

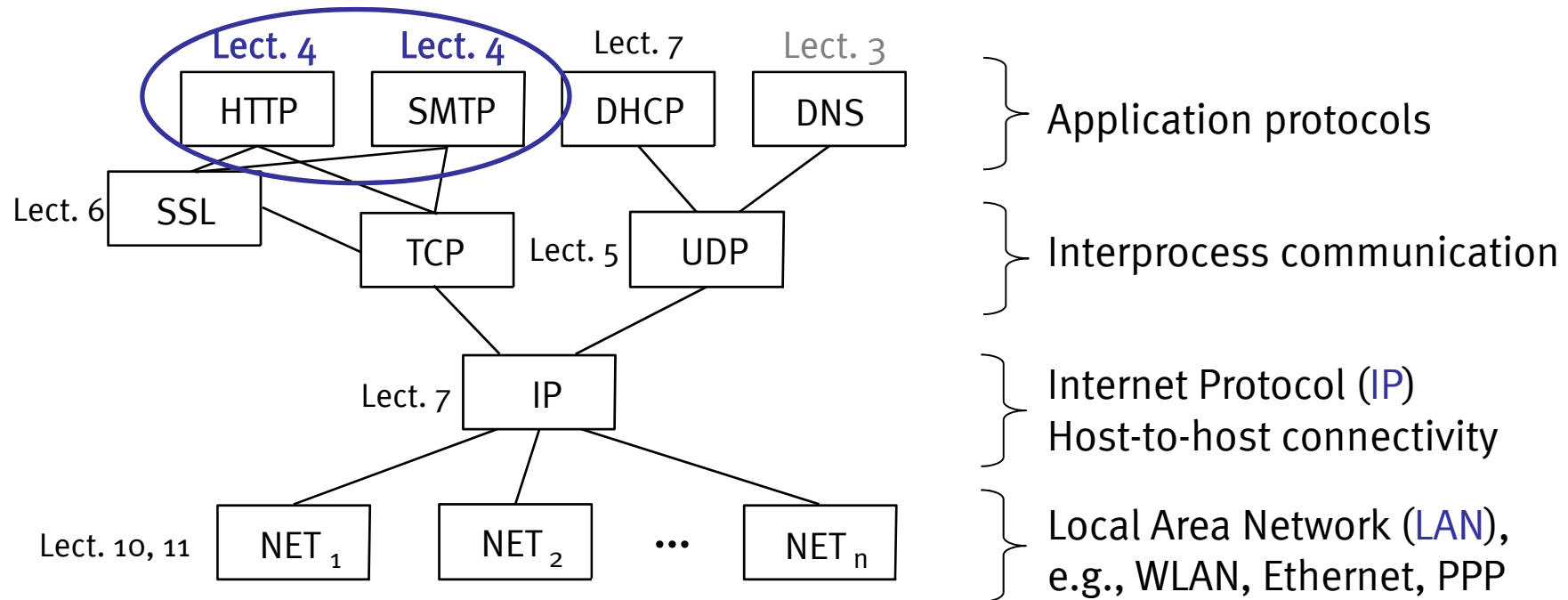


WESTFÄLISCHE
WILHELMS-UNIVERSITÄT
MÜNSTER

Computer Networks, Winter Term 2009/2010

Web and E-Mail

Recall: Internet Architecture



- What does your browser do when you enter a URI in the address bar?
- How does e-mail transfer work?

- Web
 - HTTP
 - Server State, Cookies
 - Caching
 - Proxies
- E-Mail

- 1945
 - Vannevar Bush: As we may think
 - <http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm>
 - Memex for information storage
 - Associative indexing (Hyperlinks)
- 1989
 - Tim Berners-Lee (CERN) publishes article on distributed hypertext system
 - “web of notes with links”
 - Initially for cooperation among physicists at CERN
- May 1991
 - Distributed information system based on HTML, HTTP, and client software at CERN
- August 1991
 - Availability of CERN files announced in <news:alt.hypertext>
 - <http://groups.google.com/group/alt.hypertext/msg/395f282a67a1916c>

- 1992

- NCSA Web Server available

- National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign

- 1993

- Mosaic browser created at NCSA

- 1994

- W3C (World Wide Web Consortium) founded by Tim Berners-Lee

- <http://www.w3c.org/>

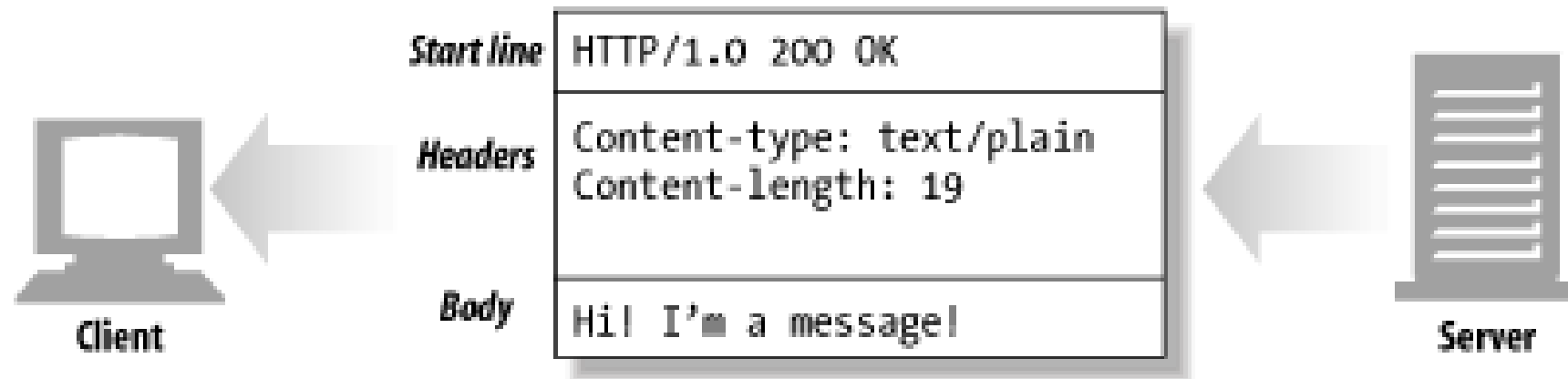
- Publication of technical reports and “recommendations”

- W₃C (HTML Spec)
 - The [World Wide Web \(Web\)](#) is a network of information resources.
- HTTP Spec
 - The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems.
- Distributed information system
 - Client-Server architecture
 - Web-Browser (client) sends HTTP requests to Web server
 - Based on
 - Internet (entire class)
 - URIs (previous lecture)
 - HTTP (now)
 - ((X)HTML, common knowledge)

- Hypertext Transfer Protocol
 - RFC 2616 (HTTP/1.1), <http://tools.ietf.org/html/rfc2616>
- Request/response protocol
 - Message format
 - Access methods
- Requires reliable transport protocol
 - Typically [TCP/IP](#), port [80](#)

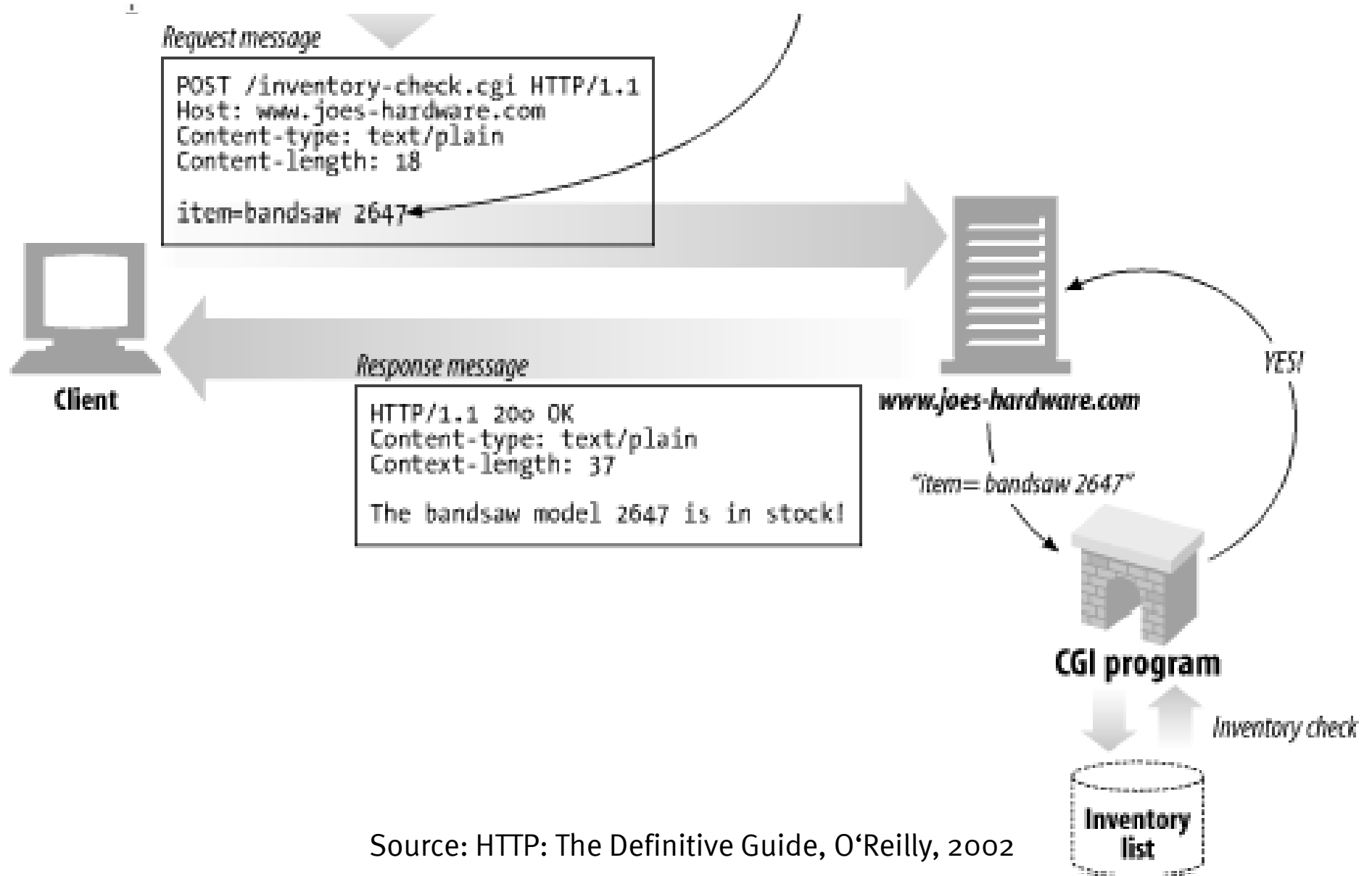
- Login to remote host (**plaintext** passwords)
 - Lecture 9:
 - telnet route-views.oregon-ix.net
- Also, establish **arbitrary TCP connections**, e.g.:
 - telnet www.google.de 80
(on Windows possibly followed by ctrl-+ or ctrl-),
set localecho [enter] [enter])
 - GET / HTTP/1.1 [enter]
 - Host: www.google.de [enter] [enter]
 - telnet wi.uni-muenster.de 25

- Requests and responses
- Generic message format of RFC 822, 1982 (822-→2822-→5322)
 - Originally for e-mail, extensions for binary data
 - **Headers**
 - In HTTP always a distinguished start-line
 - Then zero or more headers
 - **Empty line** (CRLF)
 - Optional message **body**



Source: HTTP: The Definitive Guide, O'Reilly, 2002

- Request = Request-Line ; Section 5.1
 - *((general-header ; Section 4.5
 - | request-header ; Section 5.3
 - | entity-header) CRLF) ; Section 7.1
 - CRLF
 - [message-body] ; Section 4.3
 - Request-Line = Method SP Request-URI SP HTTP-Version CRLF
 - Examples
 - GET /dbms/media/people/lechtenboerger/publications.html HTTP/1.1
Host: dbis-group.uni-muenster.de
 - OPTIONS * HTTP/1.1
Host: dbis-group
 - Note: Host header is mandatory for requests with HTTP/1.1
-
-



Source: HTTP: The Definitive Guide, O'Reilly, 2002

- **Case-sensitive** (capital letters)
- **OPTIONS** ; Section 9.2
 - Asks for server capabilities
- **GET** ; Section 9.3
 - Request for resource
- **HEAD** ; Section 9.4
 - Request information on resource
- **POST** ; Section 9.5
 - Transfers entity
 - Annotations, postings, forms, database extensions
- **PUT** ; Section 9.6
 - Creates new resource on server
- **DELETE** ; Section 9.7
 - Deletes resource from server
- **TRACE** ; Section 9.8
 - Tracing of messages through proxies

- GET under **conditions**
- Requires **request header**, e.g. (headers are **case-insensitive**)
 - If-Modified-Since
 - If-Match
 - If-None-Match
- Example
 - Request

```
GET /dbms/media/people/lechtenboerger/publications.html HTTP/1.1
Host: dbis-group
If-None-Match: "28c756-364c-ca9aa780"
```
 - Response

```
HTTP/1.1 304 Not Modified
Date: Tue, 27 Oct 2009 16:16:57 GMT
... (additional headers)
```

- 1xx: Informational - Request received, continuing process
 - 100: Continue – Client may continue with request body
- 2xx: Success - The action was successfully received, understood, and accepted
 - 200: OK
- 3xx: Redirection - Further action must be taken in order to complete the request
 - 302: Found
 - 304: Not Modified
- 4xx: Client Error - The request contains bad syntax or cannot be fulfilled
 - 403: Forbidden
 - 404: Not Found
- 5xx: Server Error - The server failed to fulfill an apparently valid request

- **Parallel connections**
 - Client may send several HTTP requests via separate TCP connections
- **Persistent connections**
 - TCP connection is reused for multiple HTTP requests
 - HTTP/1.0: Connection: Keep-Alive
 - HTTP/1.1: Persistence by default
- **Pipelined connections**
 - Client may send multiple requests over single TCP connection before receiving a response
 - In Firefox, pipelining must be enabled manually
 - about:config -> network.http.pipelining
 - Monitor with wireshark



- Web
 - HTTP
 - **Server State, Cookies**
 - Caching
 - Proxies
- E-Mail

- Stateless

- Server does not maintain client state information
- Advantages
 - State changes on server do not require client notifications
 - Recovery (restart after crash) “simple”
- E.g.: HTTP
 - Web server forgets client after request
 - No session

- Stateful

- Server maintains client state
- E.g., file server with table of pairs (Client, File) for caching
 - Keep track which client has current version
 - Performance improvement via locality
- Recovery requires to restore consistent state

- HTTP is stateless
 - Yet, Web applications often maintain client state
- E.g., personalized session after login
 - Virtual shopping cart
 - Shopping history, preferences
 - Exercises in xLx

- Need identifier to keep track of subsequent requests
- Two major variants
 - Session ID embedded in dynamically generated URIs
 - E.g., amazon.de
 - May **hinder** caching
 - URI does not identify resource any longer
 - Cookies
 - Again amazon.de
 - xLx, Google

- Computer jargon: Opaque identifiers sent back and forth
- RFC 2965: HTTP State Management Mechanism
- Idea
 - Client stores information sent by server
 - Client sends this information with subsequent requests
- Details
 - Cookie is named byte string
 - Server transfers cookie in Set-Cookie(2) header in response
 - Set-Cookie: Version 0/Netscape
 - Set-Cookie2: Version 1/RFC2965
 - (JavaScript may create cookie at client)
 - Client sends cookie in Cookie header in requests

- Cookies have name, value, optional attributes/flags
- Domain
 - DNS domain of servers to which the cookie should be sent
- Max-Age
 - Lifetime of cookie in seconds
- Discard
 - Session cookies which should be deleted when user agent exits
 - (No expires with Netscape cookies)
- Secure
 - Should only be sent via HTTPS
- HttpOnly (Microsoft extension)
 - Script access restricted; XSS counter-measure

- Web
 - HTTP
 - Server State, Cookies
 - **Caching**
 - Proxies
- E-Mail

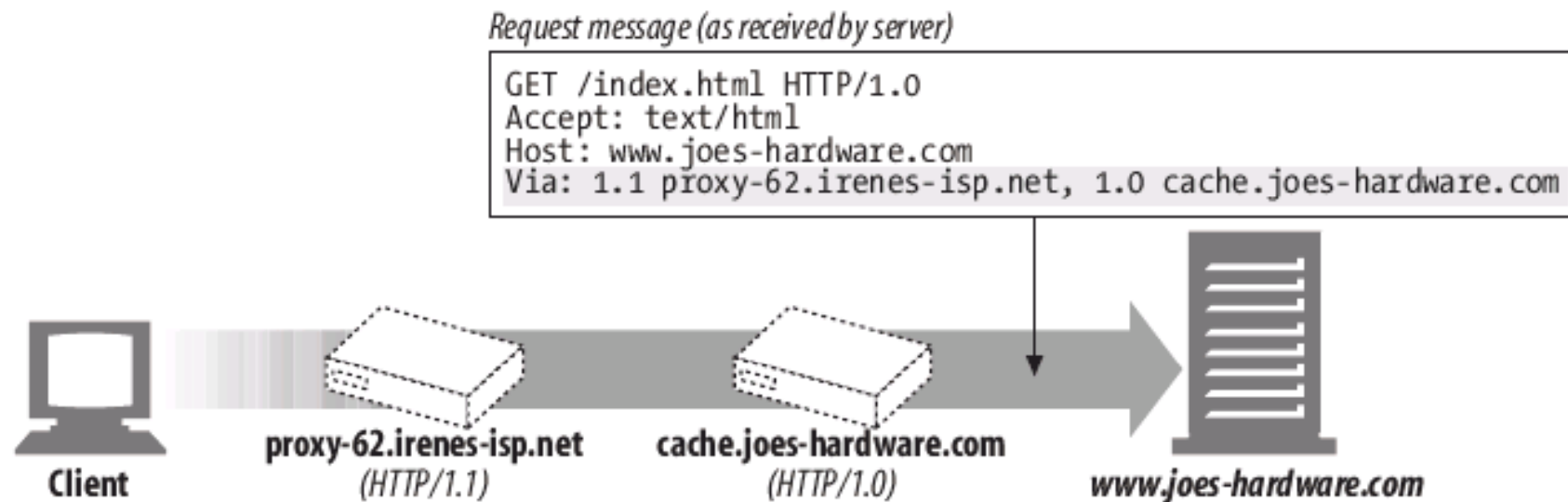
- Recall caching assumptions
 - URI identifies resource, stability, client-independence
- Semantic transparency
 - Caching is not visible to users
 - Response from cache is equivalent to hypothetical one from server
- Two mechanisms
 - Expiration
 - Server may indicate expiration date in Expires- or Cache-Control header
 - Validation
 - After expiration date, cache must check whether resource still usable
 - May return new expiration date
 - Conditional GET
 - Slow hit



- Complex rules, lots of details
- Server may limit caching
 - no-store, no-cache, must-revalidate
- Client may
 - enforce validation
 - no-cache
 - forbid caching
 - no-store

- Web
 - HTTP
 - Server State, Cookies
 - Caching
 - Proxies
- E-Mail

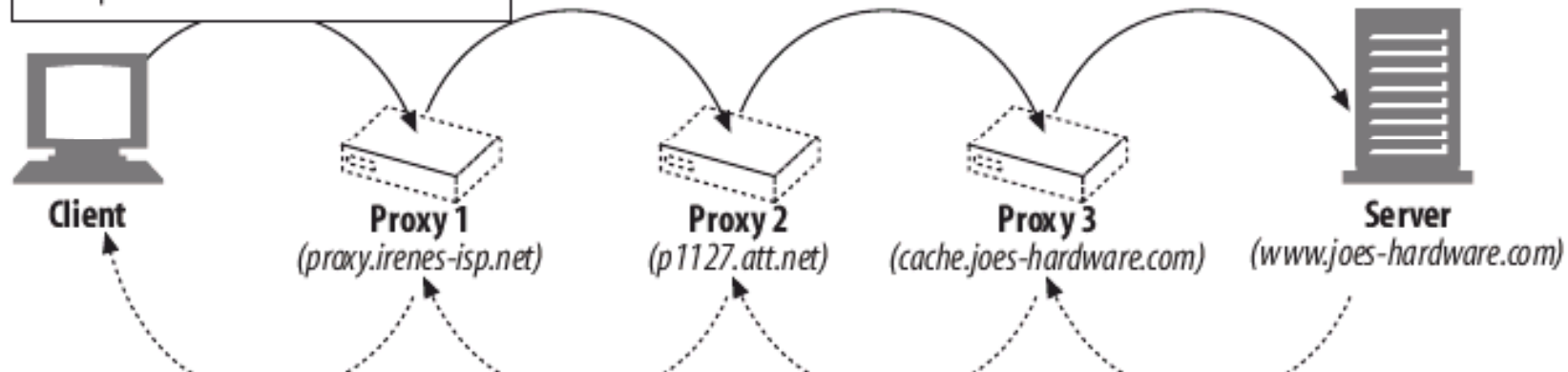
- Web proxy server is intermediary between client and server
 - Acts as server to client
 - Acts as client to server



Source: HTTP: The Definitive Guide, O'Reilly, 2002

TRACE request

```
TRACE /index.html HTTP/1.1
Host: www.joes-hardware.com
Accept: text/html
```



```
HTTP/1.1 200 OK
Content-Type: message/http
Content-Length: 269
Via: 1.1 cache.joes-hardware.com, 1.1 p1127.att.net, 1.1 proxy.irenes-isp.net
```

```
TRACE /index.html HTTP/1.1
Host: www.joes-hardware.com
Accept: text/html
Via: 1.1 proxy.irenes-isp.net, 1.1 p1127.att.net, 1.1 cache.joes-hardware.com
X-Magic-CDN-Thingy: 134-AF-0003
Cookie: access-isp="Irene's ISP, California"
Client-ip: 209.134.49.32
```

Received request

TRACE response

Source: HTTP: The Definitive Guide, O'Reilly, 2002

- Cache
- Firewall/Content filter
- Anonymizer
 - Tor, <https://www.torproject.org/>
 - JonDonym, <https://www.jondos.de/en/>
- Debugging tool
- Surrogate/Reverse proxy, Content Delivery Network (CDN)
 - Intercepts inbound messages, e.g.:
 - Load balancing
 - Geographical diversity (reduced latency, increased availability)

- Modification of
 - Web client
 - Client routes HTTP messages to proxy
 - Network
 - Switches/router redirect HTTP messages
 - DNS
 - DNS serves IP address of surrogate
 - Web-System
 - Redirect of request to proxy

- nslookup www.microsoft.com
 - www.microsoft.com canonical name = toggle.www.ms.akadns.net.
 - toggle.www.ms.akadns.net canonical name = g.www.ms.akadns.net.
 - g.www.ms.akadns.net canonical name = lb1.www.ms.akadns.net.
 - Following IP addresses for lb1.www.ms.akadns.net:
 - 207.46.19.254, 207.46.192.254, 207.46.193.254, 65.55.12.249, 65.55.21.250, 207.46.19.190
 - Subsequent query:
 - 207.46.192.254, 207.46.193.254, 65.55.12.249, 65.55.21.250, 207.46.19.190, 207.46.19.254
 - Subsequent query:
 - 207.46.193.254, 65.55.12.249, 65.55.21.250, 207.46.19.190, 207.46.19.254, 207.46.192.254
 - ...

- Web
 - HTTP
 - Server State, Cookies
 - Caching
 - Proxies
- E-Mail

- Among oldest Internet applications
- Message format
 - RFC 822 seen above
 - Extended with Multipurpose Internet Mail Extensions (MIME)
 - Content-Type (type of data contained in message)
 - Content-Transfer-Encoding (how data in message body is encoded)
- Plaintext messages
 - E-mail is like **postcard**, written with **erasable** pencil
 - Neither confidentiality nor integrity
 - Use **GnuPG** if you don't like this
 - (SSL/TLS wrong approach, recall dissemination pattern)
- Terminology
 - Mail User Agent (MUA): Your mail reader (browser, Thunderbird, Emacs)
 - Mail Transfer Agent (MTA): Mail server/daemon (sendmail, exim, postfix)

- Simple Mail Transfer Protocol, 1982 (SMTP, RFC 821-→2821-→5321)
 - Outgoing messages, MTA-to-MTA
 - **Plaintext**
 - Usually, **TCP/IP** via **port 25**
- (MTA-to-MUA via **POPS**, **IMAPS**, **HTTPS**)

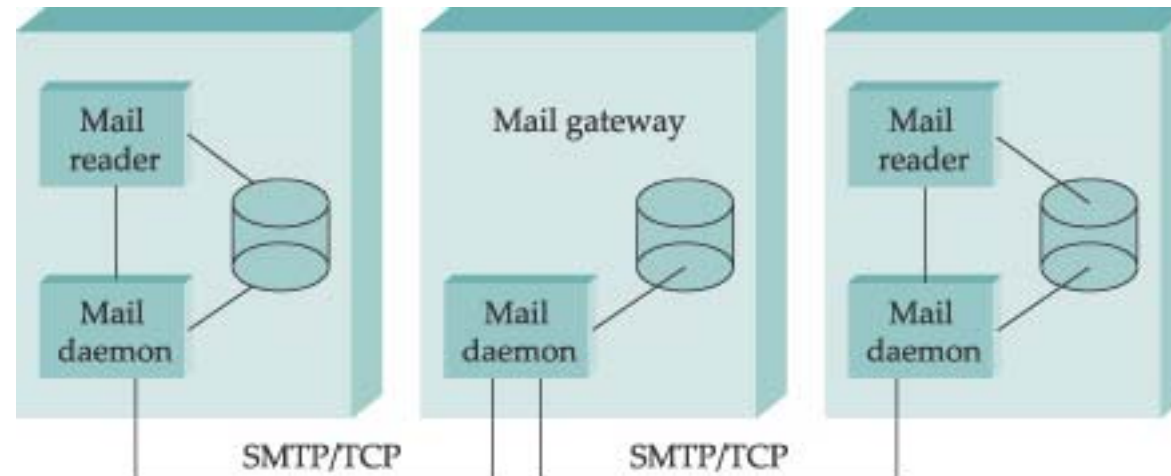


Figure is part of Powerpoint slides for Peterson and Davie:
Computer Networks (4th ed.). © 2007, Elsevier Inc.

telnet wi 25

Trying 128.176.159.139...

Connected to wi.uni-muenster.de.

Escape character is '^'].

220 wi-vm700.wi1.uni-muenster.de Microsoft ESMTP MAIL Service ready at Tue, 27 Oct 2009 11:22:11 +0100

HELO mouse.nix

250 wi-vm700.wi1.uni-muenster.de Hello [128.176.159.107]

MAIL From: micky@mouse.nix

250 2.1.0 Sender OK

RCPT To: lechten@wi.uni-muenster.de

250 2.1.5 Recipient OK

DATA

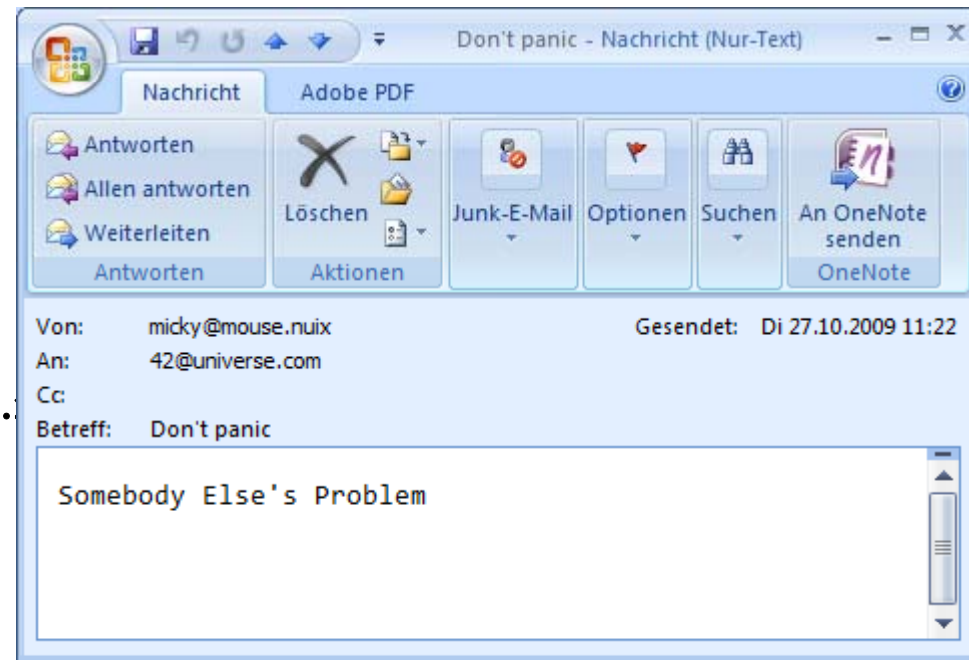
354 Start mail input; end with <CRLF>.<CRLF>

Received: from mx1.disney.com ([192.195.66.100])
Tue, 27 Oct 2009 11:19:17 +0100

To: 42@universe.com

From: micky@mouse.nuix

Subject: Don't panic



Somebody Else's Problem

250 2.6.0 <b13a2a36-f56b-43ec-ad81-41ec44190e6a@wi-vm700.wi1.uni-muenster.de> Queued mail for delivery

Microsoft Mail Internet Headers Version 2.0

Received: from wi-vm700.wi1.uni-muenster.de ([128.176.158.92]) by wi-vmail2005.wi1.uni-muenster.de with Microsoft SMTPSVC(6.0.3790.3959); Tue, 27 Oct 2009 11:22:35 +0100

Received: from **mouse.nix** (128.176.159.107) by wi-vm700.wi1.uni-muenster.de (128.176.159.139) with Microsoft SMTP Server id 8.1.375.2; Tue, 27 Oct 2009 11:22:28 +0100

Received: from mx1.disney.com ([192.195.66.20]) by smtp.mouse.nix Super Duper SMTP Server; Tue, 27 Oct 2009 11:19:17 +0100

To: 42@universe.com

From: <micky@mouse.nuix>

Subject: Don't panic

MIME-Version: 1.0

Content-Type: text/plain

Message-ID: <b13a2a36-f56b-43ec-ad81-41ec44190e6a@wi-vm700.wi1.uni-muenster.de>

Return-Path: **micky@mouse.nix**

Date: Tue, 27 Oct 2009 11:22:28 +0100

X-OriginalArrivalTime: 27 Oct 2009 10:22:35.0473 (UTC) FILETIME=[66C35410:01CA56EF]

1. Don't ask for data that you can't (or don't want to) verify!
2. Don't hide reliable information

- Web browsers and servers talk HTTP
 - Simple message format
 - Stateless request/response protocol
 - State via cookies
 - Different connection types
 - Caching for performance
- E-Mail transferred via SMTP

- HTTP used for various applications
 - Web services
 - SOAP messages
 - Ad-hoc request/reply protocols
- REST
 - Representational State Transfer
 - Software architecture for distributed hypermedia systems
 - Generalization of Web
 - Defining constraints
 - Client/Server
 - Stateless
 - Cacheable
 - Uniform interface, may use: URIs, MIME types, HTTP methods
 - Layered System
 - (Code on demand)

- Perform simple HTTP requests via telnet
- Discuss use cases for persistent and pipelined connections
 - Identify their use in packet captures
- Explain the concept of “stateless servers”
- Explain constraints and advantages of caching
- Interpret E-Mail headers
- Discuss alternatives to and weaknesses of e-mail security established by secure channels between MUA and MTA