CS 6/79995

Advanced Internet Systems

Kent State University
Dept. of Computer Science

LECT-3

Today's Topic

Internet Caching Technology

INTERNET ENGINEERING

Definitions

- Client
  - A program that establishes connections for the purpose of sending requests.
- User agent
  - The client which initiates a request. These are often browsers, editors, spiders (web-traversing robots), or other end user tools.

INTERNET ENGINEERING

More Definitions

- Server
  - An application program that accepts connections in order to service requests by sending back responses. Any server may act as an origin server, proxy, gateway, or tunnel, switching behavior based on the nature of each request.
- Origin server
  - The server on which a given resource resides or is to be created.

INTERNET ENGINEERING

More Definitions

- Proxy
  - An intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients. Requests are serviced internally or by passing them on, with possible translation, to other servers. A proxy must implement both the client and server requirements of this specification.

INTERNET ENGINEERING

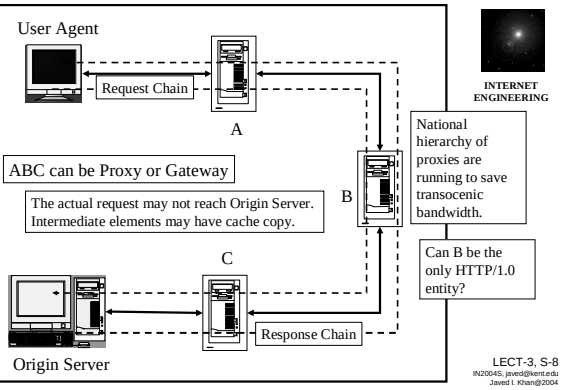
More Definitions

- Gateway
  - A server which acts as an intermediary for some other server. Unlike a proxy, a gateway receives requests as if it were the origin server for the requested resource; the requesting client may not be aware that it is communicating with a gateway.

INTERNET ENGINEERING

More Definitions

- Tunnel
  - An intermediary program which is acting as a blind relay between two connections.
  - Once active, a tunnel is not considered a party to the HTTP communication, though the tunnel may have been initiated by an HTTP request.
  - The tunnel ceases to exist when both ends of the relayed connections are closed.

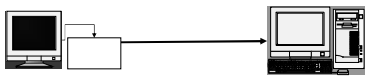


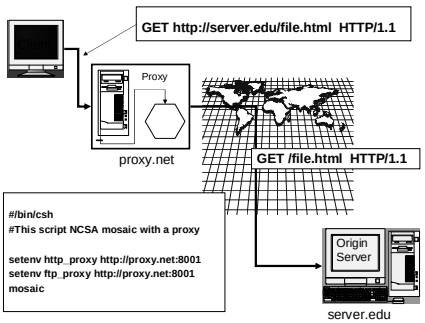
WEB CACHING

More Definitions

- Cache
  - A program's local store of response messages and the subsystem that controls its message storage, retrieval, and deletion. A cache stores cachable responses in order to reduce the response time and network bandwidth consumption on future, equivalent requests. Any client or server may include a cache, except the tunnel.

Click Here to see Netscape's Cache





A Proxy Server Cache

Why Caching?

- Reduced User Experienced Latency
  - Faster response means user spends more time on a site.
- Reduced Load on the Network
- Reduced Load on the Origin Server
  - Fewer documents to serve.
  - Fewer connections to maintain.

Components of Delay

- User Connectivity to ISP
- DNS Lookup time
- Bandwidth between User and Origin Server
- Congestion between User and Origin Server
- Load on the Origin Server
- Time to generate response
- Time to render response by the Browser

INTERNET ENGINEERING

User Advantages

- Predictability in Response/Service Time
  - The connectivity between user and ISP is more predictable than in the Internet.
- DNS Lookup Delay
  - May increase or decrease.
- Network Congestion
  - Delay decreases for static documents.
- Origin Server Load
  - Reduced.
  - Less persistent connections.
- Time to Generate Response
  - No change
- Browser Rendering
  - No change.

INTERNET ENGINEERING

What is Cacheable?

- Protocol Specific Considerations
  - Responses to "OPTIONS", "PUT", and "DELETE" methods are not cached.
  - Directive "No-store" prevents caching.
  - Directive "No-cache" forces revalidation.
  - Short "Expires".
  - Presence of "Authorization" & "Vary".
- Content Specific Considerations
  - A cacheable content is not always cached. A cache generally has its own set of additional rules.
  - Things that are prone to change:
    - Dynamically generated, cookies, scripted responses.
  - Things which may not change:
    - Electronic book
  - Things which are draining:
    - Large and less frequently requested.

INTERNET ENGINEERING

Internet Caching & HTTP

More Definitions

- First-hand
  - A response is first-hand if it comes directly and without unnecessary delay from the origin server, perhaps via one or more proxies. A response is also first-hand if its validity has just been checked directly with the origin server.
- Explicit expiration time
  - The time at which the origin server intends that an entity should no longer be returned by a cache without further validation.
- Heuristic expiration time
  - An expiration time assigned by a cache when no explicit expiration time is available.

INTERNET ENGINEERING

More Definitions

- Age
  - The age of a response is the time since it was sent by, or successfully validated with, the origin server.
- Freshness lifetime
  - The length of time between the generation of a response and its expiration time.
- Fresh
  - A response is fresh if its age has not yet exceeded its freshness lifetime.
- Stale
  - A response is stale if its age has passed its freshness lifetime. semantically transparent

INTERNET ENGINEERING

More Definitions

- Semantically transparent
  - A cache behaves in a "semantically transparent" manner, with respect to a particular response, when its use affects neither the requesting client nor the origin server, except to improve performance. When a cache is semantically transparent, the client receives exactly the same response (except for hop-by-hop headers) that it would have received had its request been handled directly by the origin server.
- Validator
  - A protocol element (e.g., an entity tag or a Last-Modified time) that is used to find out whether a cache entry is an equivalent copy of an entity.

INTERNET ENGINEERING

Caching in HTTP

- The objective of HTTP caching is to improve the performance by reducing network traffic by caching responses.
- Specific Goals:
  - Reduces/eliminates send/request entire cycles.
  - Reduce/eliminate sending full responses.
- Control Models:
  - "expiration" model
  - "validation" model

INTERNET ENGINEERING

Control Mechanisms

- Default Cache Control Algorithm:
  - A set of algorithms which work with server specified:
    - Expiration times
    - Validators
  - Algorithms remove ambiguity.
- Explicit Cache Control Directives:
  - A server or client can send explicit directives to HTTP caches. There directives overrides default algorithms.
  - If, there is conflict between directives, the most restrictive one supercedes.

INTERNET ENGINEERING

•Default
•Expiration
•Validation
•Directives

Expiration Models

- Server-Specified Expiration
  - Origin Server specifies an explicit expiration date.
  - A cache can return a fresh response without contacting the server by looking at the expiration date.
  - It completely eliminates cache to server roundtrip communication.
  - Server can specify an expiration time by:
    - Expire Header
    - `Expires: Thu, 01 Dec 1998 16:00:00 GMT`
    - max-age in Cache-Control Header
    - `Cache Control: max-age=3600`

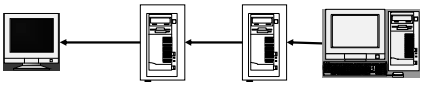
INTERNET ENGINEERING

•A cache can use a heuristic model to determine expiration date.
•Example 20% (Last modified time-creation date)

•Default
•Expiration
•Validation
•Directives

General Idea of using Age Value

- HTTP/1.1 uses the Age response-header to help convey age information between caches.
- The Age header value is the sender's estimate of the amount of time since the response was generated at the origin server.
- In the case of a cached response that has been revalidated with the origin server, the Age value is based on the time of revalidation, not of the original response.
- Age value is the sum of the time that the response has been resident in each of the caches along the path from the origin server, plus the amount of time spent in transit.



INTERNET ENGINEERING

HTTP/1.1 200 OK
Server: NCSA/1.3
Mime_version:1.0
Age: 3600
Content_type: text/html
Content_length:2000
<HTML>
…
</HTML>

Two Methods

- A response's age can be calculated in two entirely independent ways:
  - 1. now minus date_value, if the local clock is reasonably well synchronized to the origin server's clock. If the result is negative, the result is replaced by zero.

INTERNET ENGINEERING

Computing Freshness

RESPONSE_IS_FRESH = if
(FRESHNESS_LIFE_TIME>CURRENT_AGE)

If none works, cache MAY use heuristics. If FLT if greater than 24 hours it must issue warning!

FRESHNESS_LIFE_TIME
=MAX_AGE_VALUE if not there,
=EXPIRES_VALUE-DATE_VALUE

Cache Control: max-age=3600

Expires: Thu, 01 Dec 1998 16:00:00 GMT
Date: Tue, 15 Nov 1998 08:12:31 GMT

Date Header is the date at which the message was originated.

If the origin server or user agent do not put date, then the receiving cache appends the date at the receiving end.

INTERNET ENGINEERING

Computing Current Age

RESPONSE_IS_FRESH = if
(FRESHNESS_LIFE_TIME>CURRENT_AGE)

APPARENT_AGE= max(RESPONSE_TIME-DATE_VALUE, 0)
CORRECTED_RECEIVED_AGE=max(APPARENT_AGE, AGE_VALUE)
RESPONSE_DELAY=(RESPONSE_TIME-REQUEST_TIME)
CORRECTED_LOCAL_AGE=RESPONSE_DELAY+CORRECTED_RECEIVED_AGE
RESIDENT_TIME=NOW-RESPONSE_TIME
CURRENT_AGE=CORRECTED_LOCAL_AGE+RESIDENT_TIME
or
=MAX(0,RESPONSE_TIME-DATE_VALUE, AGE_VALUE)+NOW-REQUEST_TIME

Age:3600

Expires: Thu, 01 Dec 1998 16:00:00 GMT
Date: Tue, 15 Nov 1998 08:12:31 GMT

Age Header is the senders estimate of the amount of time since origin server generated the response.

HTTP/1.1 cache must send an Age Header in every response.

INTERNET ENGINEERING

Validation Models

- When a cache has a stale entry it first has to check with the origin server (or possibly an intermediate cache with a fresh response) to see if its cached entry is still usable.
- We do not want to pay the overhead of:
  - retransmitting the full response, if the cached entry is good
  - or an extra round trip, if the cached entry is invalid
  - How can we do that?
- The HTTP/1.1 protocol supports the use of conditional methods with cache validators.

•Default
•Expiration
•Validation
•Directives

INTERNET ENGINEERING

Validation Model (continued..)

- When an origin server generates a full response, it attaches a validator. It is kept with the cache entry.
- When a client (user agent or proxy cache) makes a conditional request for a resource for which it has a cache entry, it includes the associated validator in the request.
- The server then checks that validator against the current validator for the entity.
- On match, it responds with a special status code: usually, 304 Not Modified, and no entity-body.
- Otherwise, it returns a full response (+entity-body). It
  - avoids transmitting the full response if the validator matches.
  - an extra round trip if it does not match.

Can a response which lacks validator be cached? Can it be validated, when expired?

INTERNET ENGINEERING

Example: Last Modified date based Validation

- Last-modified Dates
  - The Last-Modified entity-header field value is often used as a cache validator. In simple terms, a cache entry is considered to be valid if the entity has not been modified since the Last-Modified value.

HTTP/1.1 200 OK
Server: NCSA/1.3
Mime_version:1.0
Age: 3600
Last Modified: Thu, 01 Dec 1999 16:00:00 GMT
Date: Tue, 15 Nov 1998 08:12:31 GMT
Content_type: text/html
Content_length:2000
<HTML>
…
</HTML>

What is wrong here?

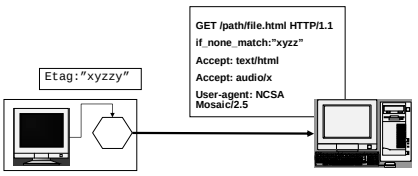
Server makes them equal.

Strong vs. Weak Validators

INTERNET ENGINEERING

Example: Conditional Validation

- Entity Tag Cache Validators
  - The ETag entity-header field value, an entity tag, provides for an"opaque" cache validator. This may allow more reliable validation in situations where it is inconvenient to store modification dates, where the one-second resolution of HTTP date values is not sufficient, or where the origin server wishes to avoid certain paradoxes that may arise from the use of modification dates.



INTERNET ENGINEERING

Explicit Cache Control Directives

- The Cache-Control general-header field is used to specify directives that MUST be obeyed by all caching mechanisms along the request/response chain.
- These directives typically override the default caching algorithms.
- Cache directives are unidirectional in that the presence of a directive in a request does not imply that the same directive should be given in the response.

- Major Classes
  - Restriction, what is cacheable (by OS)
  - Restriction, what may be stored (by OS)
  - Modification, based on Expiration Model (by OS+UA)
  - Control over Cache Revalidation (by UA)
  - Control over Entity Transformation

INTERNET ENGINEERING

•Default
•Expiration
•Validation
•Directives

Explicit Cache Control Directives

Cache-Control: cache-directive

Request Directives:

"no-cache" ["=" <"> 1#field-name <">]
| "no-store"
| "max-age" "=" delta-seconds
| "max-stale" "=" delta-seconds
| "min-fresh" "=" delta-seconds
| "only-if-cached"
| cache-extension

INTERNET ENGINEERING

Explicit Cache Control Directives

Cache-Control: cache-directive

Response Directives:

"public"
| "private" ["=" <"> 1#field-name <">]
| "no-cache" ["=" <"> 1#field-name <">]
| "no-store"
| "no-transform"
| "must-revalidate"
| "proxy-revalidate"
| "max-age" "=" delta-seconds
| cache-extension

INTERNET ENGINEERING