Web Engineering

Prof. Dr. Dr. h.c. mult. Gerhard Krüger, Albrecht Schmidt

Universität Karlsruhe
Fakultät für Informatik
Institut für Telematik

Wintersemester 1999/2000

Web Engineering

Chapter 2: Foundation - Identifiers and Protocols (cont.)

Structure of HTTP Messages

```
HTTP-message = Request | Response

generic-message = start-line
  *message-header
  CRLF [ message-body ]

start-line = Request-Line | Status-Line

message-header = field-name ":" [ field-value ] CRLF
```


General Header Fields

```
general-header = Cache-Control
 | Connection
 | Date
 | Pragma
 | Transfer-Encoding
 | Upgrade
 | Via
```

Request

```
Request = Request-Line
   *(general-header | request-header | entity-header ) CRLF
  [ message-body ]

Request-Line   = Method SP Request-URI SP HTTP-Version CRLF

Method = "OPTIONS"
| "GET"
| "HEAD"
| "POST"
| "PUT"
| "DELETE"
| "TRACE"
| extension-method
```


Request Header Fields

```
request-header = Accept
 | Accept-Charset | Accept-Encoding  | Accept-Language
 | Authorization
 | From
 | Host
 | If-Modified-Since  | If-Match  | If-None-Match
 | If-Range  | If-Unmodified-Since
 | Max-Forwards
 | Proxy-Authorization
 | Range
 | Referer
 | User-Agent
```


Response

```
Response = Status-Line
   *(general-header | response-header | entity-header ) CRLF
  [ message-body ]

Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF
```

- 1xx: Informational - Request received, continuing process
- 2xx: Success - The action was successfully received, understood,and accepted
- 3xx: Redirection - Further action must be taken in order to complete the request
- 4xx: Client Error - The request contains bad syntax or cannot be fulfilled
- 5xx: Server Error - The server failed to fulfill an apparently valid request

Response Status Codes (selection)

```
Status-Code = "100" ; Continue
   | "101" ; Switching Protocols
   | "200" ; OK
   | "201" ; Created
   | "202" ; Accepted
   | "203" ; Non-Authoritative Information
   | "204" ; No Content
   | "300" ; Multiple Choices
   | "301" ; Moved Permanently
   | "400" ; Bad Request
   | "401" ; Unauthorized
   | "402" ; Payment Required
   | "403" ; Forbidden
   | "404" ; Not Found
   | "405" ; Method Not Allowed
   | "500" ; Internal Server Error
   | "501" ; Not Implemented
```

Developers can define response codes. They must fit within one of the classes and they are treated (if unknown) as X00

Response Header Felder

```
response-header = Age
   | Location
   | Proxy-Authenticate
   | Public
   | Retry-After
   | Server
   | Vary
   | Warning
   | WWW-Authenticate
```


Entity Header Fields

```
entity-header  = Allow
   | Content-Base
   | Content-Encoding
   | Content-Language
   | Content-Length
   | Content-Location
   | Content-MD5
   | Content-Range
   | Content-Type
   | Etag
   | Expires
   | Last-Modified
   | extension-header
```

Content-MD5

Is not a security feature, it is only a mechanism to track accidental change of the message body – because if the body can be altered also the Content-MD5 could be changed.

HTTP/1.1, Request Methods I

- GET
  - to requests a resource
- HEAD
  - to request the header of a resource
- POST
  - to submit a resource (e.g. parameters)
- DELETE, PUT
  - file manipulation functions, similar to FTP
- OPTIONS
  - to request information about the server and intermediary programs
- TRACE
  - to request information about the route

HTTP/1.1, Request Methods II

- safe methods =
  calls of these methods don‘t change data on the server = these methods do not have any side-effects
  - HEAD
  - GET (should be implemented this way!)
- idempotente Methods =
  a single call of these method has the same effect as multiple calls to of these methods =
  side-effects of N > 0 identical requests is the same as for a single request.
  - HEAD
  - PUT
  - DELETE
  - GET (should be implemented this way!)

HTTP/1.1, GET Request

- Syntax: **GET <URI> <VERSION>**
- Request resource named by the URI
  - static resource
  - dynamic resource
  - providing parameters
- conditional GET
  - get the resource only under a certain condition
  - may reduce network traffic
- partial GET
  - Get only a certain part of the resource
  - may reduce network traffic

HTTP/1.1, GET Example

```
GET /index.html HTTP/1.1
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE5.0; Windows NT)
Host: www.teco.edu
Connection: Keep-Alive
```

```
HTTP/1.1 200 OK
Date: Wed, 27 Oct 1999 14:13:43 GMT
Server: Apache/1.2.1
Keep-Alive: timeout=10, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html

f78
<html><head>
<title>Telecooperation Office (TecO)</title>
...
```


Conditional GET

- Syntax:
  GET <URI> <VERSION>
  <CONDITIONAL-HEADER>: <DATE>

  z.B. **If-Modified-Since, If-Match, If-Range, etc.**

- Example:

```
GET http://www.apache.org/index.html HTTP/1.1
Host: www.apache.org
If-Modified-Since: Fri, 29 Oct 1999 13:53:40 GMT
```

```
HTTP/1.0 304 Not Modified
Date: Thu, 28 Oct 1999 13:55:13 GMT
Content-Type: text/html
Expires: Fri, 29 Oct 1999 13:55:13 GMT
```


Partial GET

- Syntax:
  GET <URI> <VERSION>
  Range: <RANGE>

  z.B. `bytes=0-499` get the first 500 bytes of the resource
  `bytes=-100` get the last 100 bytes of the resource
  `bytes=1500-` get the resource starting from byte 1500

- Beispiel:

```
GET/Default.htm HTTP/1.1
Host:www.microsoft.com
Range:bytes=0-80
```

```
HTTP/1.1 206 Partial content
Server:Microsoft-IIS/4.0
Date: Thu, 28 Oct 1999 14:21:00 GMT
Content-Type:text/html
Content-Length:81
Content-Range: bytes 0 -80/19618

<HTML><HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso88
```

HTTP/1.1, HEAD Request

- Syntax: **HEAD <URI> <VERSION>**
- request the header information of a resource name by the URI
- serves the same header information as a GET request of the same resource
- request of meta information
- used for
  - checking links
  - checking whether or not it is required to transmit the resource (e.g. compare time stamp or size)
  - may reduce network traffic

HTTP/1.1, HEAD Example I

```
HEAD / HTTP/1.0
Host: www.sun.com
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE5.0; Windows NT)
```

```
HTTP/1.1 200 OK
Server: Netscape-Enterprise/3.6
Date: Fri, 03 Nov 2000 06:10:24 GMT
Set-cookie: sessionid=TVOS0UYAAC00XAMTA1LU5YQ;path=/
Content-type: text/html
Content-length: 13657
Connection: close
```


HTTP/1.1, HEAD Example II

```
GET / HTTP/1.0
Host: www.sun.com
Accept: */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE5.0; Windows NT)
```

```
HTTP/1.1 200 OK
Server: Netscape-Enterprise/3.6
Date: Fri, 03 Nov 2000 06:10:24 GMT
Set-cookie: sessionid=TVOS0UYAAC00XAMTA1LU5YQ;path=/
Content-type: text/html
Content-length: 13657
Connection: close
```

```
<!DOCTYPE HTML ....
<HTML...
...
```

GET serves the same headers as before HEAD

HTTP/1.1, POST Request

- Syntax:
  POST <URI> <VERSION>
  <HEADER>
  <CRLF>
  <MESSAGE-BODY>
- Supply data to a resource named in the URI (see chapter on programming)
- return codes:
  - 200 OK
  - 204 No Content
  - 201 Created (location header)

HTTP/1.1, POST Example

```
POST /test.cgi HTTP/1.1
Accept:, */*
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows
  NT; DigExt)
Host: www.teco.edu:8080
Content-Length: 55
Connection: Keep-Alive

name=Maier&path=%2F&ort=Karlsruhe&submit=Submit+Request
```

```
HTTP/1.1 200 OK
Date: Wed, 27 Oct 1999 14:13:43 GMT
Server: Apache/1.2.1
Keep-Alive: timeout=10, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html

c18
<html><head>
<title>CGI-Script</title>
...
```


HTTP/1.1, OPTIONS Request

- Syntax:
  OPTIONS <URI> <VERSION>
- Request the communication options for a resource named by the URI or the communication options for a server by setting **<URI>=***
- Header fields in the response
  - **Accept**
  - **Accept-Range**
  - **Accept-***
  - **Public**
  - **Allow** (entity)

HTTP/1.1, OPTIONS Example I

```
OPTIONS * HTTP/1.1
Host: www.apache.org
```

```
HTTP/1.1 200 OK
Date: Wed, 27 Oct 1999 15:02:18 GMT
Server: Apache/1.3.10 (Unix) ApacheJServ/1.0 PHP/3.0.6
Cache-Control: max-age=86400
Expires: Thu, 28 Oct 1999 15:02:18 GMT
Content-Length: 0
Allow: GET, HEAD, OPTIONS, TRACE
```


HTTP/1.1, OPTIONS Example II

```
OPTIONS * HTTP/1.1
Host: www.microsoft.com
```

```
HTTP/1.0 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 27 Oct 1999 15:04:17 GMT
Content-Length: 0
Accept-Ranges: bytes
DASL:
DAV: 1, 2
Public: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST,
  COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK,
  SEARCH
Allow: OPTIONS, TRACE, GET, HEAD, DELETE, PUT, POST,
  COPY, MOVE, MKCOL, PROPFIND, PROPPATCH, LOCK, UNLOCK,
  SEARCH
Cache-Control: private
```

HTTP/1.1, TRACE Request

- Syntax:
  TRACE <URI> <VERSION>
- restricted by **<Max-Forwards>**
- method for diagnosis of communication
- shows the route of a message
  header field: **VIA**
- the message sent is included in the body

HTTP/1.1, TRACE – Max-Forwards I

```
TRACE /index.htm HTTP/1.1
Host: www.teco.edu
Max-Forwards: 0
```

```
HTTP/1.0 200 OK
Server: Squid/2.2.STABLE3
Mime-Version: 1.0
Date: Thu, 28 Oct 1999 18:02:04 GMT
Content-Type: text/plain
Content-Length: 66
Expires: Thu, 28 Oct 1999 18:02:04 GMT
X-Cache: MISS from www.teco.uni-karlsruhe.de
Proxy-Connection: close

TRACE /index.htm HTTP/1.1
Host: www.teco.edu
Max-Forwards: 0
```


HTTP/1.1, TRACE – Max-Forwards II

```
TRACE /index.htm HTTP/1.1
Host: www.teco.edu
Max-Forwards: 2
```

```
HTTP/1.0 200 OK
Date: Thu, 28 Oct 1999 18:00:13 GMT
Server: Apache/1.2.1
Content-Type: message/http
Age: 0
X-Cache: MISS from www.teco.uni-karlsruhe.de
Proxy-Connection: close

TRACE /index.htm HTTP/1.1
Host: www.teco.edu
Max-Forwards: 1
Via: 1.1 www.teco.uni-karlsruhe.de:3128 (Squid/2.2.STABLE3)
X-Forwarded-For: 129.13.170.1
Cache-Control: max-age=259200
Connection: keep-alive
```


HTTP/1.1, TRACE Example I without a Proxy

```
TRACE / HTTP/1.1
Host: www.microsoft.com
```

```
HTTP/1.0 200 OK
Server: Microsoft-IIS/4.0
Date: Wed, 27 Oct 1999 15:09:29 GMT
Content-Type: message/http
Content-Length: 42

TRACE / HTTP/1.1
Host: www.microsoft.com
```

HTTP/1.1, TRACE Example II with Proxy

```
TRACE http://www.microsoft.com/ HTTP/1.1
Host: www.microsoft.com
```

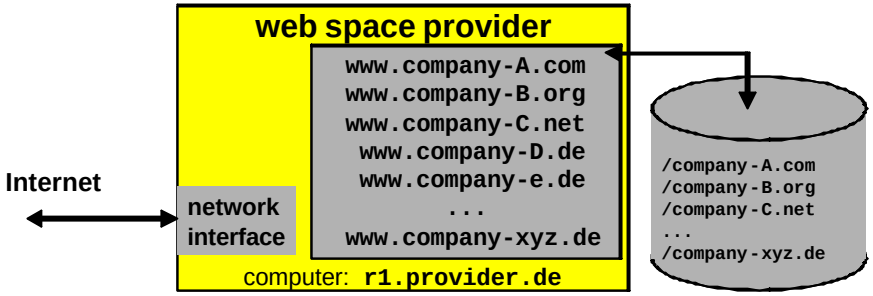
```
HTTP/1.0 200 OK
Server: Microsoft-IIS/5.0
Date: Wed, 27 Oct 1999 15:11:58 GMT
Content-Type: message/http
Content-Length: 192
Age: 0
X-Cache: MISS from www.teco.uni-karlsruhe.de
Proxy-Connection: keep-alive

TRACE / HTTP/1.0
Host: www.microsoft.com
Via: 1.1 www.teco.uni-karlsruhe.de:3128 (Squid/2.2.S3)
X-Forwarded-For: 129.13.170.1
Cache-Control: max-age=259200
Connection: keep-alive
```


Solutions - HTTP/1.1

- support for non-IP-based virtual Hosts
  - several Web Server can be hosted on a machine with a single IP-Address
- enhanced and extended caching model
  - support for proxies, tunnels and gateways
- more than one HTTP-requests per connection
  - optimize for TCP properties (e.g. TCP slow-start)
- partial transfer of resources
  - partial re-transmission
- extended authentication
  - password can be encrypted

Virtual Hosts

- idea: to host the web servers for different customers and/or domains on a machine with only one network interface
- the software (one or more web server) should handle the requests for all customers/domains on this machine
- applications scenario: web-hosting, provider



Virtual Hosts - Solutions

- different ports on the Server (from HTTP/0.9)
  - each web server/customer/domain gets a different port (e.g. 80, 1080, 2080, 8080, ...)
  - **Problem:** only one can use the default port, all others have to explicitly state their port in the URI
- IP-based virtual Hosts (from HTTP/0.9)
  - assign several IP-addresses to one computer
  - Each server/customer/domain gets its own IP address, DNS resolves name to IP address
  - **Problems:**
    - most systems restrict the number of IP-addresses per machine
    - wasting IP-Addresses (there is a shortage of IPv4 addresses)
- Non-IP-based virtual Hosts (from HTTP/1.1)
  - A web server (using one IP-address and one port) determines from HTTP-protocol level the domain for the requested resource.
  - All Domains on this machine have the same IP-Address

Non-IP-based Virtual Hosts

- HTTP-Requests uses a additional header field
  `Host: <hostname>`
  and/or optional the complete and absolute URI

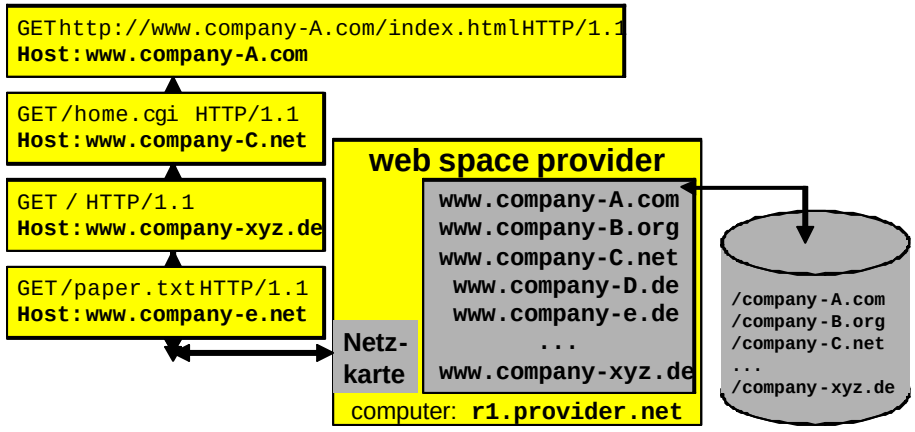
- HTTP/1.1 request without host name result in an error, e.g. the following error log entry
  `[access to /index.html failed for 129.13.170.1,`
  `reason: client sent HTTP/1.1 request without hostname]`

- all domains on one server have the same IP address

- The domains are discriminated on HTTP-protocol level and not by the TCP connection

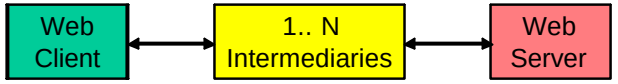
Non-IP-based Virtual Hosts - Example

- IP-addr(www.company-A.com) = 129.13.170.1
- IP-addr(www.company-B.org) = 129.13.170.1
- IP-addr(www.company-C.net) = 129.13.170.1
- IP-addr(www.company-D.de) = 129.13.170.1
- IP-addr(www.company-e.de) = 129.13.170.1
- ...
- IP-addr(www.company-xyz.de) = 129.13.170.1
- IP-addr(r1.provider.net) = 129.13.170.1



HTTP/1.1 – Intermediaries

- 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.
- 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.
- 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.



Caching / Proxies

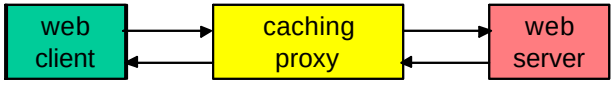
- cache
  A program's local store of response messages and the subsystem t hat controls its message storage, retrieval, and deletion. A cache s tores cachable responses in order to reduce the response time and network bandwidth consumption on future, equivalent requests. Any client or server may include a cache, though a cache cannot be used by a s erver that is acting as a tunnel.
- support of caching by
  - Header Fields, e.g. :
    `Cache-Control`, **`Expires`**, **`Age`**, **`Pragma`**
  - For more detail see chapter 4
- support for non-IP-based virtual hosts
  - client can identify the requested web server
  - header field **`Host`**

Cache-Control Header

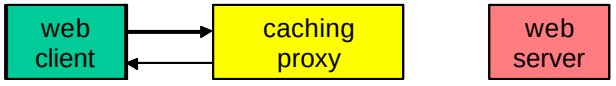
```
Cache-Control   = "Cache-Control" ":" 1#cache-directive

cache-directive = cache-request-directive
             | cache-response-directive

cache-request-directive = "no-cache" | "no-store„
   | "max-age" "=" delta-seconds
   | "max-stale" [ "=" delta-seconds ]
   | "min-fresh" "=" delta-seconds
   | "no-transform" | "only-if-cached" | cache-extension

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


Caching-Proxy - Example

- Cache - MISS
  - The requested resource is not stored in the cache
  - The resource is requested from the original server



- Cache - HIT
  - The requested resource is stored in the cache of the proxy and is still valid
  - The resource sent back directly from the caching proxy, it is not requested from the original host



Cache-MISS – Request Using a Proxy (www.teco.edu:3128)

```
GET http://www.ulm.de/info_ul/ HTTP/1.1
Host: www.ulm.de
```

```
HTTP/1.0 200 OK
Date: Tue, 26 Oct 1999 10:16:37 GMT
Server: Apache/1.2.6 Red Hat
Last-Modified: Sat, 23 Jan 1999 15:23:35 GMT
ETag: "d908a-65c-36a9e977"
Content-Length: 1628
Accept-Ranges: bytes
Content-Type: text/html
Age: 0
X-Cache: MISS from www.teco.uni-karlsruhe.de
Proxy-Connection: keep-alive
...
```

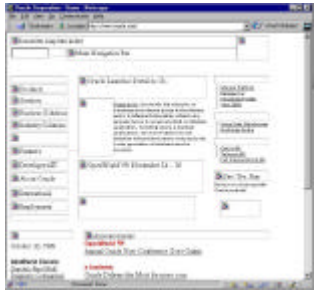

Cache-HIT - Request Using a Proxy (www.teco.edu:3128)

```
GET http://www.ulm.de/info_ul/ HTTP/1.1
Host: www.ulm.de
```

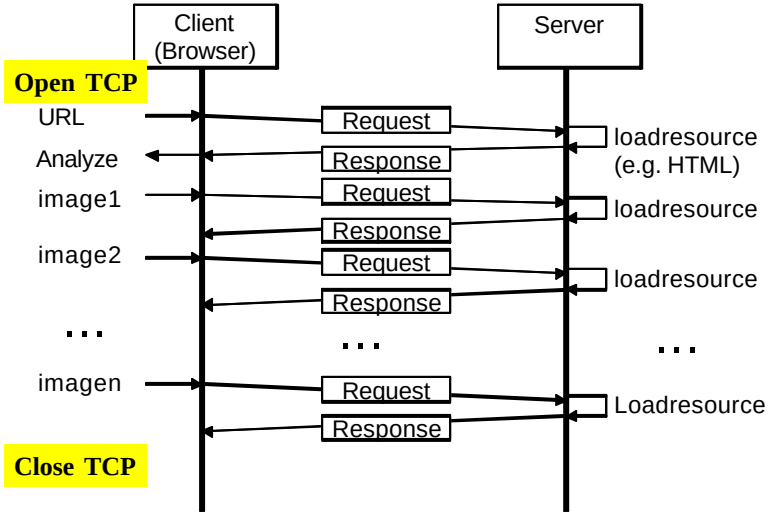
```
HTTP/1.0 200 OK
Date: Tue, 26 Oct 1999 10:16:37 GMT
Server: Apache/1.2.6 Red Hat
Last-Modified: Sat, 23 Jan 1999 15:23:35 GMT
ETag: "d908a-65c-36a9e977"
Content-Length: 1628
Accept-Ranges: bytes
Content-Type: text/html
Age: 56
X-Cache: HIT from www.teco.uni-karlsruhe.de
Proxy-Connection: keep-alive
...
```

HTTP/1.1, Persistent Connection

- header field **`Connection`**
  value: **`Keep-Alive, Close`**
- Requires less system resources
- less network traffic (less packets)
- pipelining of requests is possible
  order of documents is guaranteed by the server



Persistent Connection – only one TCP-Connection



persistent Connections – Pipelining I

```
GET / HTTP/1.1
host:www.teco.edu
connection:keep-alive

GET/images/tecologo.gifHTTP/1.1
host:www.teco.edu
connection:keep-alive

GET / teco/cgi-bin/tifre/people.plHTTP/1.1
host:www.teco.edu
connection:keep-alive
```

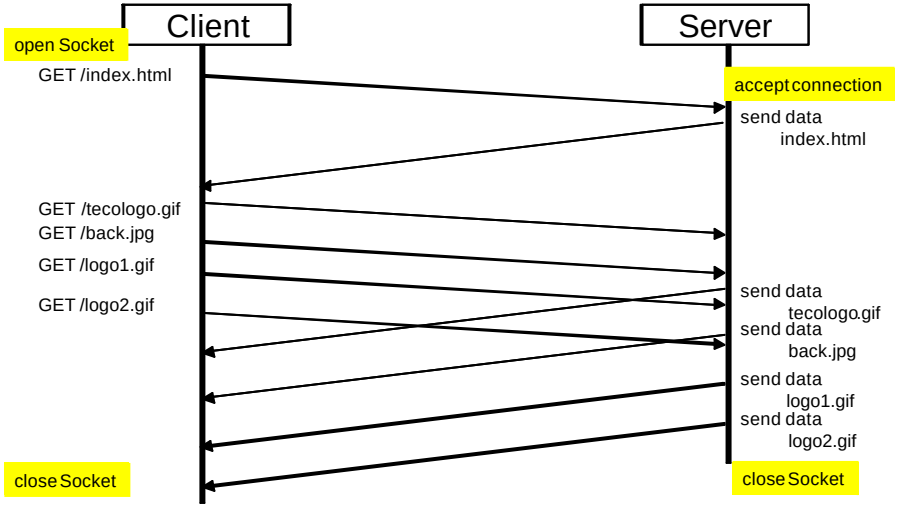
Datei: index.html

Datei: tecologo.gif

Datei: people.pl

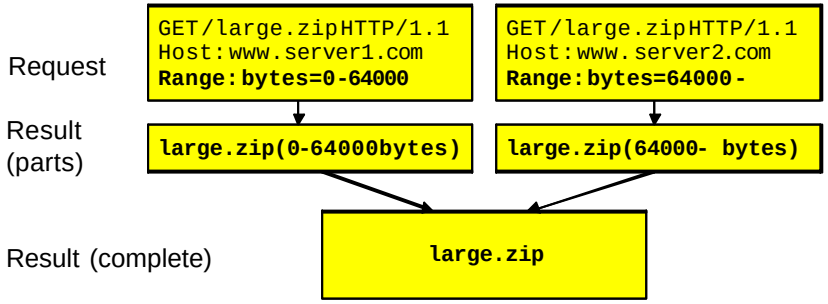
```
HTTP/1.1 200 OK
HTTP/1.1 200 OK
HTTP/1.1 200 OK
Date: Tue, 26 Oct 1999 16:56:37 GMT
Server: Apache/1.2.1
Keep-Alive: timeout=10, max=98
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html

cd2
<html><head>
<title>TecO - People</title>
...
```


persistent Connections – Pipelining II



Partieller Transfer von Ressourcen

- particial GET
- specified using the header field
  `Range: <RANGE>`
- transfer of large files in a number of smaller parts
- re-transmission of interrupted transfers
  - only the missing part must be transmitted
- transfer of a large resource concurrently from multiple servers
- example:



HTTP/1.1, Authentication

- improved authentication
  - Digest Access Authentication
  - Basic Authentication still possible and supported
- procedure
  - response with status-code: **`401 Unauthorized`** and header
    `WWW-Authenticate:Digestrealm="...",nonce="..."`
  - user is asked to provide password
  - request to the resource with additional header fields
    `Authorization: Digest realm="...", username="...", response="..."`
    `response = <MD5(<passwd>:<nonce>:...)>`
  - **`<passwd>:<nonce>`** is MD5 coded, reverse is not possible
  - server calculated also MD5 of **`<passwd>:<nonce>:...`**
  - if the response matches the servers result the document is transferred
- password is not sent!
- problem: initial exchange of passwords