

# CS615 - Aspects of System Administration

## Popular Services – HTTP, SNMP, SSH

---

Department of Computer Science  
Stevens Institute of Technology  
Jan Schaumann

`jschauma@cs.stevens.edu`

`http://www.cs.stevens.edu/~jschauma/615A/`

# HTTP: Hypertext

---

W W W

## HTTP: Hypertext

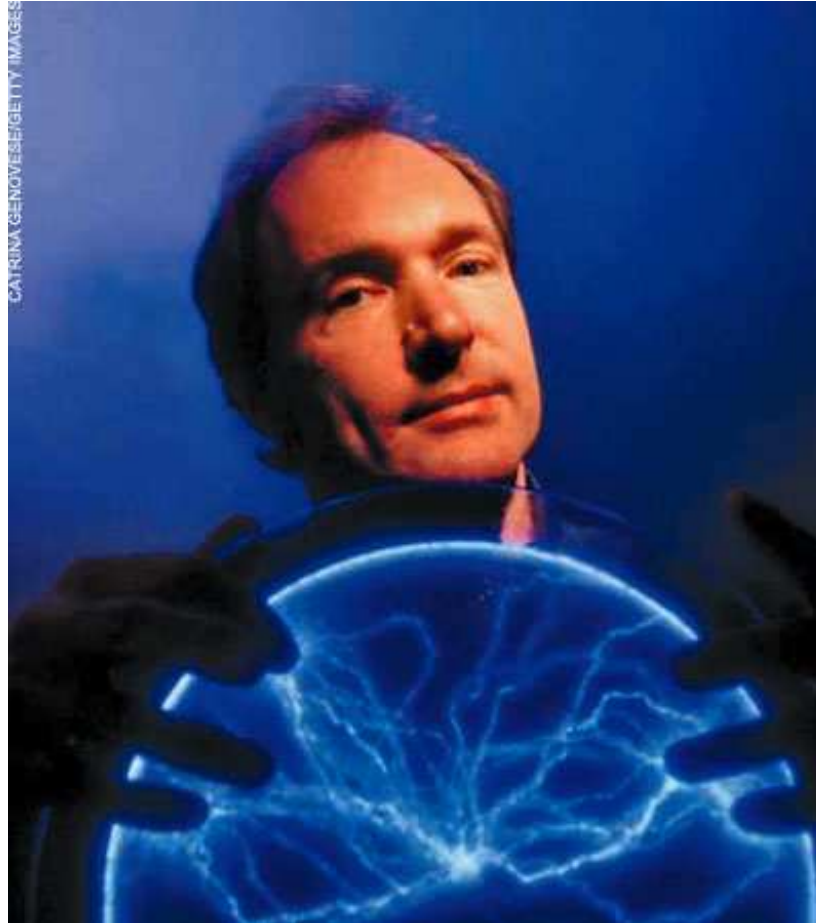
---

W W W

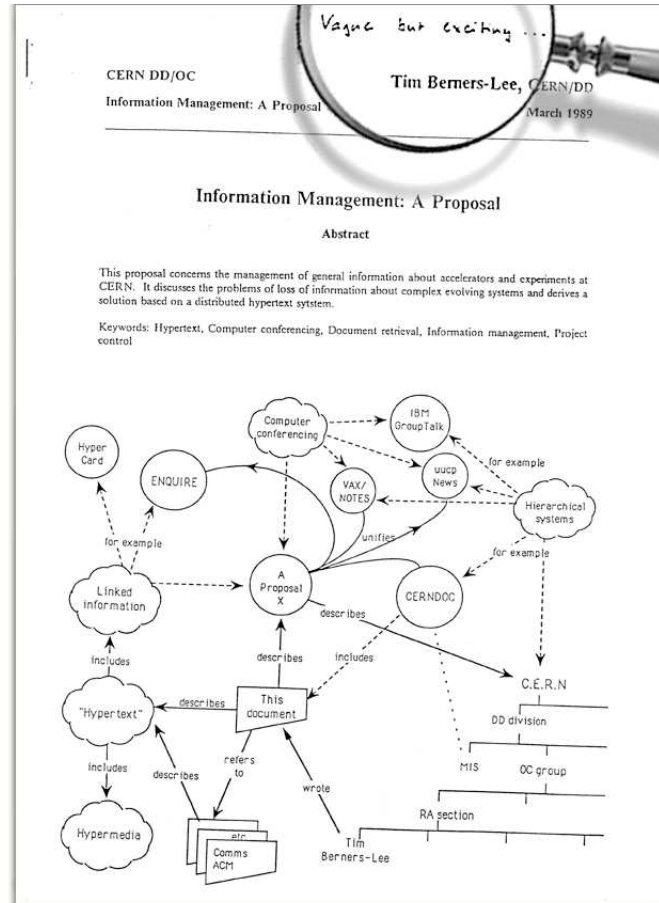
*“The World Wide Web is the only thing I know of whose shortened form takes three times longer to say than what it’s short for.” – Douglas Adams*

# HTTP: Hypertext

---

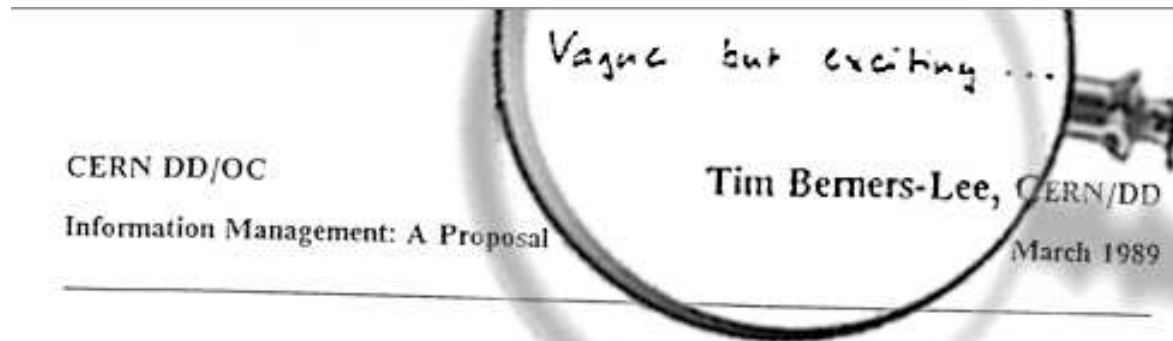


# HTTP: Hypertext



# HTTP: Hypertext

---



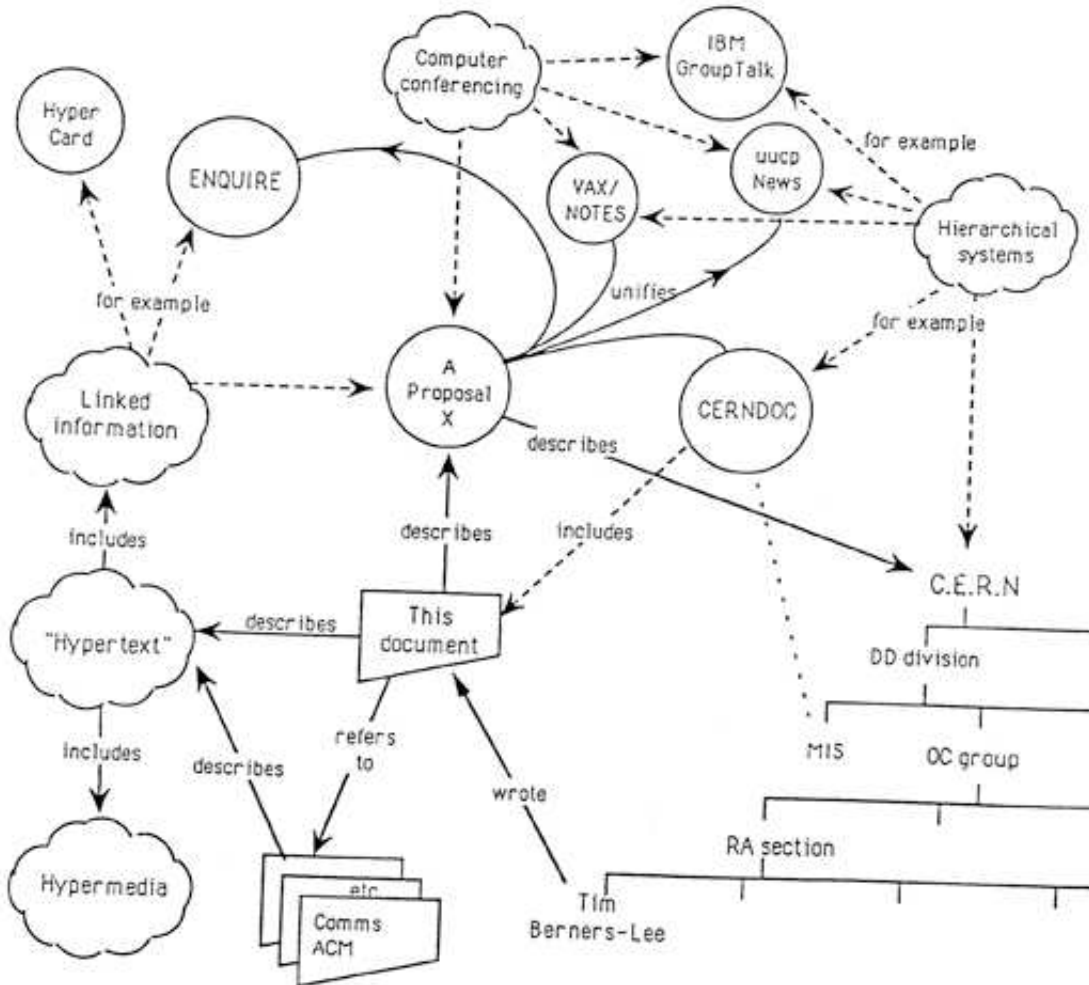
## Information Management: A Proposal

### Abstract

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project control

# HTTP: Hypertext



HTTP

---

# Hypertext Transfer Protocol

## RFC2616

# HTTP

---

HTTP is a request/response protocol.

## HTTP: A client request

---

```
$ telnet www.yahoo.com 80
Trying 69.147.76.15...
Connected to www-real.wa1.b.yahoo.com.
Escape character is '^]'.
GET / HTTP/1.0
```

## HTTP: a server response

---

```
HTTP/1.1 200 OK
Date: Thu, 09 Apr 2009 17:37:06 GMT
P3P: policyref="http://info.yahoo.com/w3c/p3p.xml", CP="CAO DSP COR CUR ADM
DEV TAI PSA PSD IVAi IVDi CONi TELo OTPi OUR DELi SAMi OTRi UNRi PUBi IND PHY
ONL UNI PUR FIN COM NAV INT DEM CNT STA POL HEA PRE LOC GOV"
Cache-Control: private
Vary: User-Agent
X-XRDS-Location: http://open.login.yahooapis.com/openid20/www.yahoo.com/xrds
Last-Modified: Thu, 09 Apr 2009 17:36:16 GMT
Accept-Ranges: bytes
Content-Length: 9490
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<html>
<head>
<title>Yahoo!</title>
```

# The Hypertext Transfer Protocol

---

HTTP is a request/response protocol:

1. client sends a request to the server
2. server responds

# The Hypertext Transfer Protocol

---

HTTP is a request/response protocol:

1. client sends a request to the server
  - request method
  - URI
  - protocol version
  - request modifiers
  - client information
2. server responds

## HTTP: A client request

---

```
$ telnet www.yahoo.com 80
Trying 69.147.76.15...
Connected to www-real.wa1.b.yahoo.com.
Escape character is '^]'.
GET / HTTP/1.0
```

# The Hypertext Transfer Protocol

---

HTTP is a request/response protocol:

1. client sends a request to the server
  - request method
  - URI
  - protocol version
  - request modifiers
  - client information
2. server responds
  - status line (including success or error code)
  - server information
  - entity metainformation
  - content

## HTTP: a server response

---

```
HTTP/1.1 200 OK
Date: Thu, 09 Apr 2009 17:37:06 GMT
P3P: policyref="http://info.yahoo.com/w3c/p3p.xml", CP="CAO DSP COR CUR ADM
DEV TAI PSA PSD IVAi IVDi CONi TELo OTPi OUR DELi SAMi OTRi UNRi PUBi IND PHY
ONL UNI PUR FIN COM NAV INT DEM CNT STA POL HEA PRE LOC GOV"
Cache-Control: private
Vary: User-Agent
X-XRDS-Location: http://open.login.yahooapis.com/openid20/www.yahoo.com/xrds
Last-Modified: Thu, 09 Apr 2009 17:36:16 GMT
Accept-Ranges: bytes
Content-Length: 9490
Connection: close
Content-Type: text/html; charset=utf-8
```

```
<html>
<head>
<title>Yahoo!</title>
```

## The Hypertext Transfer Protocol

---

Server status codes:

- 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

# HTTPS

---

HTTPS == HTTP over SSL or TLS

- use of a different server port (443)
- basically use of HTTP protocol just as with plain TCP
- connection initiation:
  - client initiates a connection to the server
  - client sends TLS ClientHello
  - client and server perform TLS handshake (according to TLS protocol (RFC 2246))
  - client makes regular HTTP requests

## The WWW Common Gateway Interface

---

The Common Gateway Interface (CGI) is a simple interface for running external programs under an information server in a platform-independent manner.

- in use since about 1993
- files identified as CGI scripts or programs are *executed* rather than served
- parameters may be passed
- CGI to generate output suitable to be returned to client (including HTTP headers)

## General (Web)Server Considerations

---

### Server startup:

- sanity checking (configuration files, lock files)
- bind to proper port
- spawn a number of children to handle requests
- serve requests
- log requests and errors

## The Apache webserver

---

- usually runs the *httpd* program as a *dæmon*
- usually runs as a dedicated user such as *nobody* or *www*
- can bind to specific IP-address/port pairs
- supports so-called *virtual hosts*
- supports a large number of *modules*, either specified at compile time or dynamically loaded modules. Examples:
  - *mod\_access* – access control based on client hostname, IP address etc.
  - *mod\_auth* – user authentication
  - *mod\_cgi* – execution of CGIs
  - *mod\_rewrite* – rule-based rewriting engine to rewrite requested URLs on the fly
  - *mod\_suexec* – execution of CGIs as specified users/groups

## Apache Virtual Hosts

---

*Virtual Hosts* allows you to run more than one web site on a single machine.

- *Virtual Hosts* can be
  - IP-based
  - port-based
  - name-based
- name-based *Virtual Hosts* do not work with SSL

## Apache log files

---

- two general log files
  - error log
  - access log
- customizable log formats
- some pre-defined log formats

- Common Log Format

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

- Combined Log Format

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
```

## Apache log files

---

```
$ hostname
murky.cs.stevens.edu
$ more /usr/people/guest/weblogs/cs/www-access.log
155.246.141.170 - - [09/Apr/2009:13:55:22 -0400] "GET /Common/P1.jpg HTTP/1.1" 200 33474
155.246.141.170 - - [09/Apr/2009:13:55:22 -0400] "GET /Common/P5.jpg HTTP/1.1" 200 41599
155.246.141.170 - - [09/Apr/2009:13:55:22 -0400] "GET /Common/P8.jpg HTTP/1.1" 200 31095
155.246.141.170 - - [09/Apr/2009:13:55:22 -0400] "GET /Common/P4.jpg HTTP/1.1" 200 35338
155.246.141.170 - - [09/Apr/2009:13:55:23 -0400] "GET /tl_curve_white.gif HTTP/1.1" 404 861
155.246.141.170 - - [09/Apr/2009:13:55:23 -0400] "GET /tr_curve_white.gif HTTP/1.1" 404 861
155.246.141.170 - - [09/Apr/2009:13:55:23 -0400] "GET //Common/Stevens_logo_CS.gif HTTP/1.1" 200 3590
155.246.141.170 - - [09/Apr/2009:13:55:23 -0400] "GET /favicon.ico HTTP/1.1" 200 3262
201.11.251.8 - - [09/Apr/2009:13:55:23 -0400] "GET /~rbender/otakon05/regs/img_5091.jpg HTTP/1.0" 200 73234
72.14.199.115 - - [09/Apr/2009:13:55:35 -0400] "GET /support/rss/update.xml HTTP/1.1" 302 362
155.246.140.226 - - [09/Apr/2009:13:56:55 -0400] "GET /ProgramInfo/bs_cs_current.php HTTP/1.0" 200 176209
128.82.52.231 - - [09/Apr/2009:13:58:21 -0400] "GET /~amiasnik/MA222/P1/MA222Project.pdf HTTP/1.1" 200 72939
155.246.231.27 - - [09/Apr/2009:13:57:44 -0400] "GET /~jschauma/615A/quiz4.txt HTTP/1.1" 200 4120
94.112.159.220 - - [09/Apr/2009:13:58:57 -0400] "GET /%7Erbender/otakon06/regs/img_0872.jpg HTTP/1.1" 200 40331
$ more /usr/people/guest/weblogs/cs/www-errors.log
[Tue Aug 01 05:19:54 2006] [error] [client 69.12.139.109] File does not exist:
    /home/mbubb/public_html/CS520/kernel/null, referer: http://www.cs.stevens.edu/~mbubb/CS520/kernel/kernel.html
[Tue Aug 01 05:23:24 2006] [error] [client 158.132.12.81] File does not exist:
    /home/quynh/public_html/student-work/cs638-sp05/spinCorrelate_files
[Tue Aug 01 06:09:50 2006] [error] [client 64.4.8.120] File does not exist:
    /usr/local/web/websites/stevens/cs/www/robots.txt
[...]
[Thu Apr 09 14:00:08 2009] [crit] [client 155.246.89.207] (13)Permission denied:
    /home/ysun/.htaccess pcfg_openfile: unable to check htaccess file, ensure it is readable
[Thu Apr 09 14:01:46 2009] [error] [client 155.246.125.70] File does not exist:
    /usr/local/web/websites/stevens/cs/www/tl_curve_white.gif, referer: http://www.cs.stevens.edu/
```

## Apache Authentication, Authorization and Access Control

---

- *.htaccess* files provide a way to make configuration changes on a per-directory basis.
- *AllowOverride* directive specifies what options can be set in *.htaccess*
- *htpasswd* command is used to create username/password pairs
- authorization can be done for users or groups
- authentication based on where users are connecting from by use of *Deny from* or *Allow from* directives

## Dynamic Content with CGI under Apache

---

- set *ScriptAlias* for a global directory
- use *AddHandler / SetHandler* to designate files or subdirectories as CGIs
- CGIs must produce a proper MIME-header
- things can go wrong:
  - Internal Server Error – check logs. Most likely cause of error: wrong permissions or missing MIME-header
  - Forbidden – wrong permissions on file or directory
  - you get source code of your CGI program or a POST Method Not Allowed message – Apache was not configured to allow execution of CGIs
- use *mod\_suexec* to allow execution of CGIs as another user (for example, to access resources that *www* does not have access to)

## Apache configuration file example

---

stevens.conf

# Simple Network Management Protocol

---

MIBs, OIDs, ASN.1, ... all that

# SNMP

---

A complete network management system to monitor network-attached devices.

# SNMP

---

Base concepts:

- managed devices run an *snmp agent* or *dæmon*
- information about the device is exposed in *management information bases*
- parts of a system are made available in *read-only* mode
- parts of a system may be made available in *write* mode
- certain conditions may trigger actions or *traps*
- normally uses UDP 161 for the *agent* and 162 for the *manager*

# SNMP

---

## Management Information Bases (MIBs):

- hierarchical namespace
- contains *Object Identifiers* (OIDs)
- written in *Abstract Syntax Notation One* (ASN.1)
- often vendor defined

## SNMP Versions

---

### SNMPv1:

- de-facto standard
- poor security (“community strings” act as passwords)

### SNMPv2:

- improvements in the area of performance (GETBULK instead of GETNEXT) and security
- comes in the flavors *SNMPv2c*, *SNMPv1.5* and *SNMPv2u*

### SNMPv3:

- official standard
- adds authentication, privacy and access control

## SNMP: An example

---

```
$ snmpwalk -c public -v 1 drude.cs.stevens.edu
SNMPv2-MIB::sysDescr.0 = STRING: SunOS drude 5.10 Generic i86pc
SNMPv2-MIB::sysUpTime.0 = Timeticks: (1744098103) 201 days, 20:43:01.03
IF-MIB::ifNumber.0 = INTEGER: 3
IF-MIB::ifDescr.1 = STRING: lo0
IF-MIB::ifDescr.2 = STRING: iprb0
IF-MIB::ifDescr.3 = STRING: iprb1
IF-MIB::ifType.1 = INTEGER: softwareLoopback(24)
IF-MIB::ifType.2 = INTEGER: infiniband(199)
IF-MIB::ifType.3 = INTEGER: infiniband(199)
IF-MIB::ifMtu.1 = INTEGER: 8232
IF-MIB::ifMtu.2 = INTEGER: 1500
IF-MIB::ifMtu.3 = INTEGER: 1500
IF-MIB::ifSpeed.1 = Gauge32: 127000000
IF-MIB::ifSpeed.2 = Gauge32: 100000000
IF-MIB::ifPhysAddress.2 = STRING: 0:e0:81:3:c9:b0
IF-MIB::ifInOctets.2 = Counter32: 890590019
IF-MIB::ifOutOctets.2 = Counter32: 110919181
RFC1213-MIB::atPhysAddress.2.1.155.246.89.1 = Hex-STRING: 00 09 44 D1 64 00
HOST-RESOURCES-MIB::hrStorageType.12 = OID: hrStorageFixedDisk
HOST-RESOURCES-MIB::hrStorageDescr.12 = STRING: /export/home
HOST-RESOURCES-MIB::hrStorageSize.12 = INTEGER: 34123365
HOST-RESOURCES-MIB::hrStorageUsed.12 = INTEGER: 36473
[...]
```

## Homework

---

Write a program that determines a given hosts network interfaces' bandwidth usage via SNMP. The program has the following usage:

```
bwinfo [-h] [-c community] [-i interface] [-t time] hostname
-c community  use the given community string
-h           print a short usage summary and exit
-i interface  only generate output for the given interface
-t time      use the given number in seconds in between pools
```

# SSH

---

secure encrypted terminal sessions

# SSH

---

“secure” replacement for telnet, rlogin, rsh

# SSH

---

## Components:

- sshd(8)
- ssh(1)

# SSH

---

## Components:

- sshd(8)
- ssh(1)
- scp(1)
- sftp(1)
- ssh-agent(1)
- ssh-keygen(1)

# SSH

---

Authentication done in primarily two modes:

- password authentication
- public-key authentication

# SSH

---

Authentication done in primarily two modes:

- password authentication
- public-key authentication

Communication *always* involves public-key encryption.

# SSH

---

Authentication done in primarily two modes:

- password authentication
- public-key authentication

Communication *a/ways* involves public-key encryption.

Except when it doesn't (`cipher:none`).

# SSH

---

## SSH *hostkeys*

- each host has (at least) one private key
- upon connection, it is verified against a public key
- discrepancies are reported (and should be investigated!)
- server and client perform a challenge-response handshake involving a random number (which becomes the session key) in SSHv1 or via Diffie-Hellman key agreement in SSHv2

# SSH

---

## SSH *userkeys*

- used during *public-key authentication*
- the user has a *private* key
- the remote host has the corresponding *public* key
- the private key may be passphrase protected
- the passphrase used never leaves the local host

# SSH

---

## SSH *agents*:

- allow the user to add multiple keys once, then no longer need to provide the passphrases
- agents can be *forwarded*
- communication happens through a unix-domain socket

## SSH configuration

---

```
sshd_config(5)  
ssh_config(5)
```

## Reading

---

### HTTP:

- RFC 2616
- RFC 2818
- <http://httpd.apache.org/docs/>
- <http://www.w3.org/Protocols/>
- <http://cgi-spec.golux.com/draft-coar-cgi-v11-03-clean.html>

### SNMP:

- [http://en.wikipedia.org/wiki/Simple\\_Network\\_Management\\_Protocol](http://en.wikipedia.org/wiki/Simple_Network_Management_Protocol)
- RFCs 1157, 3411, 3418 and others
- `snmpcmd(1)`