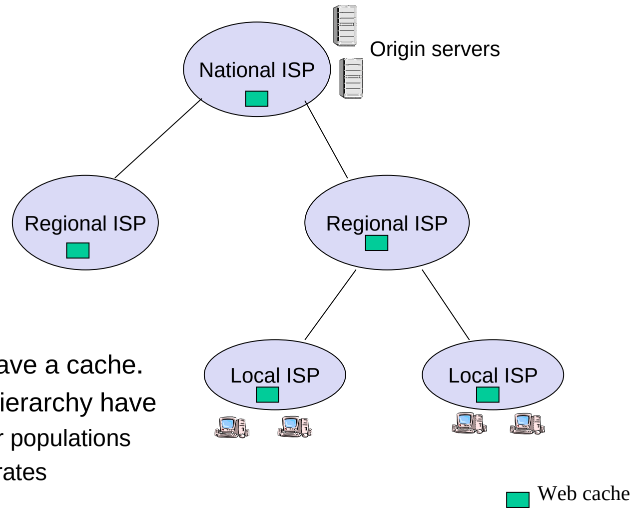
Reduce traffic on links

- between institutional network and regional ISP.
- Reduce traffic on transoceanic links.

Spread load of overloaded origin server to caches.

- An Internet dense with caches allows a content provider to offer high-performance distribution at low cost.
  - Inexpensive server
  - Low-bandwidth Internet connection

Design Techniques: Hierarchical Caching



Each ISP can have a cache.

ISPs higher in hierarchy have

- larger user populations
- higher hit rates

Cooperative Caching

Multiple sibling caches within a single ISP.

One or more of the siblings could contain the requested object.

Cooperation:

- ICP (Internet Cache Protocol): siblings send messages to each other to find a copy of object (Intercache communication)
- CARP (Cache Array Routing Protocol): URL space is partitioned (Hash-based Request Routing)
- Can have cooperating sibling caches in each ISP in each tier of a hierarchy.

Caching: Other caching terms

Reverse proxy caching:
- Caches close to the origin server. Independent of client-side proxy caching

Active Caching:
- Aims at caching dynamic content, e.g. personalized content

Content Prefetching:
- Retrieve data from remote servers in anticipation of client requests

Cache Digest:
- A summary of the contents of an Internet Object Caching Server

Well-known cache systems:
- Harvest: introduced the idea of hierarchical caching (mainly for FTP)
- Squid: extended Harvest for HTTP, introduced ICP

September 11 with Web caching

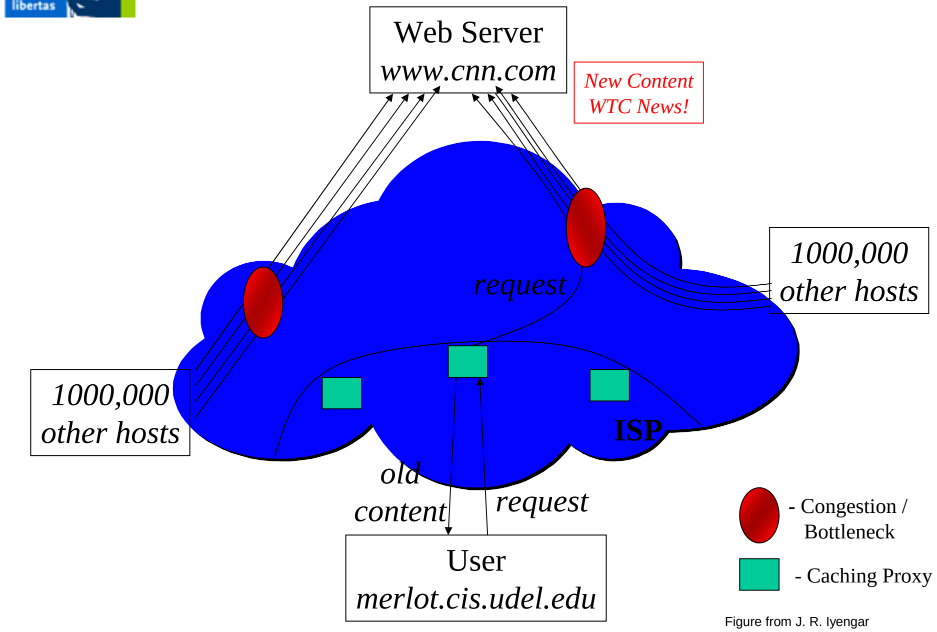


Figure from J. R. Iyengar

Content Distribution Networks

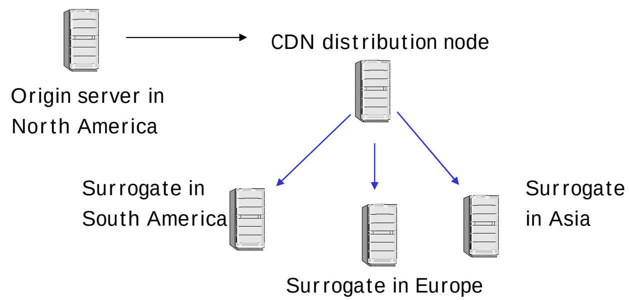
Content distribution networks (CDNs) are a mechanism to deliver content to end users on behalf of origin web sites.

CDNs consist of a collection of surrogates (non-origin servers) that attempt to offload work from origin servers by delivering content on their behalf.

For each request, the CDN locates a surrogate "close" to the client that serves the request. Different notions of close are:
- Network proximity
- Bandwidth availability
- Availability of content
- Low latency, e.g. choosing a lightly loaded server (or not heavily loaded server)

CDNs vs. Web caches

A CDN can be regarded as a set of widely-dispersed caches but there are two major differences to web caches:
- Surrogates are coordinated by a mechanism that routes client requests to good surrogate
- Surrogates are potentially populated by other means than requests by clients



September 11 with CDN

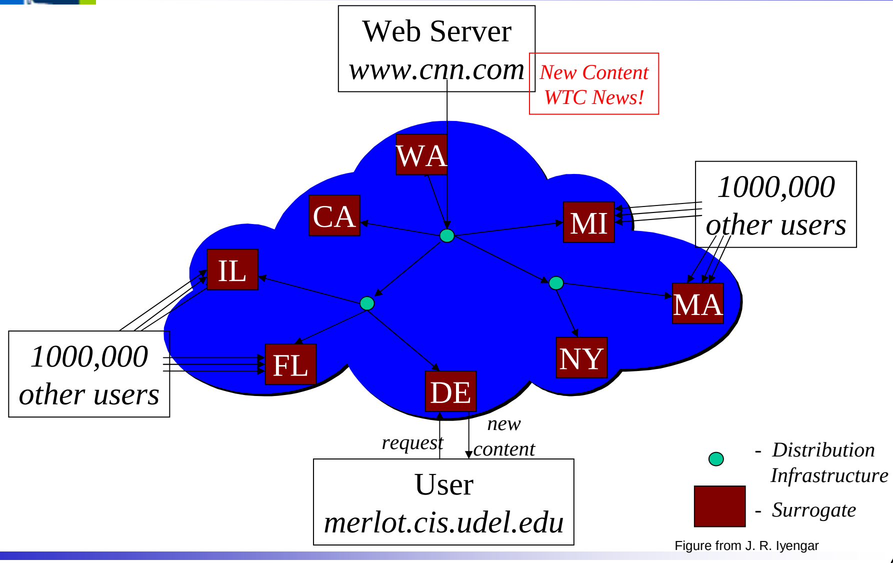


Figure from J. R. Iyengar

CDN Techniques

There are two major techniques for redirecting client requests for objects served by the CDN to a particular CDN server (sometimes called Request Routing):

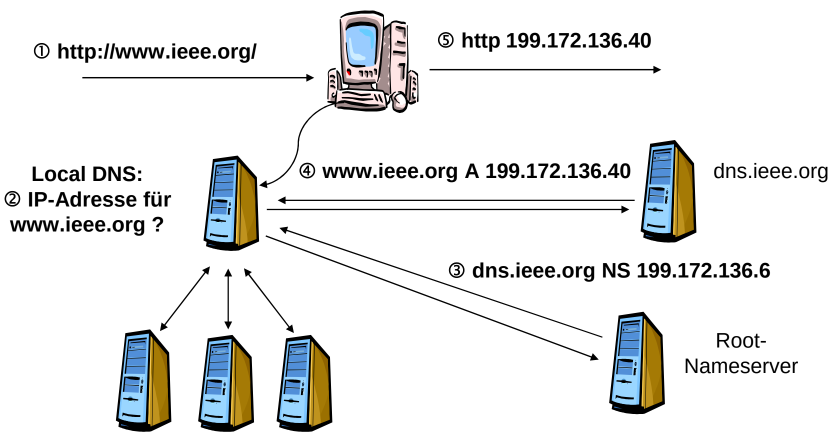
1. DNS redirection
   a. Full-site content delivery
   b. Partial-site content delivery
2. URL rewriting

Other techniques are:

- Anycast (does not consider surrogate load)
- Transport-layer request routing: can be used in combination with DNS redirection.

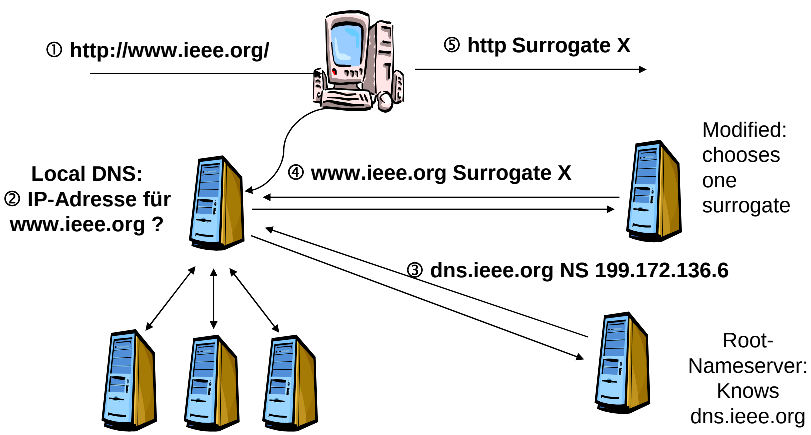
DNS Redirection

Normal DNS operation:



DNS Redirection

Modified DNS:



DNS Redirection

Advantages:

- Simple – no changes to existing protocols, clients or servers
- General – works for all IP-based applications, independent of transport protocol used

Full- and Partial-site Content Delivery

Full-site:

- All requests to the CDN are redirected via DNS
- Surrogates either serve content from their cache or forward requests to the origin server
- Used by Adero, Netcaching, ..

Partial-site:

- Origin sites modifies the embedded URLs for objects (images) so that these URLs are resolved by the CDN's DNS server
- Actual syntax varies with the CDN. Speedera changes www.foo.com/bar.gif to foo.speedera.net/www.foo.com/bar.gif

DNS redirect and TTL

DNS resource records contain Time to Live field that specifies how long a client may cache a resource record.

RFC 1912 recommends TTL values of 1-5 days

- Nameservers typically use a TTL of 1 day.

The DNS of CDNs have very small TTL values of 10-200 seconds

- Aim: better load balancing

Problems with small TTL values

Small TTLs lead to problems:

- Clients must perform DNS lookups more frequently
  - This can increase client latency
- Increased load on DNS
  - It has been observed that in many cases the time between the HTTP GET request and the arrival of the first data packet accounts for 30-40% of the response time, the main reason is the bad performance of DNS

Effectiveness of DNS-based Server Selection

Study by A. Shaikh, R. Tewari and M. Agrawal:

- Without careful TTL tuning, client latency can increase
  - In particular, when web pages contain more embedded objects
- Typical client-nameserver distance is 8 or more hops. Furthermore clients and nameservers often have disjoint paths to surrogates.
  - Latency to nameserver is poor indicator of latency to client

URL Rewriting

URL Rewriting:

- Origin server rewrites URL links as part of dynamically generating pages to redirect clients to different servers.
- At resource access time, the page is dynamically rewritten with the IP address of one of the surrogates, avoiding the need for a DNS lookup.
- Problem: First request must be served from origin server

Hybrid approach:

- Use URL rewriting to identify a particular server that might resolve to the IP address of another surrogate

Performance of CDNs (1)

Study by Johnson et al. that evaluated performance of two CDNs (Akamai and Digital Island)

1. CDNs are able to succesfully provide services by avoiding significantly "bad" services as opposed to being able to pick the best ones.
2. CDN's occasionally make bad choices in picking servers for clients that have measured latencies worse than going to the original client thereby degrading service for client rather than improving them.
3. The use of CDN's actually does improve performance on average when considering both performance using the origin server as well as comparing the choice of server to other possible choices.

Performance of CDNs (2)

Study by Krishnamurty et al.:

- CDNs offer much better performance than origin servers
- Significant differences in download times between different CDNs
- Compared the download time for a newly obtained surrogate to a fixed and the previous surrogate (i.e. effect of low TTL values):
  - In almost all cases, the response time was better using the previous or fixed server
  - Indicates that even worst-case client response time is generally not improved with a DNS lookup to find a new server
  - Confirms the findings by Shaikh and Tewari that careful tuning of TTL values is important (and difficult)

Literature

- B. Krishnamurthy, C. Wills, and Y. Zhang . “On the Use and Performance of Content Distribution Networks” Proceedings of SIGCOMM IMW 2001, California, November 2001.
- A. Shaikh, R. Tewari, M. Agrawal. “On the Effectiveness of DNS-based Server Selection”, Proc. IEEE INFOCOM 2001, April 2001.
- K. Johnson, J. Carr, M. Day, and M. F. Kaashoek. “The measured performance of content distribution networks”. 5th International Web Caching and Content Delivery Workshop, Lisbon, Portugal, May 2000.
- G. Barish and K. Obraczka. World Wide Web Caching: Trends and Techniques. IEEE Communications Magazine Internet Technology Series, May 2000.