B211 Internet Computing

Web Proxies

Learning Objectives

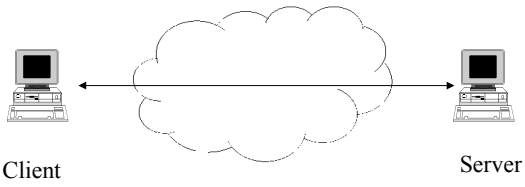
1. Understand the purpose of having web proxy servers.
2. Understand the basic technical operations of web proxy servers.

Lecture Outline

- What are proxies?
- Why use Proxies?
- Caching Proxies
- Non-HTTP Gateways
- Reverse Proxies

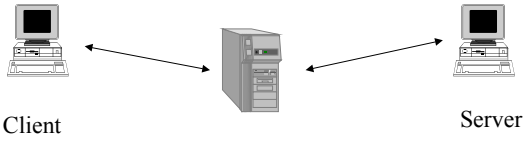
Intermediaries in Network Communication

- All the communications we have talked about up till now consist a client communicating directly with a server at two ends of the Internet.



Intermediaries in Network Communication

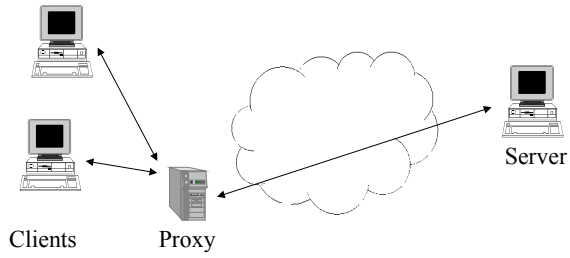
- In client-server communications these days, this direct connection is becoming rarer. We usually have some intermediate agent (program/machine) acting in between the client and the server to forward messages.

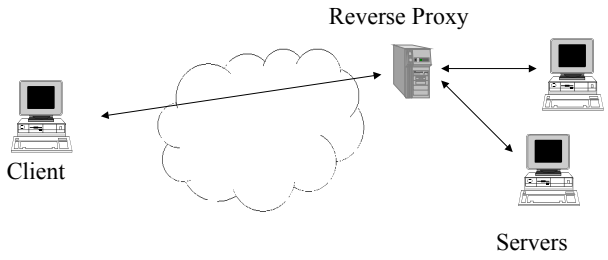


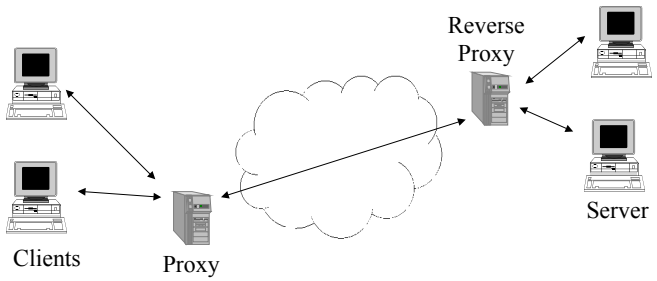
- This intermediary is usually called a *proxy*, or a *proxy server*.

Intermediaries in Network Communication

- Proxies can sit on the client side, or on the server side.







Proxies and Firewalls

- Proxies came into prominence with the widespread use of the Web (HTTP).
- Early proxy functions came from having to deal with *firewalls*.
  - A *firewall* is a piece software that restricts and controls messages going in an out of a machine or a network for security reasons.
    - » Eg. it can restrict what messages a client can send, which machine's messages to let through, etc.
  - A proxy was put in front or behind the firewall to give users all the functionalities of the web, but still being protected by the firewall.
  - Early use of the word "proxy" was in this role as a "gateway" to external services.

Proxies and Web Proxies

- Note that any intermediary between clients and servers can be called a proxy.
  - Eg. SOCKS (http://www.socks.nec.com/) is an important low protocol and system to support proxying any TCP/IP application service.
- In this lecture, we will focus on far-and-away the most popular one of all, the web proxy.

Why use Proxies?

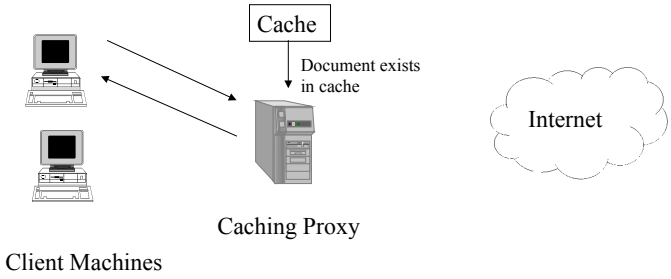
- There are various reasons why we would have an proxies between clients and servers:
  - Caching messages
  - Sharing access
  - Anonymizing clients
  - Transforming Messages
  - Filtering
  - Gateway to non-HTTP messages

Caching Messages

- Caching received messages (eg. HTTP responses) is one of the primary purposes of having proxies.
- Proxy caching can significantly decrease network traffic and costs when
  - multiple clients within a network accesses the same resources for the same originating server, and/or
  - a client do not do its own local caching.
- Caching may not be appropriate in some circumstances:
  - Search engine robots should always get direct copies from originating servers.
  - Dynamic server-side content like CGI should not be cache.

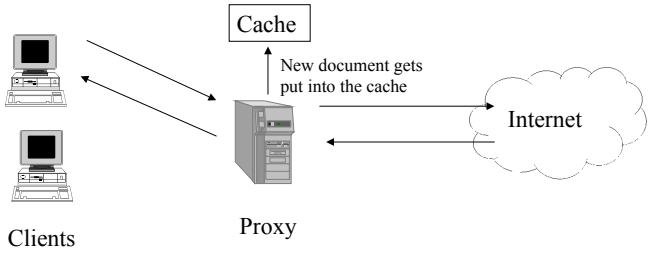
How Caching Proxies Work:

- If a document exists in a the proxy's cache:



How Caching Proxies Work:

- If a document doesn't exist in a the proxy's cache:



Configuration for Caching

- Caching proxies are configured to decide:
  - Which documents are used frequently enough to justify caching.
  - Which documents in cache to overwrite if the cache becomes full.
  - When to connect to the originating server to retrieve documents.
  - When documents get out-of-date (a document that is not out of date is called "*fresh*").
  - etc.
- Messages like HTTP responses usually contain cache-control headers that proxies can use to determine how long to keep a certain document.

Active and Passive Caching

- Passive Caching
  - proxy waits for local clients to make requests, then cache the appropriate retrieved documents.
- Active Caching
  - during low activity, proxy fetches documents it believes local clients would want.

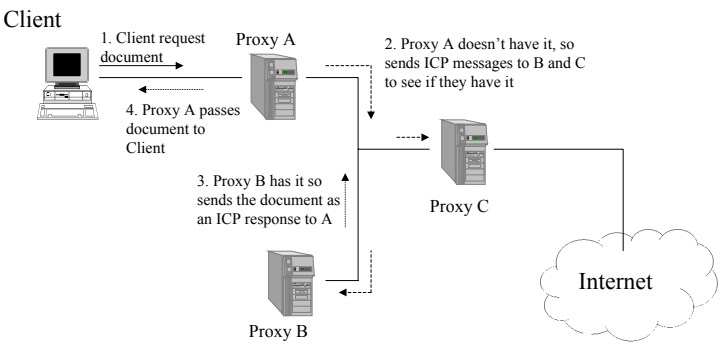
Handling Multiple Caches

- Some LANs have more than one caching proxies.
- Proxies in these environments must be able to effective exchange cached data.
- Protocols for managing inter-proxy communications:
  - Internet Cache Protocol (ICP)
  - Cache Array Routing Protocol (CARP)

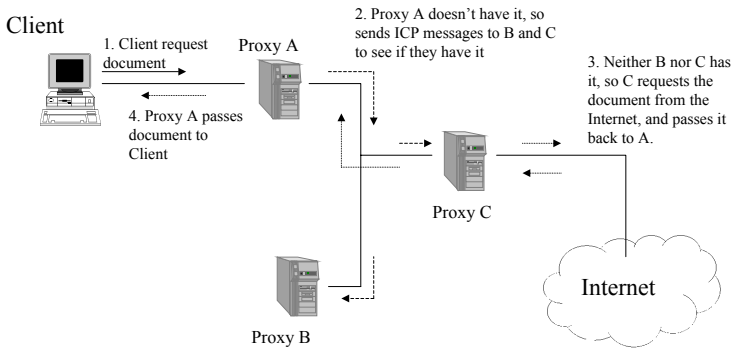
The Internet Cache Protocol (ICP)

- ICP specifies a message format for communication between proxies
- Exchange information about whether a certain page exists on a certain proxy's cache
- Proxies uses 3 ports:
  - For HTTP requests by clients
  - For exchanging ICP messages with other proxies
  - For retrieving pages stored on another proxy's cache

An example of how ICP works:



Another example of ICP:



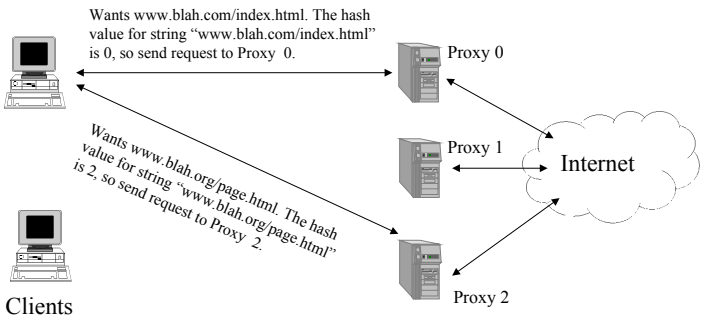
How CARP works:

- All proxies in the environment arranged as an "array".
- All URLs are designated to belong to one and only one proxy in this array, based on calculating a hash function.
- Clients determine which proxy "owns" the URL (by computing the hash function) and queries the particular proxy.

How CARP works:

- The specified proxy either
  - sends the requested document to the client if it has it in its cache, or
  - fetches the document from the URL address
- Advantages:
  - All clients can calculate exactly which proxy server handles which URL.
  - Document corresponding to a single URL is not duplicated on multiple proxies.
  - The load is distributed evenly (if the the hash function is good) between all proxy servers.

How CARP works:



Sharing Access

- Proxies can allow sharing of connections to web resources.
- Eg.
  - When multiple clients access the same resources on the same originating server, the proxy can make one connection to serve them all.
  - When multiple clients makes access the different resources, but still on the same originating server, the proxy can "regulate" the connections - if there is a delay with one request to that server, it makes sense to delay another request to the same server.

Hiding Client Information

- Requests from proxies do not contain the header data from the clients, servers cannot track as much information as before.
  - Eg. Getting client IP addresses, browser versions through User-Agent fields, using cookies to store user information, etc
- This function of *anonymizing* clients and users is becoming more and more important with today's emphasis on privacy.

Transforming Messages

- Proxies can also be configured to transform requests and responses based on what they know about the clients.
- Eg.
  - Client only supports HTTP/1.0, but originating server have extra capabilities in HTTP/1.1. Proxy can communicate more efficiently with an originating server in HTTP/1.1, but transform the messages back to HTTP/1.0 when sending the result back to the client.
  - Client is on a low bandwidth connection. Proxy can compress the messages before sending it to the client.

Filtering

- Proxies can be configured to filter improper requests and responses.
- Some example filtering:
  - Reject requests to inappropriate URLs.
  - Remove inappropriate search strings from query-strings to search engines.
  - Removes identifying header fields (like user email addresses).
  - Check responses for viruses.

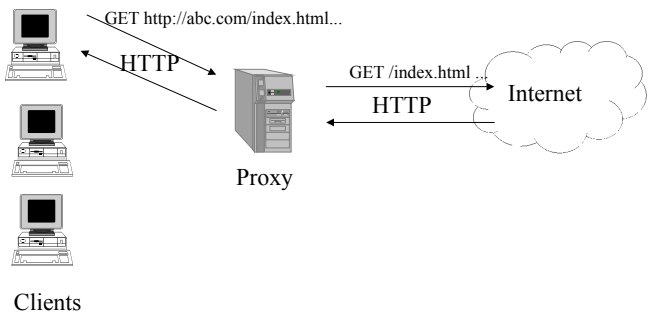
Gateway to non-Web Services

- A web proxy can give seamless access to other non-HTTP services available on the Internet.
  - Clients that only communicate in HTTP can use the proxy to communicate with these other services.

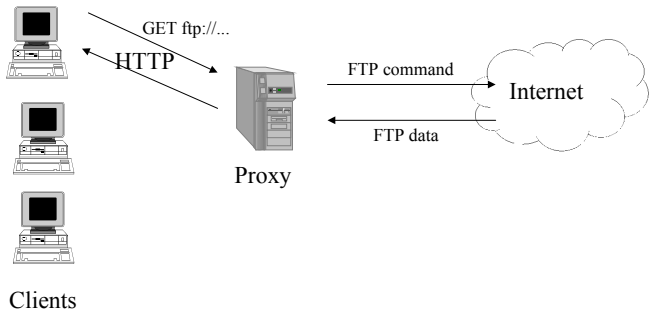
Acting as a Gateway

- The process:
  - Client sends a request to the proxy (usually on its own local LAN) using HTTP.
  - Proxy server retrieves the documents using an appropriate protocol (HTTP, FTP, gopher, etc), if not there already.
  - Proxy server sends the information back to the client in HTTP.

A HTTP Request using a Proxy:



An FTP Request using a Proxy:



Reverse Proxies

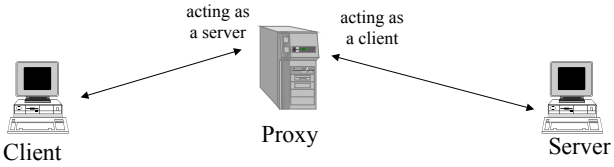
- Proxies were initially used mainly on the client-side to aggregate clients, as described up to now.
- Today, there are *reverse proxies* existing on the server-side to help with the traffic on multiple servers.
    - Instead of accessing servers directly, clients or client-side proxies access these reverse proxies to make their requests.

Why Reverse Proxies?

- The reasons for having reverse proxies (some similar to the purposes normal proxies to a collection of clients):
    - Distribute and balance the load on a collection of servers.
    - Cache the responses for future requests to the same resources.
    - Hides the servers and decrease the vulnerability of the servers to direct security attacks.

Acting as a Client and a Server

- The proxy, although commonly called a proxy *server*, is really acting as a client and a server at the same time.



Acting as a Client and a Server

- In its role as the intermediary, the proxy must maintain a large amount of state information to effectively keep the bidirectional communication going. Eg.
    - If a client aborts a request, the proxy must stop it's own request to the originating server. But if the response has already arrived, then it must cache but NOT send the response back to the client.
    - Handle cookie header problems when two clients accesses the same cookie-enabled web resource at the same time - cookies from servers are usually only meant for one particular client.

Proxies and Browsers

- Browsers can be set to directly connect to proxies.
- All ISPs have established proxies their users can voluntarily connect to.

Transparent Proxies

- In some cases, ISPs want:
  - to force subscribers to use the proxy, whether they want to or not.
  - subscribers to use a proxy, but don't want them to know.
  - clients to be proxied, but don't want to go to all the work of updating the settings in hundreds or thousands of web browsers.
- ISPs can set-up *transparent proxying*
  - All HTTP communications by their subscribers are proxied without anything any special settings on the subscribers' machines.
  - As far as the client software knows, it is making direct connections to originating servers.

Example Proxy Software:

- Some example proxy software:
  - Squid (http://www.squid-cache.org/)
  - Microsoft's Internet Security and Acceleration (ISA) Server (http://www.microsoft.com/isaserver/default.asp)
  - Netscape Proxy Server (http://wp.netscape.com/proxy/v3.5/datasheet/)
- For a longer list, see example:
  - http://directory.google.com/Top/Computers/Software/Internet/Servers/Proxy/