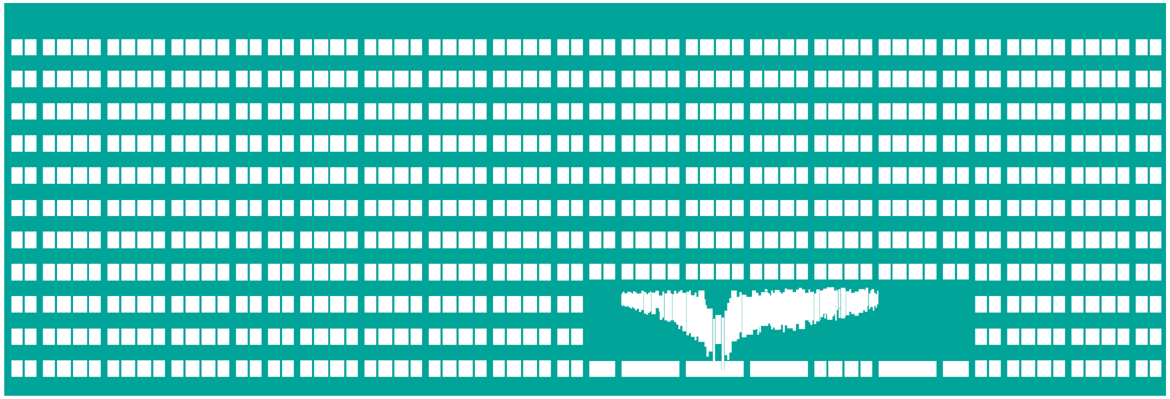
VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE | DEPARTMENT OF COMPUTER SCIENCE

# Application-layer Protocols and Internet Services



## Computer Networks
## Lecture 8

# Terminal Emulation

# Purpose of Telnet Service

- Supports remote terminal connected via network connection (TCP/23)
- Telnet server bridges characters between TCP connection and OS front-end
  - OS thinks that it interacts with the directly connected terminal

# Telnet Options

- Telnet protocol contains (binary) commands for mutual option negotiation
  - Option value proposal + ACK/NAK
  - Options are negotiated separately for both sides
  - Examples of Telnet options:
    - echo (local/remote)
    - terminal type
    - character/line mode
    - ...
- Supports process interruption (Ctrl-C) – using TCP Urgent Pointer

# Network Virtual Terminal – NVT

- NVT = virtual character-oriented device equipped with keyboard and printer
- Defines the common set of functions/parameters that must be supported by all real terminals
- Supposes 7-bit characters
  - In reality, all 8 bits are commonly passed
- Lines end with Carriage Return (CR) followed by Line Feed (LF )

# Secure Shell (SSH)

- Similar to Telnet but encrypts the character stream
  - Utilizes asymmetric cryptography
    - Encryption keys are negotiated dynamically using Diffie-Hellmann algorithm
  - Also support server authentication
- Based on the Secure Socket Layer (SSL) that provides universal encrypted connection-oriented transport service (TCP/22)
- SSH channel may be also used for another purposes
  - File transfer (scp), encrypted X-Window, ...

# File Transfer

# File Transfer Protocol (FTP)

- (bidirectional) file transfer between 2 systems
  - From client to server or from server to client
  - Direct transfer between 2 servers
- Supports user authentication
  - + authorization using the OS´s system of file access rights
- Uses separate control (TCP/21) and data (TCP/20) connection
  - Control connection - commands
  - Data connection (temporary)- transfer of file contents, directory listing
- Supports both transparent transfer (binary data) or line separator conversions (text files)

# Basic FTP Control Commands

- USER, PASS – user authentication
- CWD – change working directory
- LIST – directory listing
- PORT/EPRT – definition of clients TCP port for data connection from the server
    - e.g.  EPRT |2|1080::8:800:200C:417A|5282|
- TYPE – specification of file type (ASCII/binary)
- RETR, STOR – retrieve and store file
- DELE – delete a file
- ABOR – forceful transfer abortion
    - Control channel remains active for the duration of file transfer
- PASV/EPSV – change to passive mode
- QUIT – terminates the control connection
- …

# FTP Command Responses

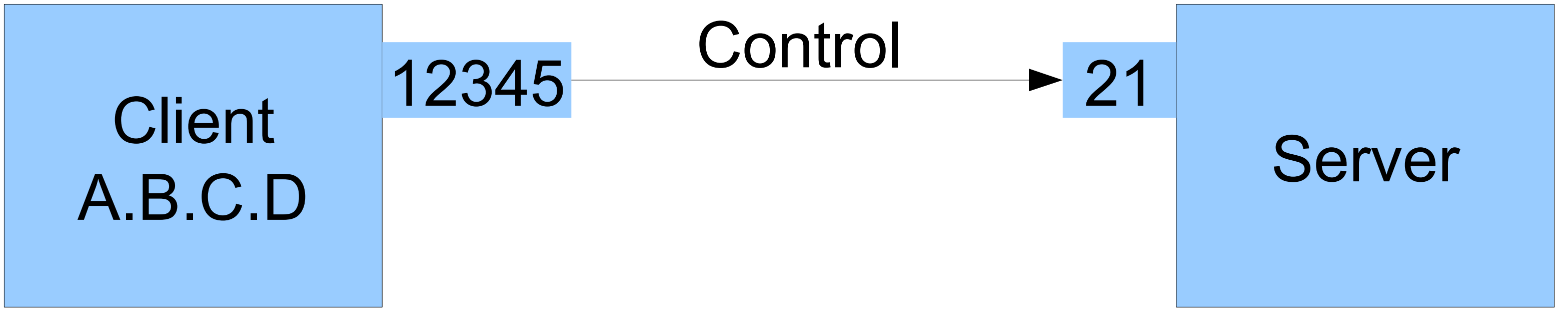
- 3-digit code for machine processing
  - Individual digits specify response type group and subgroup
- Auxiliary (variable) text for human user
- Multi-line responses are supported
  - First line contains - after response code
  - Following lines must not start with a number
  - Last line contains response code followed by space

*Note: Similar 3-digit codes will be found in many other Internet services*

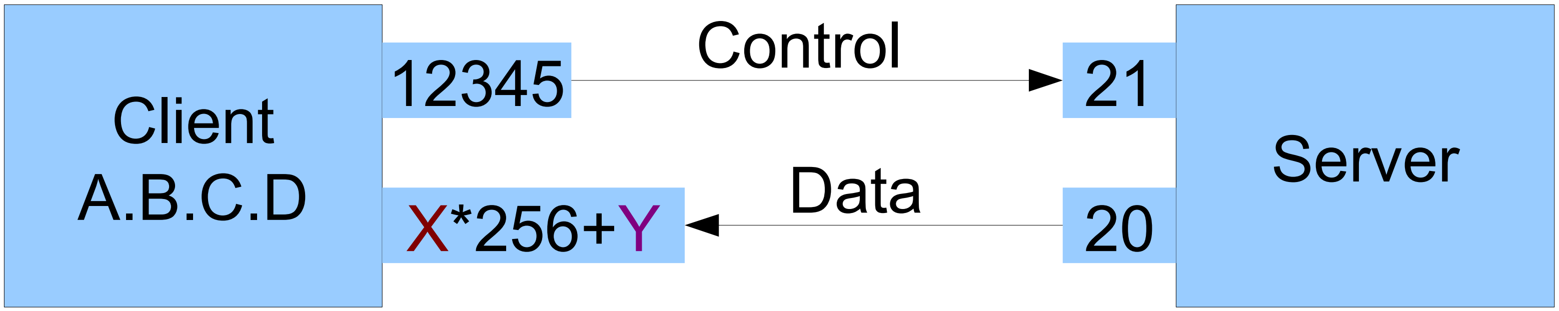
# FTP Active and Passive Mode

- Active mode (default)
  - Data connection is initiated by server
    - From port 20 to the client port previously specified using PORT command
- Passive mode (optional)
  - Entered after PASV command
  - Data connection is initiated by client
    - To port specified previously as a reply of  PASV command
  - Allows FTP to pass through the firewall that permits the connection initiation only from the inside network

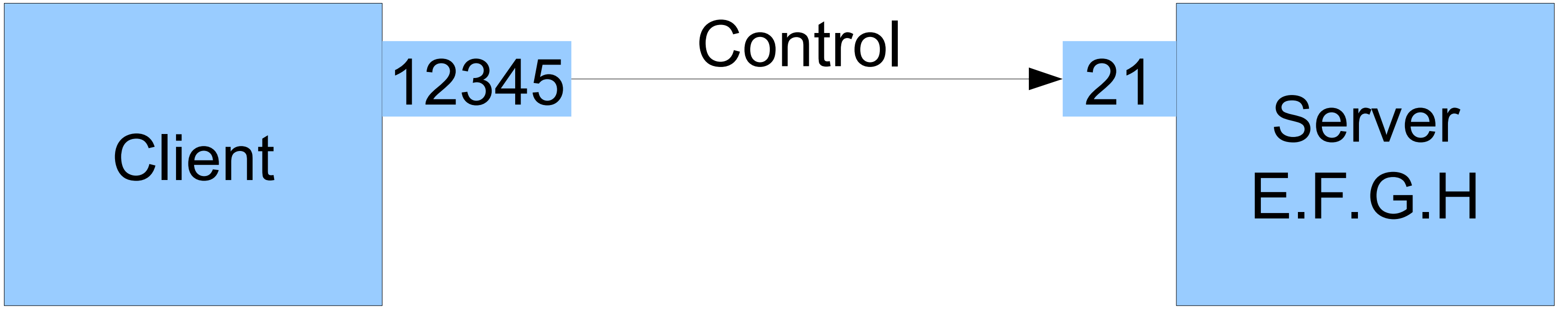
# Active Mode Example



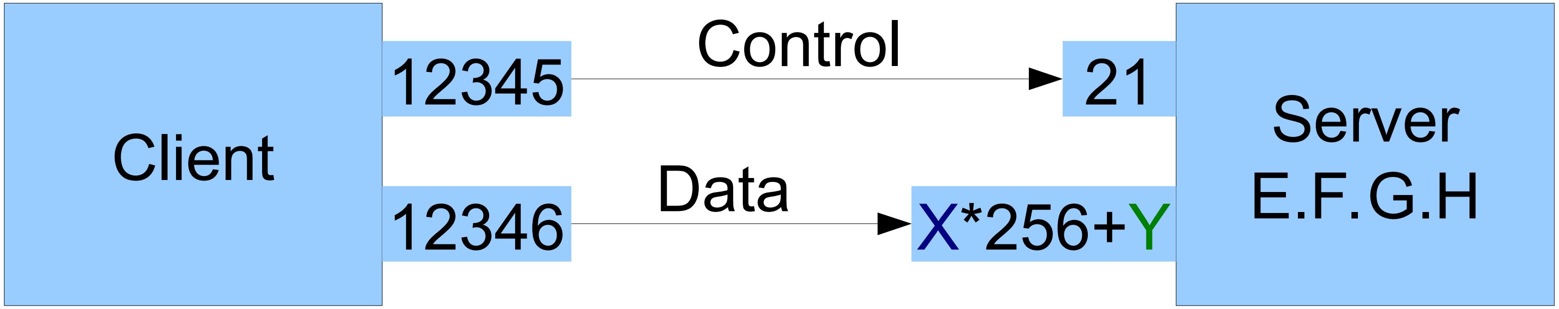
C → S:PORT A,B,C,D,X,Y
e.g. PORT 158,196,135,88,236,176



# Passive Mode Example



C → S: PASV
S → C: 227 Entering Passive Mode (E,F,G,H,X,Y).
e.g.      227 Entering Passive Mode (158,196,135,88,213,78).



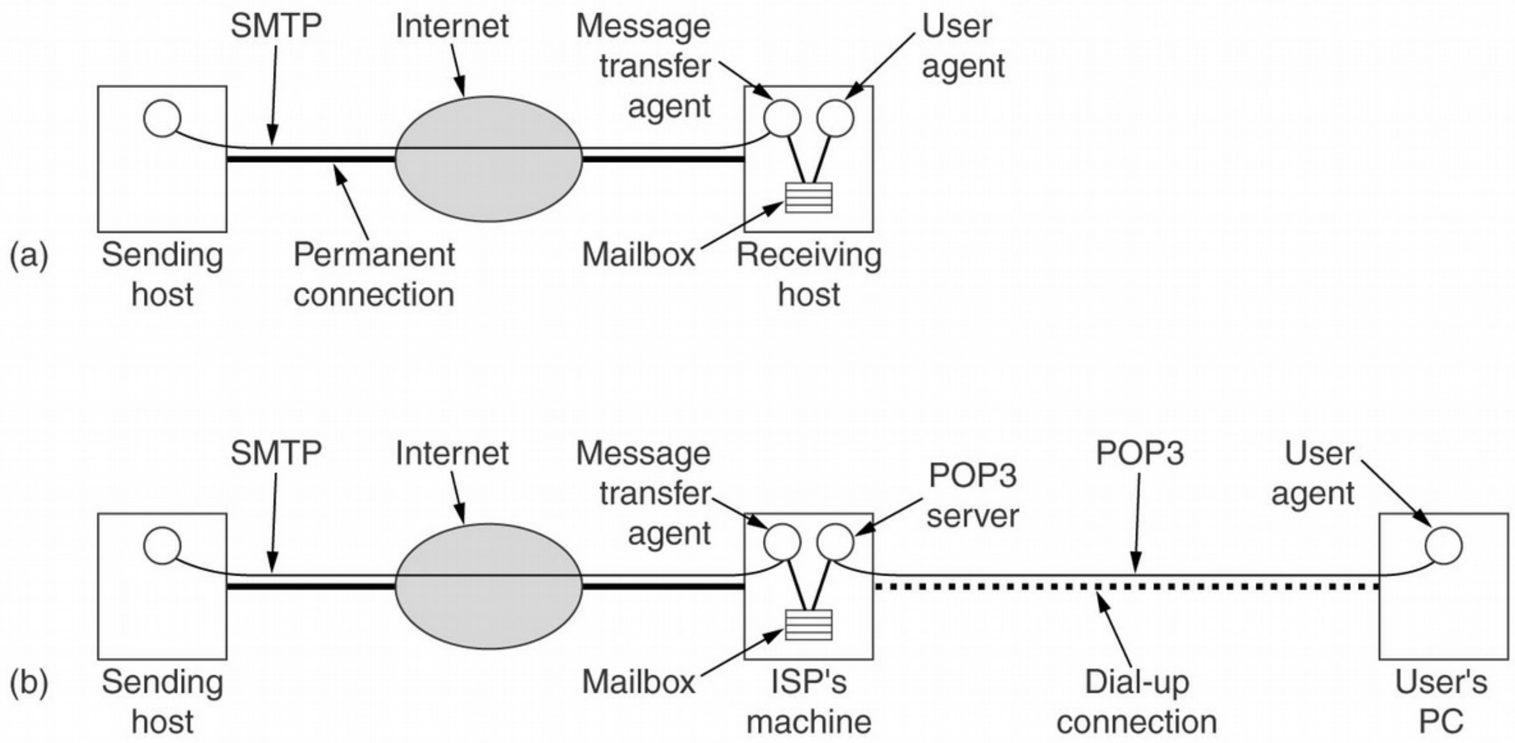
# Trivial FTP (TFTP)

- Simple file transfer implementation
- Runs over UDP
  - stop and wait protocol
  - first packet carries a file that have to be stored/retrieved
  - packets are numbered and acknowledged, lost frames are retransmitted
  - lacks any authentication
- Utilized for firmware upload, backing up and retrieval of system configuration and obtaining of OS image by diskless stations

# Electronic Mail

# Basic Terms and Roles

- User agent (UA)
- Message Transfer Agent (MTA)
- Mailbox



# E-mail Security

- All mail-related protocols are text-oriented
- No encryption is performed by default
- SMTP is not authenticated in most cases
  - there exists an authentication extension
- Users commonly authenticate using plaintext password with POP and IMAP
  - optionally, MD5 hash may be used
- Encryption may be accomplished using SSL (POP3S, IMAPS) or L7 extensions

# The Message Structure

- Envelope
  - Identifies sender and receiver
- Header
  - list of rows with "name: value" format
  - empty line serves as header end marker
  - Intermediate mail servers (MTAs) may add additional header lines
    - e.g. identities of MTAs involved in message delivery and times when they received the message
- Message Body

# Message Body

- Originally NVT ASCII (7-bit) without any specific structure
  - Max line length 1kB
  - Max message length 64kB
  - Higher limits are commonly applied today
- Multimedia Internet Mail Extension (MIME) was introduced later
  - A message may consist of multiple media of various types encoded in various formats
    - Multipart messages
  - Description of the message structure and media types/ encoding formats is contained in the message header

# Relaying of the Message

- Directly from sender mail client to mail server with receiver's mailbox
  - Problematic for dial-in clients in case if the message cannot be delivered immediately
- From sender mail client to a proxy mail server (outgoing SMTP gateway)
  - Mail servers pass the message hop-by-hop reliably
  - If a mail server cannot forward the message, the attempt is repeated
    - After a maximum number of unsuccessful attempts is reached, message undeliverable error is generated
- From the mail client or mail server to a mail gateway to non-SMTP network

# Simple Mail Transfer Protocol (SMTP)

- Used to pass messages from mail client to mail server or between mail servers
- TCP/25, TCP/587 (submission, SMTPS), TCP/465 (SMTPS, not official anymore)
  - unencrypted
  - usually not authenticated
- Text-oriented commands
- Multiple messages may be passed over a single SMTP session
  - In both directions (TURN command)

# Basic SMTP Commands

- HELO - identifies "client" MTA/MUA
    - EHLO is used nowadays instead to detect extra features
    - does not provide the „real“ authentication
- MAIL FROM: – identification of the sender
- RCPT TO: – identification of the receiver
    - Multiple receivers may be specified
- DATA – marks the beginning of message data
    - Header + message body
- „.“ as a first and only character on the line marks the end of message data (MUA changes . to ..)
- TURN – exchange of client and server roles
- QUIT - terminates SMTP session and TCP connection
- Extra: AUTH, STARTTLS, SIZE (in MAIL FROM:), …

# Some Additional SMTP Commands

- VRFY – verifies the existence of the mailing list
- EXPN – returns a list of e-mail addresses of the mailing list participants

Often forbidden for security reasons

# Multimedia Mail Extension (MIME)

- Allows to structure a message body
- Specifies how to interpret data of individual message parts
  - type/subtype, e.g. text/html
- Defines how are binary data encoded
  - As original message structure specification considered only ASCII data
- Description of message structure is carried in additional header fields

# MIME Headers

- MIME-Version:
- Content-Type:
  - text, multipart, message, application, image, audio, video
- Content-Transfer-Encoding
  - 7bit (NVT ASCII, default), quoted-printable, base64, binary
- Content-Description:

# Post Office Protocol v. 3 (POP3)

- Used to retrieve user's mailbox
- Client-server architecture
- Operates over TCP/110 (POP3s – TCP/995)
  - Unencrypted
  - Remote user is authenticated using plaintext password or MD5 hash
- Text-oriented control commands
- Responses are prefixed by + , +OK, or -ERR instead of numeric responses typical for other protocols (+ for commands sending e.g. a nonce)

# Basic POP3 commands

- USER, PASS – user authentication
  - APOP – authenticates using MD5 hash of the  timestamp sent from the server concatenated with user's  password
- LIST – list of IDs of the stored messages
- STAT – number and size of all messages
- RETR – retrieve a message with a given ID
  - (directly to the same TCP connection)
- DELE – mark a message for deletion
- RSET – unmarks all messages marked for deletion
- QUIT – deletes marked messages, quits the session and TCP connection
- TOP – retrieves first N lines of a given message
- UIDL – provides a permanent ID of a given message (persists between sessions)
- Newer extras: STLS, AUTH, SASL

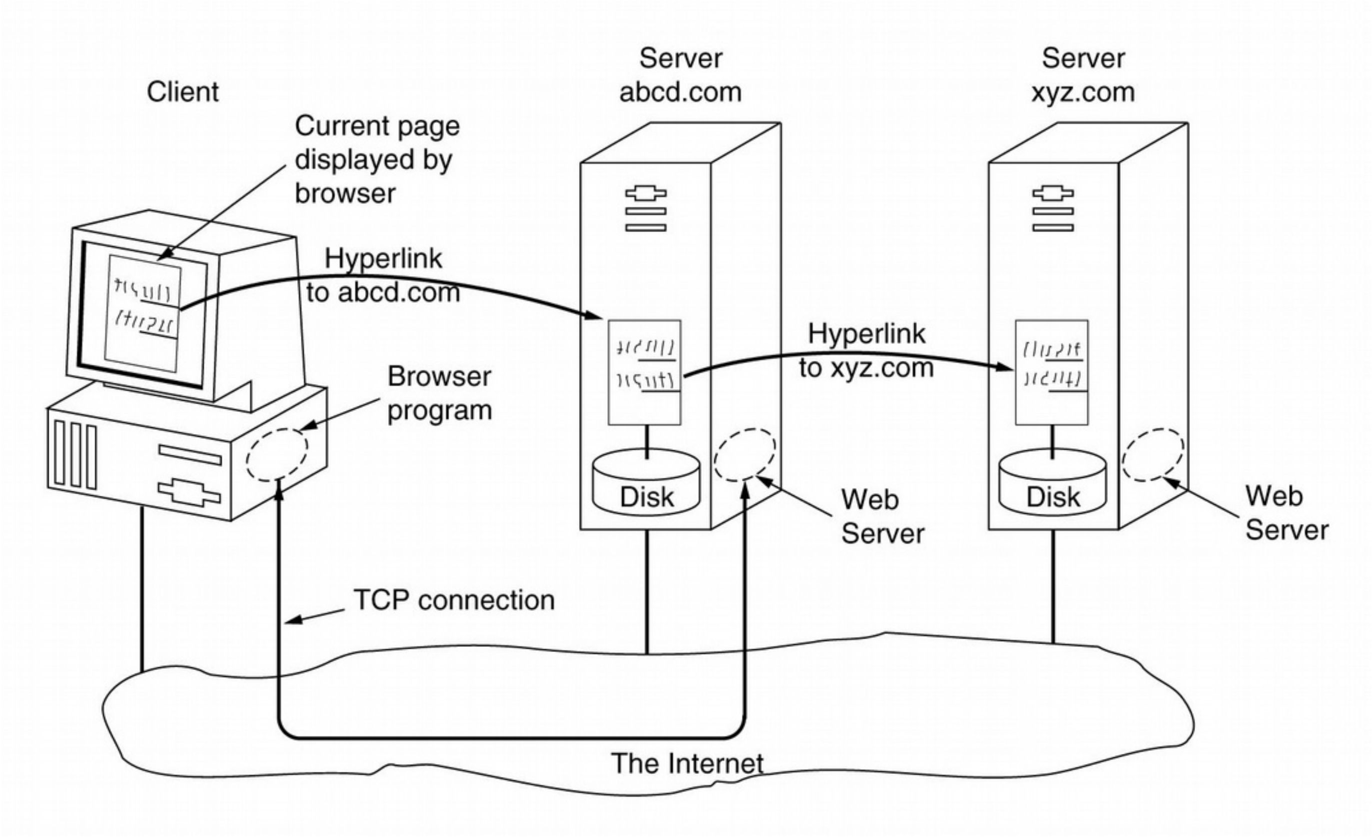
# Internet Message Access Protocol (IMAP)

- More sophisticated then POP3
- Client-server architecture
- TCP/143, TCP/993 (IMAPS)
- Presumes that messages remain preserved on the server
- Designated with the aim to limit data transfers between client and server
  - Suitable especially for mobile clients
  - Selective downloading of messages and message parts
  - Client may ask the server to search in the messages without passing them to the client
- Enhanced authentication

# World Wide Web

# Origin of the WWW and its Architecture

- CERN
- http://www.w3.org



# URI and URL

- Uniform Resource Identifier (URI)
  - String of characters used to identify or name a resource on the Internet
- Uniform Resource Locator (URL)
  - Specifies where an identified resource is available and the mechanism for retrieving it
  - protocol://user:password@machine:port/path

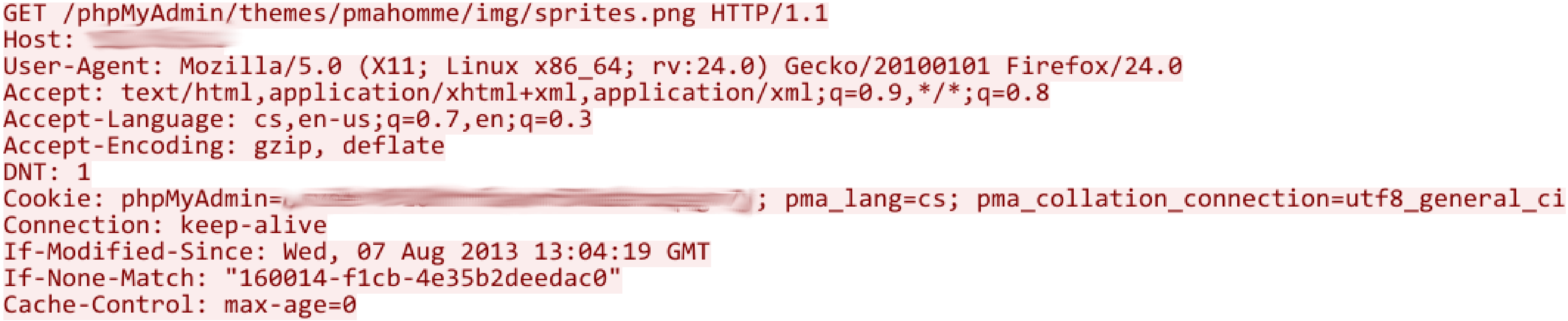
# Hypertext Transfer Protocol (HTTP)

- Client-server architecture, request-response protocol
  - TCP/80
- Uses URLs to identify a resource
- Utilizes MIME to describe media
  - In contrast to SMTP presumes binary connection
- Supports the access authorization
- Supports the page relocation

# Format of HTTP Request

command (GET, POST, PUT, …) PATH protocol
headers

(data-content of web form)



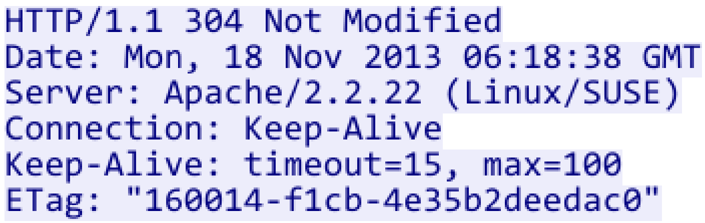
```
GET /phpMyAdmin/themes/pmahomme/img/sprites.png HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:24.0) Gecko/20100101 Firefox/24.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: cs,en-us;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: phpMyAdmin=[redacted]; pma_lang=cs; pma_collation_connection=utf8_general_ci
Connection: keep-alive
If-Modified-Since: Wed, 07 Aug 2013 13:04:19 GMT
If-None-Match: "160014-f1cb-4e35b2deedac0"
Cache-Control: max-age=0
```

# Format of HTTP Response

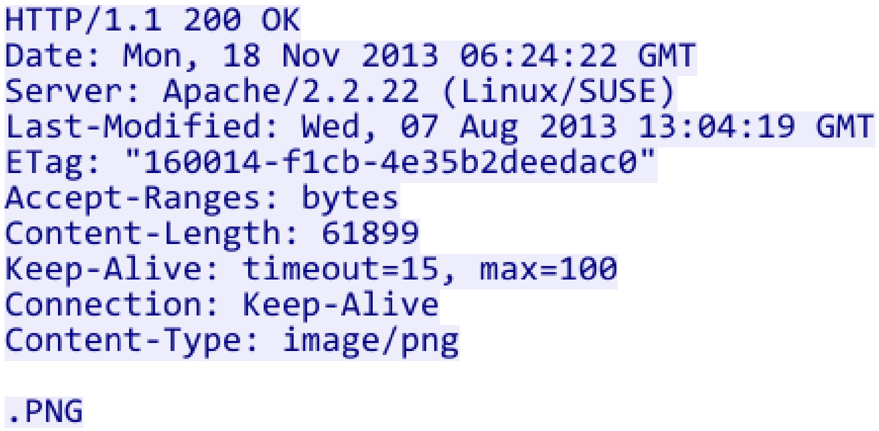
Protocol + resp. code + response message (HTTP/1.1 200 OK)

headers

(data-content of the Web page)



```
HTTP/1.1 304 Not Modified
Date: Mon, 18 Nov 2013 06:18:38 GMT
Server: Apache/2.2.22 (Linux/SUSE)
Connection: Keep-Alive
Keep-Alive: timeout=15, max=100
ETag: "160014-f1cb-4e35b2deedac0"
```



```
HTTP/1.1 200 OK
Date: Mon, 18 Nov 2013 06:24:22 GMT
Server: Apache/2.2.22 (Linux/SUSE)
Last-Modified: Wed, 07 Aug 2013 13:04:19 GMT
ETag: "160014-f1cb-4e35b2deedac0"
Accept-Ranges: bytes
Content-Length: 61899
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: image/png

.PNG
```

# Basic HTTP Commands (Methods)

- GET
  - Asks for the resource specified by URL
- HEAD
  - Asks for the header of the specified resource
- POST
  - Sends data (web form) to the server
- PUT
  - Stores the document to a file on the server
- DELETE
  - Deletes the resource
    - Requires authentication or not supported at all

# Additional HTTP Methods

- LINK, UNLINK
  - creates/deletes a link to the resource
    - Requires authentication or not supported at all
- OPTIONS
  - Determines a list of methods (commands) supported by the server
- TRACE
  - Allows to follow processing of the HTTP request
    - Useful for debugging of Internet applications
- PATCH (proposed RFC 5789, 2010)
  - Partially modifies specified resource

# HTTP reply codes

- 1xx – Informational
- 2xx – Successful
- 3xx – Redirection
  - 304 Not Modified
- 4xx – Client Error
  - 401 Unauthorized,
  - 403 Forbidden,
  - 404 Not Found,
  - …
- 5xx – Server Error
  - 500 Internal Server Error,
  - 502 Bad Gateway,
  - …

# Most useful HTTP Request Headers (1)

- Accept
  - specifies media that are accepted by the client (commonly *)
- Accept-Charset
  - character sets accepted by the client
- Accept-Encoding
  - encodings accepted by the client
- Accept-Language
  - languages accepted by the client
- Authorization
  - authentication credentials of the client

# Most useful HTTP Request Headers (2)

- If-Modified-Since
  - applies the request only if the document was modified from the given date
- Referrer
  - informs where the client obtained the URL of the requested resource
    - advertisements, statistics, searching for wrong references
- User-Agent
  - WWW browser name and version
- Upgrade
  - indicates protocol upgrade request (HTTP/2, web sockets, …)

# Useful HTTP Response Headers (1)

- Content-Encoding
  - encoding of the document provided
- Content-Language
  - language of the document provided
- Content-Length
  - length of the document
    - Important for HTTP 1.1 that does not terminate the connection after the document is sent to the client
- Content-Type
  - MIME type of the message body (e.g.: text/html)
  - May also contain ;charset=*character_set*
- MIME-Version

# Useful HTTP Response Headers (2)

- Date
  - date when the document was sent/generated
- Expires
  - expiration time of the document contents
- Last Modified
  - time of the last modification of the document

# Useful HTTP Response Headers (3)

- Location
  - automatic redirect, provides URL of the relocated document
- Retry After
  - a recommended time interval to repeat the request after Service Unavailable response
- Server
  - name and software of the HTTP server
- WWW-Authenticate
  - list of authentication supported by the server
- Refresh
  - asks the client to renew the contents of the document after a given time interval

# HTTP 1.0 (RFC 1945)

- Connection is initiated by client, terminated by server
- If the web page consists of multiple media files, every file is downloaded using a separate TCP connection

# HTTP 1.1 (RFC 2068)

- Client may ask the server to keep the TCP connection after finishing the requested document transfer
  - It is not necessary to establish a new TCP connection for every request
- Support for virtual hosts
  - Multiple logical servers listening on the same IP address
  - Host: request header or GET command has to contain the complete URL, including the virtual WWW server name
- Client may ask a specific part of a document
  - useful if a connection fails during transfer
  - utilized by download accelerators to overcome per-flow rate limitation
- Supports data compression during transfer

# HTTP 1.1 – Chunks

Transfer-Encoding: chunked

- If present in header, we do not know/specify content length, but send chunks of data instead
  - Both requests and responses
  - Often used together with Content-Encoding: gzip
- We include a line with hexadecimal length of each block before we send it
  - transfer ends if the length equals to 0.

```
a
ABCDEFGHI\n0
```

# HTTPS

- „secure" HTTP – encrypted channel
- HTTP over SSL/TLS
  - Server certificates
- Typically on port TCP/443
- Certificate authorities, DV and EV certificates

# HTTP/2

- Standard proposed in RFC 7540 in 2015
- Successor of Google SPDY protocol
- Interleaving of requests (not only pipelining)
  - Multiplexing multiple streams in a single connection, traffic prioritization
- Efficient coding for HTTP header fields
  - Header compression
- Push responses from server
- Client indicates the use of HTTP/2 by headers:
  - Upgrade: h2c
  - HTTP2-Settings: base64_settings
  - HTTP/1.1 servers ignore the Upgrade header
- Defined for HTTP & HTTPS, browsers mainly implement only for HTTPS (mandatory enc.)

# Cookies

- Support for stateful transactions
  - Helps the server to keep the identity of the client between requests
- Utilizes message header
  - Set-Cookie (from server to client)
  - Cookie (from client to server)
- Cookie structure:
  - Name, value
  - server, path
    - If a corresponding URL is accessed, the cookie is provided to the server
  - flag secure
    - send the cookie exclusively over HTTPS
  - Comment, max-age

# Protocols for Autoconfiguration of the Network Connection Parameters

# Bootstrap Protocol (BOOTP)

- Provides configuration of TCP/IP network connection according to station's MAC address
  - BOOTP server maintains a database of MAC-to-IP mappings
  - BOOTP clients ask the server to provide network connection parameters
  - IP address, subnet mask, default gateway, boot TFTP server and boot image name is provided
- Messages are propagated as UDP broadcasts
  - May be converted to unicast to allow traversal over routers

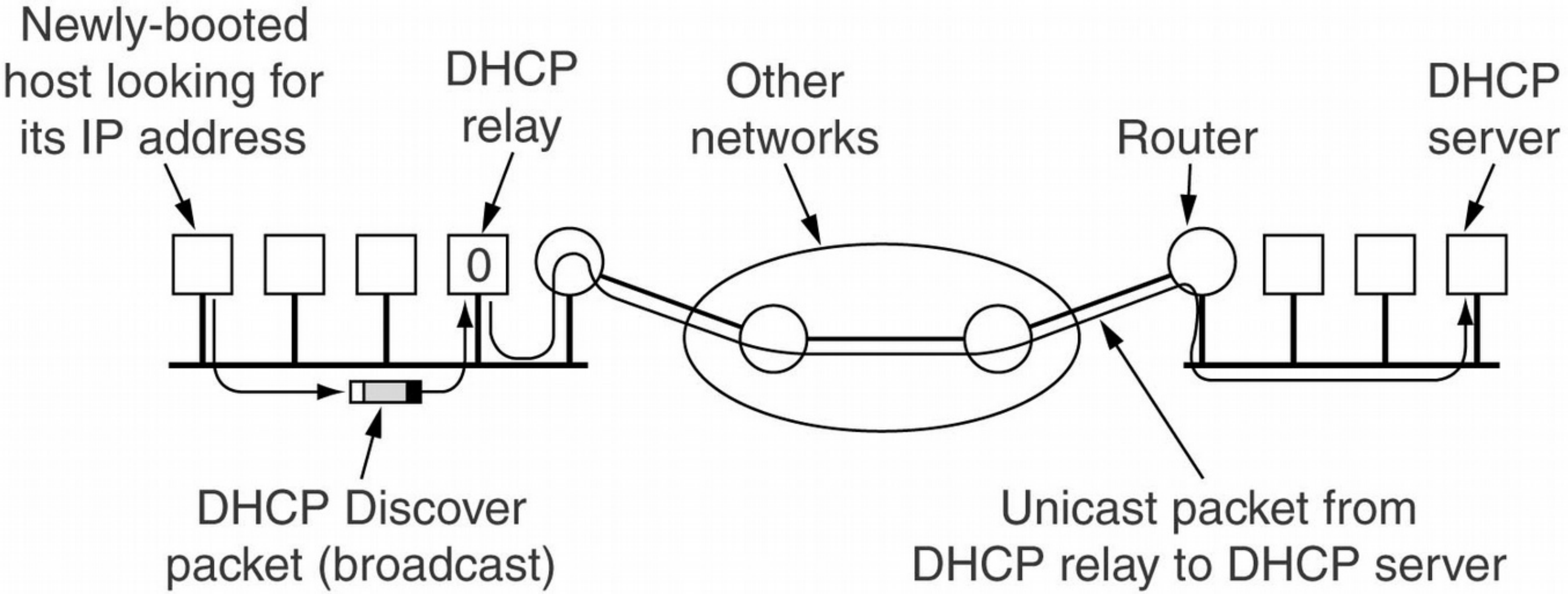
# Dynamic Host Configuration Protocol (DHCP)

- Dynamically provides network connection parameters using a pool of  available addresses
  - Address is leased for limited period
  - Lease has to be periodically renewed
- Addresses may be also provided according to pre-configured MAC-to-IP bindings
  - Compatible with BOOTP
- First DHCP message is sent as UDP broadcast
- In contrary to BOOTP, DHCP may also provide optional parameters
  - Support for defining of additional parameters

# DHCP Messages

- DHCP Discover – search for  DHCP server (broadcast)
- DHCP Offer - DHCP server offers parameters to lease
- DHCP Request – client asks the server to reserve previously offered parameters
- DHCP ACK – the server acknowledges the requested parameters

# Relaying of DHCP Requests


Newly-booted host looking for its IP address
DHCP relay
Other networks
Router
DHCP server
0
DHCP Discover packet (broadcast)
Unicast packet from DHCP relay to DHCP server