

Network Services, VU 2.0

HTTP, FTP

Dipl.-Ing. Johann Oberleiter
Institute for Information Systems, Distributed
Systems Group

Agenda

- URIs
- HTTP (Hyper Text Transfer Protocol)
- WebDAV
- FTP

URI

- Unique Resource Identifier
 - Remembered by people
 - Transcribed from one network resource to another -> characters accessible on each keyboard
- RFC 3896
- URI =
scheme:hierarchical-part
[?query] [#fragment]
 - Hierarchical-part absolute or relative
 - Hierarchical-part may contain authority part

URI Examples

- <ftp://ftp.is.co.za/rfc/rfc1808.txt>
- <http://www.ietf.org/rfc/rfc2396.txt>
- [ldap://\[2001:db8::7\]/c=GB?objectClass=one](ldap://[2001:db8::7]/c=GB?objectClass=one)
- <mailto:John.Doe@example.com>
- <news:comp.infosystems.www.servers.unix>
- <tel:+43-1-58801-58400>
- <telnet://192.0.1.8:25/>
- <urn:oasis:names:specification:docbook:dtd:xml:4.1.2>

URI / 1

- <http://www.ietf.org/rfc/rfc2396.txt>

Scheme part

Authority part

Hierarchical part

URI / 2

- <http://www.example.at/search?xyz=abc>

Query-Part

- <http://www.ex1.at/abc.html#my-anchor>

Anchor

HTTP / 1

- Protocol for Information Systems
 - Distributed, collaborative, hypermedia
 - In use by WWW initiative since 1990
- General idea: request-response
- HTTP/0.9
 - Simple protocol for raw data transfer across Internet
- HTTP/1.0 (RFC 1945)
 - Extended by allowing messages to use MIME-format
- HTTP/1.1 (RFC 2616)
 - More strict
- Standard Port: TCP 80

HTTP / 2

- Request
 - Request method
 - URI
 - Protocol version
 - MIME-like message
 - Request modifiers
 - Client information
 - Body content
 - Generic syntax: "Method Request-URI HTTP-Version"
- Response
 - Status line
 - including message protocol version
 - Success or error code
 - MIME-like message
 - Server information
 - Entity metainformation (content-type, length, date of modification, ...)
 - Entity-body content

HTTP / 3 – Request methods

- GET
 - Retrieve information identified by Request-URI
 - May refer to a process instead to a data entity
 - See Dynamic Web
 - Conditional GET
 - if request message contains additional header fields
 - Eg. If-Modified Since, If-Match, If-None-Match, If-Range
 - Goal to reduce bandwidth
- HEAD
 - Like GET but does not return message-body
 - HTTP header identical

HTTP / 4 – Request methods

- POST
 - Requests entity enclosed in request as additional item for entity identified in Request-URI
 - URI determines handler for the post
 - Examples
 - Annotation of existing resources
 - Posting a message to bulleting boards, newsgroups, ...
 - Providing a block of data, such as the result of submitting a form, to a data-handling process
 - Extending a database through append operation
 - Actual Function determined by server
 - Response contains result of the action

HTTP / 5 – Request methods

- OPTIONS
 - Communication options available on the request/response chain identified by URI-Request
- PUT
 - Enclosed entity shall be stored under supplied Request-URI
- DELETE
 - Delete resource identified by Request-URI
- TRACE
 - Debugging method
- CONNECT
 - For proxies to dynamically switch being a tunnel (SSL)

HTTP / 6 – Status Codes

- Informational 1xx
 - Prior regular response
 - If unexpected May be ignored
 - Proxies must forward 1xx responses
 - 100 Continue
 - Client SHOULD continue with its request
- Successful 2xx
 - Request successful
 - 200 OK
 - 201 Created, 202 Accepted,...

HTTP / - Status Codes

- Redirection 3xx
 - Further actions need to be taken by user to fulfill request
 - 301 Moved Permanently
 - New URI given in Location field of response
 - If possible client shall change link
 - 302 Found
 - New URI given in Location field of
 - 303 See Other
 - Similar to 302 but different URI should be retrieved with GET
 - Primarily to allow output of POST-activated script to redirect user agent
 - 304 Not Modified
 - For conditional GET requests

HTTP – Status Codes

- Client Error 4xx
 - 400 Bad request
 - 401 Unauthorized
 - 402 Forbidden
 - Authorization won't help, shall not be repeated
 - 404 Not Found
 - No match found for Request-URI
 - 408 Request Timeout
 - 410 Gone
 - Resource no longer at server

HTTP – Status codes

- Server Error 5xx
 - 500 Internal Server Error
 - 501 Implementation
 - 503 Service Unavailable
 - Overloading of server
 - 505 HTTP Version Not supported

HTTP – Persistent Connections

- HTTP connection closed after one request
 - Assumption that client fetches more requests to same server
 - Standard in HTTP/1.1 persistent connection desired
 - Controlled with Connection: close / keep-alive header
 - Server time-out closes connection automatically
- Advantages
 - Opening/closing fewer TCP connections
 - CPU time saved in routers and all participating hosts
 - Fewer packets caused by TCP opens
 - HTTP requests/responses pipelined
 - Client make multiple requests on same TCP connection without waiting for a response
 - Latency of subsequent requests reduced
 - No time spent in TCPs connection opening handshake

HTTP State Management

- HTTP Sessions to manage state
 - HTTP is stateless
 - Server manages variables for each session
 - Session-ID used to identify session in requests
- Identification of session
 - URL-Rewriting
 - `http://www.example.com?sessionID=SID1234`
 - HTML Hidden Field
 - `<input type="hidden" name="sessionID" value="SID1234"/>`
 - Cookies
 - Additional Request-Header-Field
 - `Cookie: $Version="1"; sessionID="SID1234"`
 - Cookie generated by server
 - Sent to user agent in response field
 - `Set-Cookie2: $Version="1"; sessionID="SID1234"`

HTTP Authentication

- Methods to authenticate users
 - Restrict access to resources
- Not secure unless used with external secure system (eg. SSL)
- Based on challenges
 - Server poses a challenge to client
 - Client has to response with correct answer
- Restriction is based on realms
 - String value
 - Defines/Names protection space

HTTP Authentication

- C: requests protected resource
- S: 401 Unauthorized
 - WWW-Authenticate header field includes at least one challenge that must be fulfilled by client
- C: Authorization header field in request
 - Contains credentials containing authentication information for a realm
- Server responds with resource

HTTP Authentication - Basic

- Client identifies itself with UserID & Password
- Challenge: "Basic" realm
 - `WWW-Authenticate: Basic realm="WaynesWorld"`
- Credentials
 - `"UserID:Password" base64 encoded`
 - `Authorization: Basic XYZ1235456==`
- Weak
 - Problem: Base64 bijective

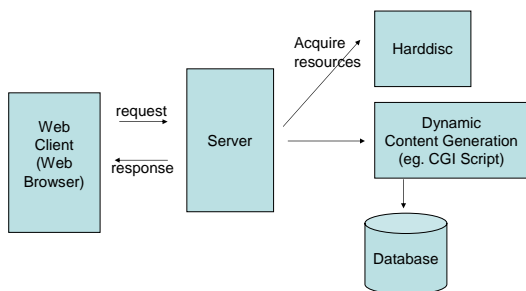
HTTP Authentication - Digest

- Challenge
 - contains a "nonce" value
- Valid response contains a checksum
 - Username + Password + nonce + HTTP method + Request-URI
 - Default uses MD5 checksums (128bit)
- Password never sent in the clear
- Quality of Protection (qop)
 - Different protection levels
 - Authentication, Integrity checking, Confidentiality checking

HTTP Authentication - Digest

- WWW-Authenticate: Digest
 - realm="WaynesWorld",
 - nonce="dcd98b1234567890acd23467",
 - opaque="12345",
- Authorization: Digest username="Wayne"
 - realm="WaynesWorld",
 - nonce="dcd98b1234567890acd23467",
 - uri="/index.html",
 - response="67890abcdef1234567890ab"

Web Server

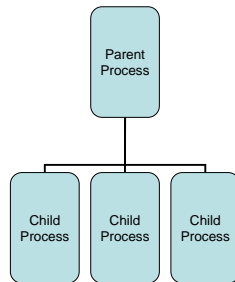


Web Server / Today

- Apache 1.3
 - Market leader
- Apache 2.0 / 2.1
- iPlanet (Sun ONE)
- Internet Information Server (IIS) 6.0

Apache Architecture 1.3

- Prefork model
 - Controlling parent process
 - Configurable number of child processes exist
- Each child listens on a socket for a request
- Parent process watches if appropriate number of child processes exist
 - Adjusts number of processes
- Characteristics
 - Robustness & Reliability
 - Problem: scalability on some platforms (processes heavyweight)
 - Security
 - all pages readable by user running child processes



Apache Architecture 2.0

- Based on Multi-Processing Module (MPMs)
 - Library of routines available on all platforms
 - Threading/Processing
 - Input/Output
- Prefork MPM (like in Apache 1.3)
- Worker MPM
- Perchild MPM
- WinNT MPM / OS-specific MPM

Apache 2.0 Architecture / Worker

- Hybrid thread/process MPM
 - Parent creates number of child processes
 - Each child has static number of threads
 - On thread/client devoted to listen to network
 - Each process may serve multiple requests
 - When server connects more connections
 - New client processes are created
 - Same number of threads
- Characteristics
 - Improved scalability
 - Less Reliability & Robustness
 - If One thread crashes entire process will terminate

Apache 2.0 Architecture / Perchild

- Hybrid thread/process MPM
 - Creates specific number of child processes
 - Each has specific number of threads
 - If load increases
 - One child creates new thread(s)
 - Number of child processes static
- Characteristics
 - Most scalable on the right platform (eg. Unix)
 - Least reliable
 - Administrators will configure fewer processes
 - The more threads run in one process the less reliable it becomes
 - Each child process can run as another user or group
 - Good when hosting many virtual hosts because security contexts are separated

Apache 2.0 Architecture / WinNT

- Single child process / multiple threads
- Takes advantage of Windows NT kernel features
 - Reuse of sockets created for previous requests
 - Increase of scalability by use of Completion ports
 - 2 operations at the same time: accepts a connection and reads first packet
- Characteristics
 - Child Process dies
 - Server becomes unresponsive
 - May run as a Windows NT server

Apache 2.0 / Architecture

- Extensible
 - By modules
 - Large variety of modules
 - mod_ssi (Server Side Includes)
 - mod_cgi (cgi scripts)
 - mod_dav (WebDAV)
 - mod_auth_basic, mod_auth_digest
 - ...

Apache 2.0

WebDAV

- Digital Authoring & Versioning (RFC 2518)
- Extends HTTP
 - Authoring of documents via HTTP
 - Kind of file system
 - HTTP URL namespace model
 - Accessible via HTTP (hence, Internet)
 - Platform independent
- Method parameter information
 - HTTP header (like in HTTP/1.1)
 - Encoded in XML request entity body

WebDAV / Terms

- Properties
 - Data about data (eg. Author, subject, ...)
- Collections
 - New type of Web resource
 - State consists of at least a list of internal members (resources itself)
- Locking
 - Ability to keep more than one person from working on a document

WebDAV

- HTTP methods for properties
 - Ability to create, remove, and query information about resources
- HTTP methods for collections
 - Ability to create sets of documents and to retrieve hierarchical membership listings (similar to file system directories)

WebDAV / HTTP methods

- PROPFIND
 - Retrieves properties defined on a resource
- PROPPATCH
 - Set/remove properties on a resource
- MKCOL
 - Create new collection
- GET,HEAD for collections
 - Retrieve whatever information is identified by Request-URI

WebDAV / HTTP methods

- DELETE
 - Depth infinity
- PUT
- COPY & MOVE
 - Copies or moves a resource/collection
 - Overwrite may be requested
 - Nesting depth may be provided for collections

WebDAV - Versioning

- What about the V in WebDAV?
- Not included in original WebDAV
 - RFC (2518)

WebDAV - Versioning

- Versioning Extensions (RFC 3253)
 - Defines extension to existing HTTP and WebDAV methods
 - New Resource types (properties & methods)
- Basic Versioning Features
- Advanced Versioning

WebDAV – Basic Versioning

- Goals
 - Put a resource under version control
 - Determine whether a resource is under version control
 - Determine whether a resource update will automatically be captured as a new version
 - Create and access distinct versions of a resource

WebDAV – Basic Versioning

- Methods
 - VERSION-CONTROL
 - Create a version-controlled resource at Request-URI
 - REPORT
 - Returns information about a resource (infos about multiple versions)
 - CHECKOUT
 - Applied to a checked-in version-controlled resource to allow modifications
 - CHECKIN
 - Applied to a checked-out version-controlled resource to produce a new version

WebDAV – Creating a Version-Controlled Resource

- VERSION-CONTROL request-URI
 - On versionable resource
- Creates 2 new resources
 - "Version History Resource"
 - Not necessarily visible in http scheme URL space
 - "Version Resource" (=Version)
 - Added to new version history resource
 - New and distinct URL
- Converts versionable resource into a "version-controlled" resource
 - Identified by same resource as original versionable

WebDAV – CheckIn/CheckOut

- CheckIn
 - Goal: user wants to add a new version of a resource (after modification)
 - New "Version Resource" created
 - Added to "Version Resource History"
 - Become active resource

WebDAV – Advanced Versioning

- Goals
 - Parallel development
 - Configuration management of sets of web resources
 - Similar what CVS, Subversion, Perforce, etc can already do
- Methods
 - MERGE simultaneous changes

WebDAV - Extensions

- WebDAV Ordered Collections Protocol
 - RFC 3648
 - Server-side support for ordering of collection members
 - Client may change order
- WebDAV Access Control Protocol
 - RFC 3744
 - Permits clients to read and modify access control lists with permissions for resources on the server

WebDAV – Request Sample

PROPFIND /mydocs/thebible HTTP/1.1
Host: www.server.com
Depth: 1
Content-Type: text/xml; charset="utf-8"
Content-Length: xxxx

```
<?xml version="1.0" encoding="utf-8"?>
<D:propfind xmlns:D="DAV:">
  <D:prop xmlns:R="http://www.server.com/mydocs/">
    <R:author/>
    <R:creation-date/>
  </D:prop>
</D:propfind>
```

Retrieves Named Properties

WebDAV – Response Sample

HTTP/1.1 207 Multi-Status
Content-Type: text/xml; charset="utf-8"
Content-Length: xxxx

```
<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:">
  <D:response>
    <D:href>http://www.server.com/mydocs/thebible.doc</D:href>
    <D:propstat>
      <D:prop xmlns:R="http://www.server.com/mydocs/">
        <R:author>
          <R:Name>unknown</R:Name>
        </R:author>
      </D:prop>
      <D:status>HTTP/1.1 200 OK</D:status>
    </D:propstat>
  </D:response>
</D:multistatus>
```

File Transfer Protocol

- RFC 959
- Started in 1971 (!), RFC 114
- Transfer of a file from one host to another
- Based on 2 connections
 - Control connection (Server TCP port 21)
 - Data connection created each time a file is transferred (Server TCP port 20)
- Uses TELNET NVT protocol on control connection
- Limited number of file types supported
 - ASCII, Binary, sometimes EBCDIC

File Transfer Protocol

- Client initiates always Control connection
- Active FTP
 - Client opens a random data port for listening
 - Server connects to this open client data port with its own port 20
 - Firewall problem – server has to go through client firewall
- Passive FTP
 - Server listens on data port (not port 20)
 - Client connects to open server data port
 - Not all FTP clients/servers support passive FTP

FTP commands

- Access Control
 - USER & PASS
 - CWD (Change Working Directory)
- Transfer Parameter Commands
 - PORT – specifies data port
 - PASV – passive mode
 - TRANSFER MODE (Stream, Block, Compressed)
- Service Commands
 - RETR retrieve a file)
 - STOR store a file
 - LIST list files
 - RMD, MKD remove, make directory
 - STATUS

WWW Robot / 1

- Program that traverses hypertext structure
 - Automatically by following hyperlinks
 - Also called Spider, or Web crawling
- Primarily used for Search Engines
 - index and rank documents
 - Extract references to external documents
- Problems with robots
 - Rapid fire: many (parallel retrievals in a short time)
 - Server and network being overloaded
 - Black hole: endless loop, robot trapped

WWW Robots / 2

- "How do I Get My Site Listed on Google"
 - <http://www.google.com/webmasters/1.html>
 - URL Form: added to a list but not into search engine
 - On next crawl URL robot is applied on this URL
- Robot Exclusion
 - Not desirable to include all files in Robot traversal
 - File on Local URL: /robots.txt
 - User-Agent: *
 - Disallow: /tmp/
 - Alternative: Meta-Tag in HTML files
 - `<meta name="robots" content="noindex,nofollow">`

WWW Caching

- Browser cache
 - Included in Web browser
 - Checks if representation stored on local disc is up-to-date
- Proxy cache
 - Larger scale (100-1000s users)
 - Good at reducing latency and network traffic
 - For Popular representations used in departments/companies, ...
 - Examples
 - Squid (www.squid-cache.org),
 - MS Internet Security and Acceleration Server
- Gateway cache
 - To make sites themselves more scalable
 - Eg. Akamai

WWW Caching

- HTML Meta Tag
 - META No-cache
 - Problem: not all browsers support it
- HTTP Header
 - Expires: Thu, 2 Jun 2005 13:10:00 GMT
 - Good for files that change rarely
 - Clock synchronisation of WebServer and cache
 - Cache-Control response Header
 - no-store, max-age (similar to expires but relative)
 - no-cache (cache submits request to server)
- Internet Cache Protocol (RFC 2186, RFC 2187)
 - Synchronisation of Caches
 - More lightweight than HTTP
 - On miss a cache submits an ICP request to cache siblings
 - Returns HITS and/or MISSES
 - Original cache uses these returns to resolve its own miss (via HTTP)