

# CACHE MANAGEMENT

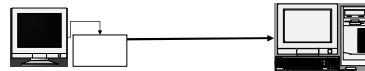
24

## More Definitions

- Cache

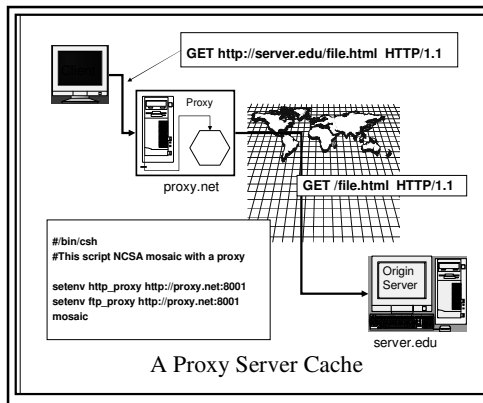
- A program's local store of response messages and the subsystem that controls its message storage, retrieval, and deletion. A cache stores cachable responses in order to reduce the response time and network bandwidth consumption on future, equivalent requests. Any client or server may include a cache,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

*[Click Here to see Netscape's Cache](#)*



INTERNET  
ENGINEERING

LECT-11, S-25  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004



A Proxy Server Cache



INTERNET  
ENGINEERING

LECT-11, S-26  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

## 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

LECT-11, S-27  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

## 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

LECT-11, S-28  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

## 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

LECT-11, S-29  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

### 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

LECT-11, S-30  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

### 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. These directives overrides default algorithms.
  - If, there is conflict between directives, the most restrictive one supercedes.




INTERNET  
ENGINEERING

LECT-11, S-31  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

•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


LECT-11, S-32  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

•Default  
 •Expiration  
 •Validation  
 •Directives

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

### 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.

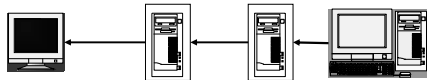


INTERNET  
ENGINEERING

LECT-11, S-33  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004


```

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:
  - 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.
  - age\_value, if all of the caches along the response path implement HTTP/1.1.
- Given that we have two independent ways to compute the age of a response when it is received, we can combine these as:
 
$$\text{CORRECTED\_RECVD\_AGE} = \max(\text{NOW} - \text{DATE\_VALUE}, \text{AGE\_VALUE})$$



INTERNET  
ENGINEERING

LECT-11, S-34  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

### 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!

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.

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



INTERNET  
ENGINEERING

LECT-11, S-35  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

### Computing Current Age

$$\text{RESPONSE\_IS\_FRESH} = \text{if} \\ (\text{FRESHNESS\_LIFE\_TIME} > \text{CURRENT\_AGE})$$

$$\text{APPARENT\_AGE} = \max(\text{RESPONSE\_TIME-DATE\_VALUE}, 0)$$

$$\text{CORRECTED\_RECEIVED\_AGE} = \max(\text{APPARENT\_AGE}, \text{AGE\_VALUE})$$

$$\text{RESPONSE\_DELAY} = (\text{RESPONSE\_TIME} - \text{REQUEST\_TIME})$$

$$\text{CORRECTED\_LOCAL\_AGE} = \text{RESPONSE\_DELAY} + \text{CORRECTED\_RECEIVED\_AGE}$$

$$\text{RESIDENT\_TIME} = \text{NOW} - \text{RESPONSE\_TIME}$$

$$\text{CURRENT\_AGE} = \text{CORRECTED\_LOCAL\_AGE} + \text{RESIDENT\_TIME}$$

or

$$= \max(0, \text{RESPONSE\_TIME-DATE\_VALUE}, \text{AGE\_VALUE}) + \text{NOW} - \text{REQUEST\_TIME}$$

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.

LECT-11, S-36  
IN2004S, javed@kent.edu  
 Javed I. Khan@2004

### 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

LECT-11, S-37  
IN2004S, javed@kent.edu  
 Javed I. Khan@2004

### 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?

LECT-11, S-38  
IN2004S, javed@kent.edu  
 Javed I. Khan@2004

### 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 1998 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**

LECT-11, S-39  
IN2004S, javed@kent.edu  
 Javed I. Khan@2004

### 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.

```

GET /path/file.html HTTP/1.1
If-none-match: "xyzz"
Accept: text/html
Accept: audio/x
User-agent: NCSA
Mosaic/2.5

```

ETag: "xyzz"

LECT-11, S-40  
IN2004S, javed@kent.edu  
 Javed I. Khan@2004

### 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

•Default  
 •Expiration  
 •Validation  
 •Directives

LECT-11, S-41  
IN2004S, javed@kent.edu  
 Javed I. Khan@2004

### 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
```

LECT-11, S-42  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

### 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
```

LECT-11, S-43  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

```
#bin/csh
#This script NCSA mosaic with a proxy
setenv http_proxy http://proxy.net:8001
setenv ftp_proxy http://proxy.net:8001
mosaic
```

A Proxy Server Cache

LECT-11, S-44  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

### Cache Performance

Hit Ratio

$$T_{ave} = h.T_{cache} + (1-h).T_{origin}$$

LECT-11, S-45  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

### Cache Location

|           | Client Cache | Proxy Cache | Server Cache |
|-----------|--------------|-------------|--------------|
|           |              | outbound    | inbound      |
| # clients | one          | few         | most         |
| # servers | all          | most        | few          |
| # docs    | all          | most        | few          |

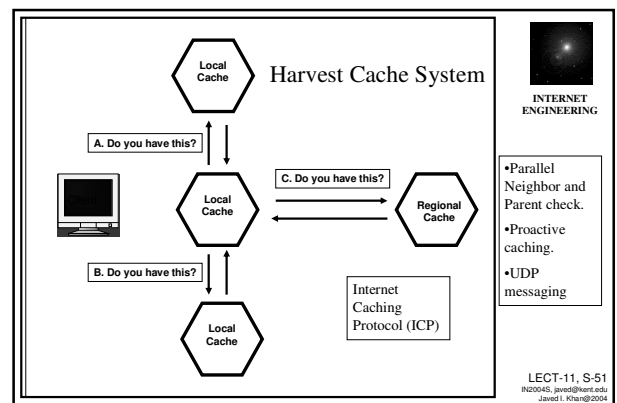
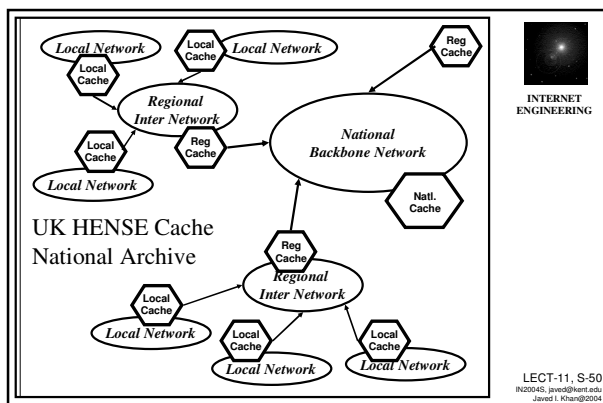
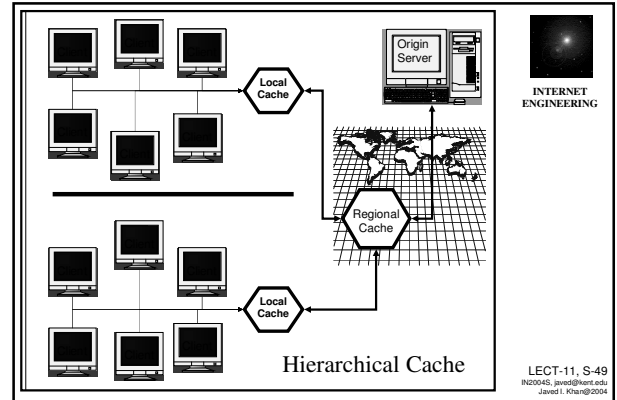
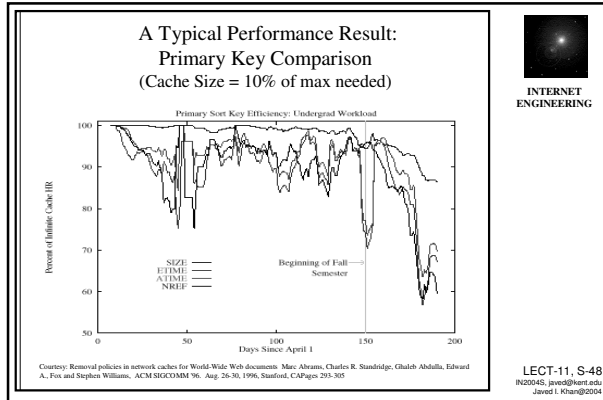
This talk

LECT-11, S-46  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004

### Cache Replacement Policies

- It is a delicate balance between the cache and performance. Following are some attributes which can be used for cache document replacement:
  - SIZE document size (bytes)
  - ETIME time doc entered cache
  - NREF number references to doc
  - ATIME time doc last accessed

LECT-11, S-47  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004



### Reference Material

- Primary source RFC-2068.
  - Specially read section:
    - Section 13
    - Section 14.9
- Books:
  - Web Protocols & Practices, Addison Wesley, Krishnamurty & Rexford, 2001.
    - Sections 6 & 7.

LECT-11, S-52  
IN2004S\_javed@kent.edu  
Javed I. Khan@2004