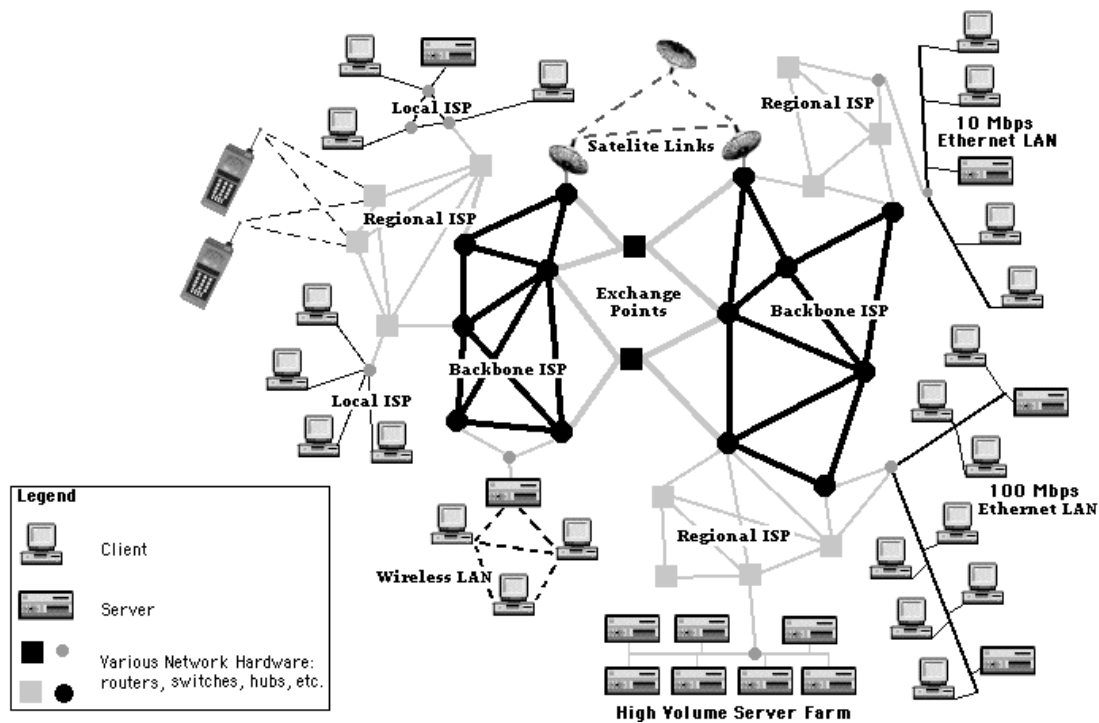


WWW Technologies

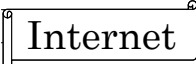

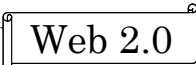
Eleni Stroulia

The Internet



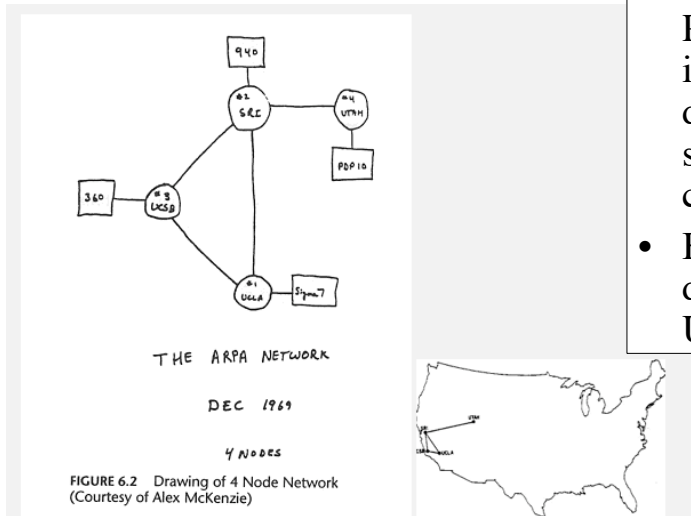
How did we get here?

A Short History of Technology

-  Internet
 - The world as a “global village”
-  WWW
 - A standard hypermedia language for describing content
 - A standard protocol for accessing content
-  Web 2.0
 - Anything can be “content”
 - Everybody is a provider and consumer
 - A business model emerges for the on-line society

ARPANET 1969

- Few large computers with different OS
- An Interface Message Processor (IMP) was needed, in front of each computer, to deliver packet switching services between the computers.
- BBN won a contract to develop a network among UCLA, SRI, UCSB, Utah



<http://www.w3c.rl.ac.uk/primers/history/origins.htm>

July 1977: The Internet is born!

- The first a large scale demonstration of internetworking:
- A message was sent from a van on the San Francisco highway
 - by radio to ARPANET,
 - by satellite to Norway,
 - by a land line to UCL,
 - by satellite back to the USA, and
 - by ARPANET to Los Angeles.
- A round trip of 94000 miles across several networks and technologies to deliver a message 800 miles away!

Look at <http://www.opte.org/maps/> for Internet maps



Vannevar Bush: Memex

- “As We May Think” (Atlantic Monthly 1945)
 - the human mind works by association
 - the memex machine should imitate it with “associative trails” running through pieces of stored information, strengthened as they are traversed

10-07

based on notes on the <http://www.cknuckles.com/webapps/companion>

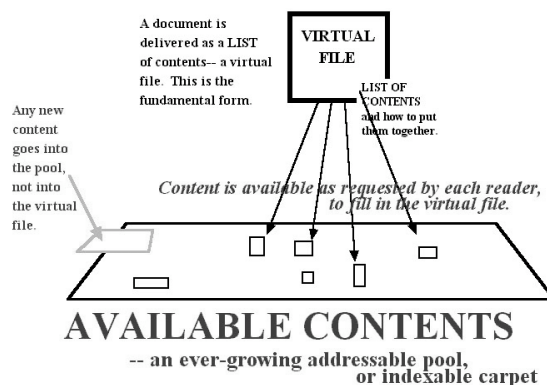
8



Ted Nelson: Xanadu

AN AUTHOR-BASED, LITERARY AND CULTURAL DESIGN The Xanadu Document Model

– built on the assumption of perpetual change and re-use



<http://xanadu.com/xuTheModel/>

- In 1965, he defined hypertext
 - A body of written or pictorial material
 - interconnected in a complex way (not appropriate for paper)
 - that may contain
 - summaries or maps of its contents and their interrelations
 - annotations, additions and footnotes from readers

10-07

based on notes on the <http://www.cknuckles.com/webapps/companion>

9



Tim Berners Lee: (1990)

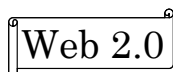
The first Web server + browser

- On Christmas Day 1990, a working prototype browser and server running on a NeXT computer went live
 - Clicking on a link always opened a new window.
 - It was just as easy to edit pages as to view them.
- A paper submitted to Hypertext '91 was rejected but there was a demonstration of the system, which did not impress the hypertext community:-(
 - The system was too simple
 - Links sometimes broke

10-07

based on notes on the <http://www.cknuckles.com/webapps/companion>

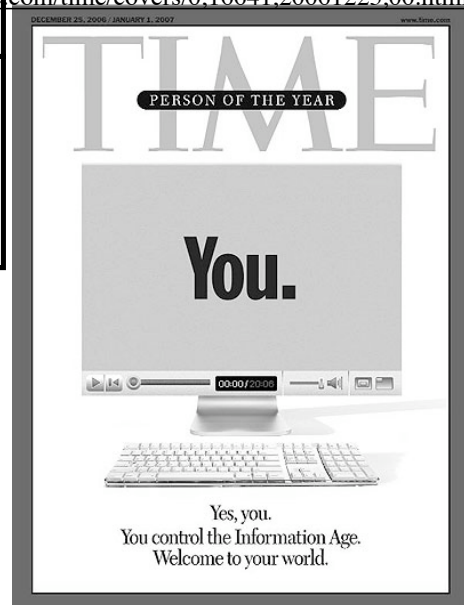
11



Web 1.0 vs Web 2.0

• Companies	• Communities
• Reading	• Writing http://www.time.com/time/covers/0,16641,20061225,00.html
• Britannica	• Wikipedia
• Client-Server	• Peer to peer (P2P)
• HTML	• XML
• Home pages	• Wikis and Blogs
• Portals	• RSS
• Taxonomy	• Tags
• Wired	• Wireless
• Owning	• Sharing
• IPOs	• Trade sales
• Netscape	• Google
• Screen scraping	• APIs
• Dialup	• Broadband
• Hardware costs	• Bandwidth costs

<http://joedrumgoole.com/blog/2006/05/29/web-20-vs-web-10/>



10-07

based on notes on the <http://www.cknuckles.com/webapps/companion>

12

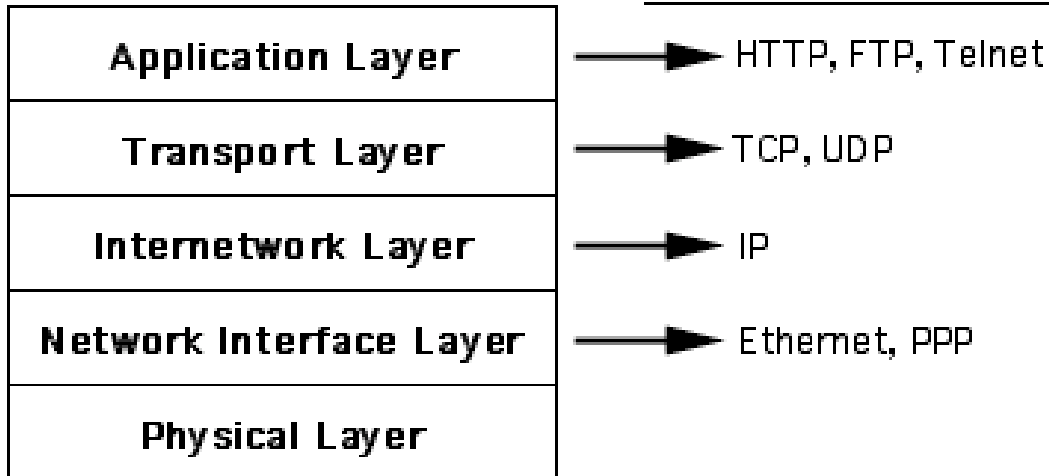
A bit more technically...

What is there on the Internet?

- Users use “clients” (usually browsers) to access information and services from web sites and web applications hosted on Servers
- Sometimes, popular web sites and applications run on server farms to distribute the load on the huge number of hits
- Servers have fixed IP addresses
- A server may reside on the backbone (large scale segments of the Internet operated by major organizations) or on the periphery of the Internet, where smaller ISPs reside
- Organizations operate LANs (using Ethernet) on the periphery of the Internet

The 5-layer model

The most common protocols
that operate on each layer



Protocol: a formal description of
formats and rules that messages between
computers must follow to communicate

The physical layer

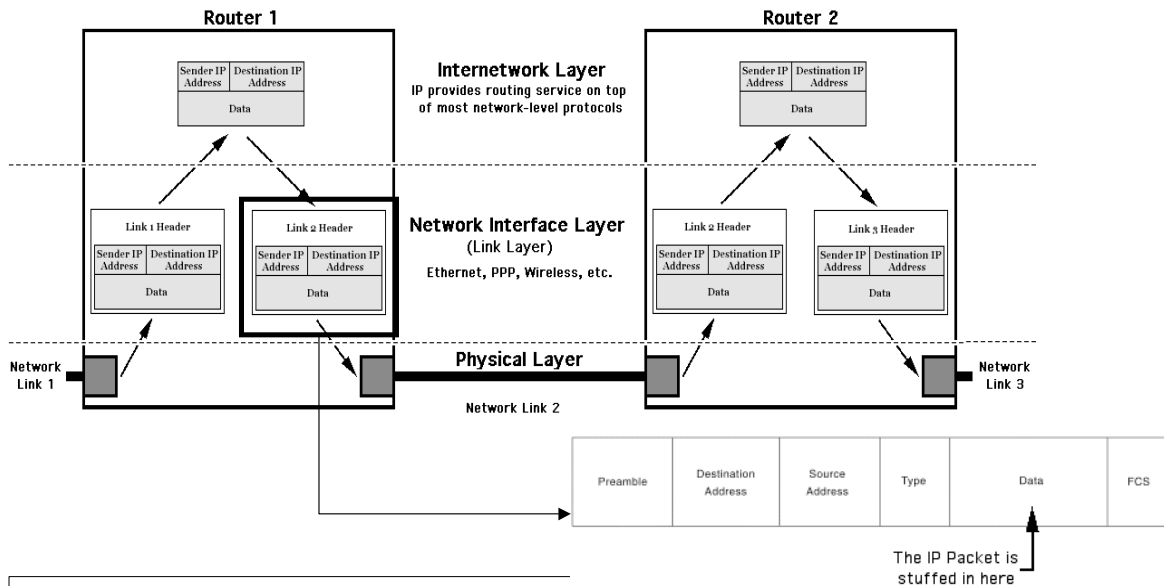
- Bits move across a shared medium between two pieces of hardware
- Depending on the hardware medium, different rules govern the transmission of energy
 - Electron streams through copper cable
 - Electromagnetic radiation frequencies through fiber optic
 - ...

The network-interface (data link) layer

- Bits are organized, in a hardware-independent manner, in data packages
- Ethernet (<http://pclt.cis.yale.edu/pclt/COMM/ETHER.HTM>)
 - Each station has a unique MAC addresses and transmits “packets” of data
 - “Multiple Access”: every station is connected to a single copper wire
 - “Carrier Sense”: before transmitting data, a station checks the wire to see if any other station is already sending transmitting
 - “Collision Detection”: Two stations can begin to send data at the same time, and their signals will "collide" nanoseconds later. When such a collision occurs, the two stations stop transmitting, "back off", and try again later after a randomly chosen delay period.

The internetwork layer (**I**nternet**w**orking **P**rotocol)

- Routers direct packets among different networks based upon IP addresses.
 - Flow Control: each router checks what its neighbors are doing and tries to pick good (and not congested) routers
 - Time To Live (TTL): each packet is allowed to be forwarded a max number of times
- IP is connectionless and does not guarantee delivery



Preamble: Warns receiver of incoming frame.

Destination/Source Addresses: Hardware MAC addresses.

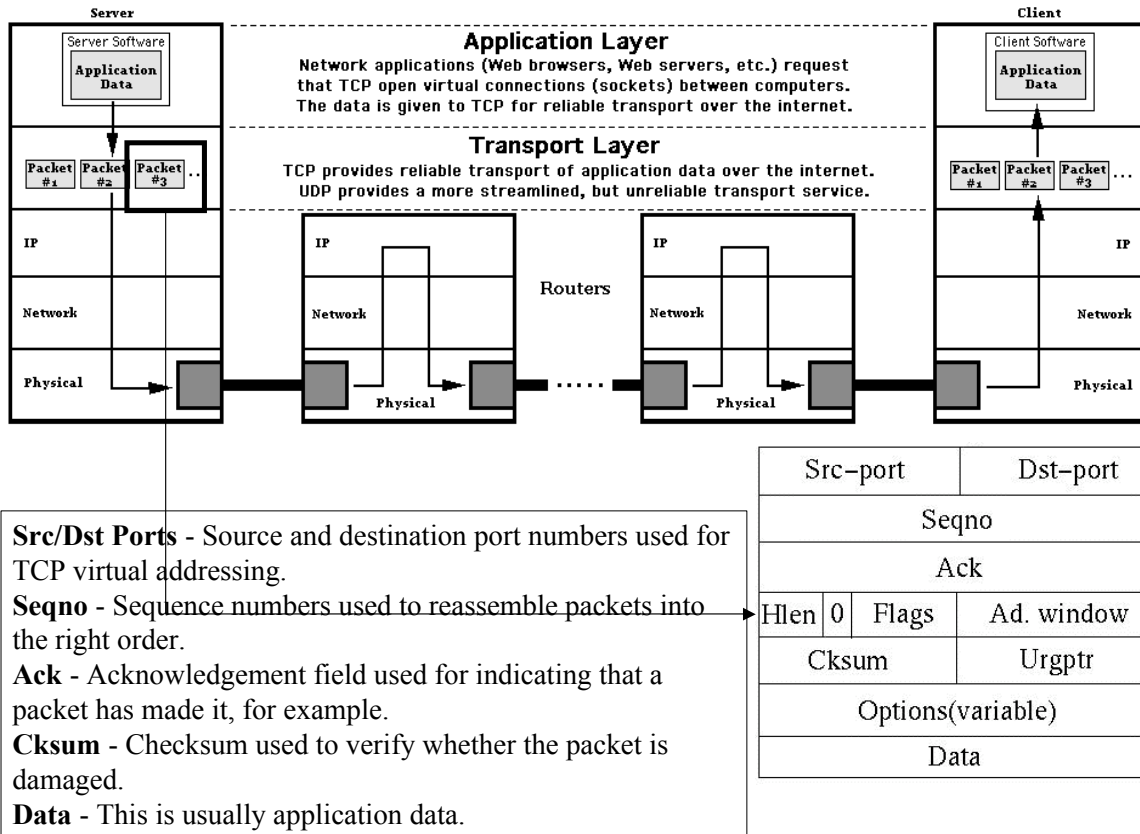
Type: The type specifies the upper-layer protocol to receive the data, i.e., IP

Data Payload: After physical-layer and link-layer processing is complete, the data contained in the frame is sent to the upper-layer protocol specified in the Type field.

Frame Check Sequence (FCS) - A 4-byte error check sequence which is created by the sending device and recalculated by the receiving device to check for damaged frames.

The transport layer

- **Transmission Control Protocol (TCP):**
 - Establishes sockets between client and server
 - Divides data into packets, stamps them with packet number and checksum
 - Resends upon lack-of-ack or checksum-failure response
 - Acknowledges arrival on the other end, puts packets back together in order
- **User Datagram Protocol (UDP):**
 - No guarantee of delivery, often single packet transmission, used for Domain Name Service.
- **Why is it good to split data in packets?**



10-07

based on notes on the <http://www.cknuckles.com/webapps/> companion

21

The application layer

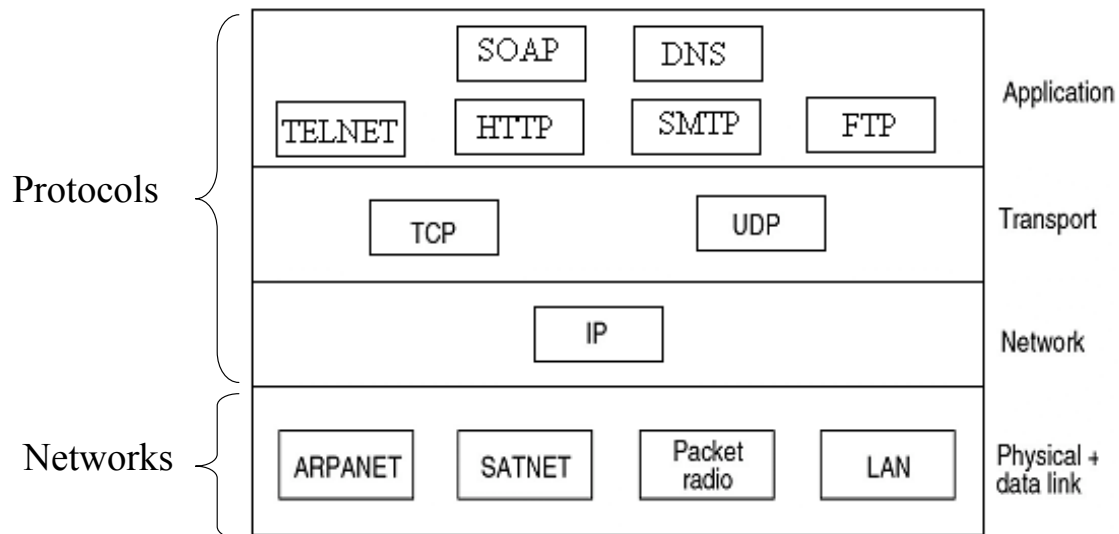
- telnet
- ftp
- SMTP, POP, IMAP, Web-based e-mail
- HTTP

10-07

based on notes on the <http://www.cknuckles.com/webapps/> companion

22

The Internet Protocol Stack

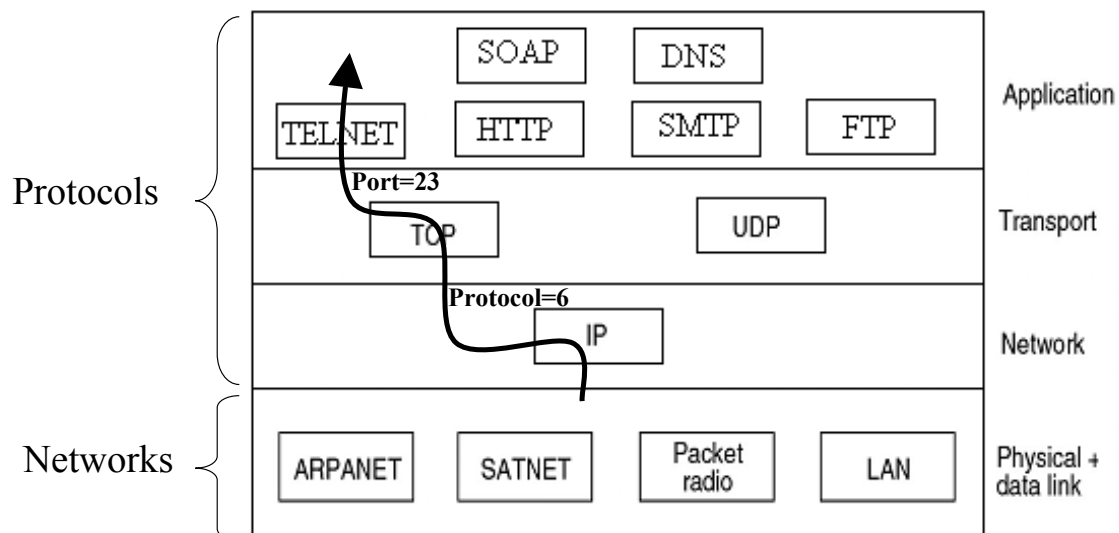


10-07

based on notes on the <http://www.cknuckles.com/webapps/> companion

23

Selecting an Application Layer Protocol



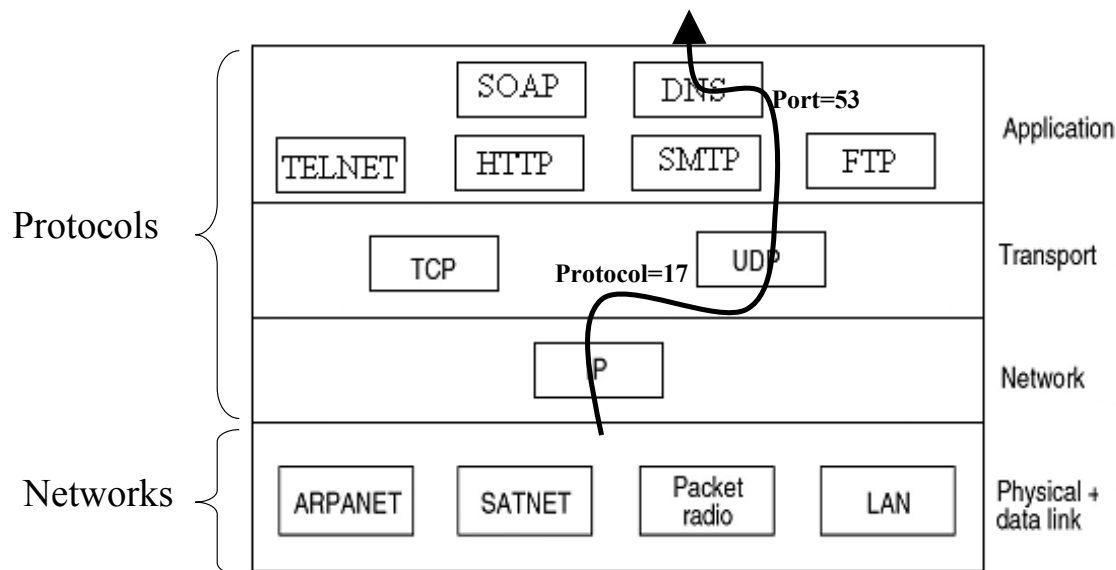
In Unix systems see /etc/protocols for Protocol numbers /etc/services for Port numbers.

10-07

based on notes on the <http://www.cknuckles.com/webapps/> companion

24

Selecting an Application Layer Protocol



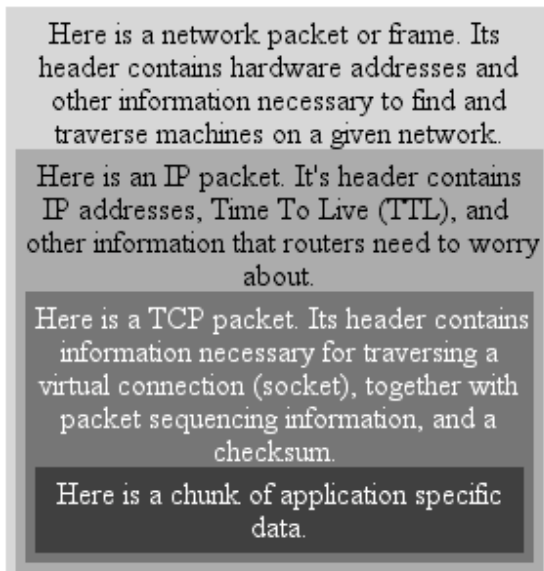
In Unix systems see /etc/protocols for Protocol numbers /etc/services for Port numbers.

10-07

based on notes on the <http://www.cknuckles.com/webapps/companion>

25

The structure of packets



- A layer treats the information from the layers above as packet data. Of course, the layers do "talk" to each other to some extent to get the correct information into the headers.
- At the network level, even the internet addressing information is obscured. the next level down is electrons (or light) flowing.
- As a packet goes back up the stack, headers are stripped and discarded to re-expose the header pertinent to that level.
- As a packet jumps around to different routers, the network level header may be repeatedly stripped and recreated as necessary to negotiate different network links.
- By the time the packet makes it to the destination application, all the extra baggage is removed and the

<http://www.cknuckles.com/webapps/chap01/extras/encapsulation.html>

10-07

based on notes on the <http://www.cknuckles.com/webapps/companion>

26

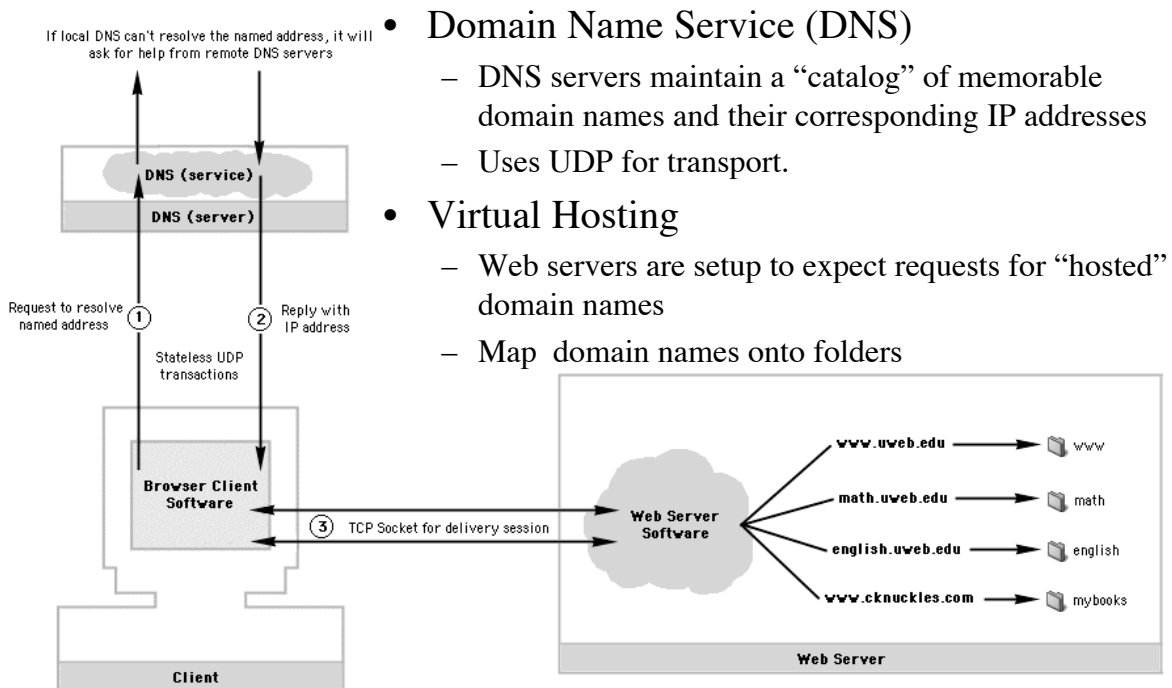
At the Application Layer...

10-07

based on notes on the <http://www.cknuckles.com/webapps/ companion>

27

DNS and Virtual Hosting



10-07

based on notes on the <http://www.cknuckles.com/webapps/ companion>

28

The structure of URLs

`protocol://named.address.com/directory/path/to/document.html`



- How: The application layer protocol. Example: Browsers and Web servers use http (hypertext transfer protocol)
 - Where: The named address of the resource -- Translated to IP address for internet travel, then used to locate the virtual space (folder) on the Web server.
 - What: Path to specific resource in that folder.
-
- There may be a special port, qualifying the “where”, if the web server is not listening to the default port 80
 - The file extension implies its type
 - There may also be a query string when the target recipient is a program

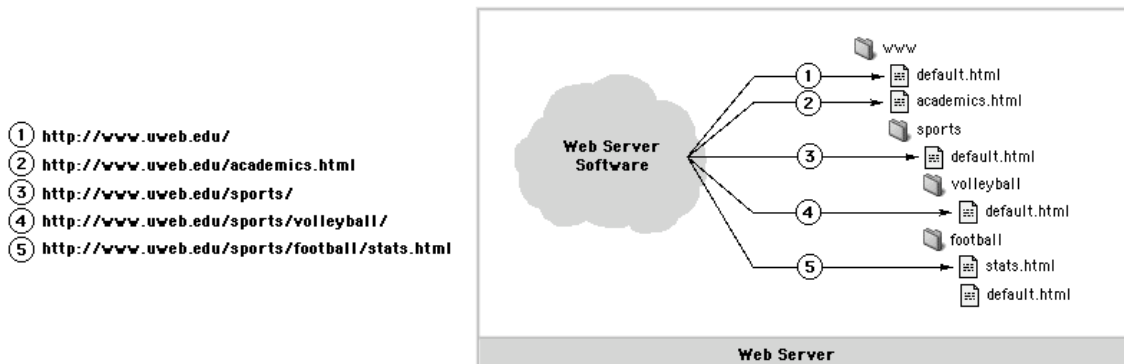
10-07

based on notes on the <http://www.cknuckles.com/webapps/> companion

29

The structure of URLs

- The what part of a URL identifies a particular resource within the folder assigned to the domain name (where part of URL).
- The request for a directory loads a default file in that directory. If there is no default file, then the directory contents may be listed.
- The request for a file loads that file.



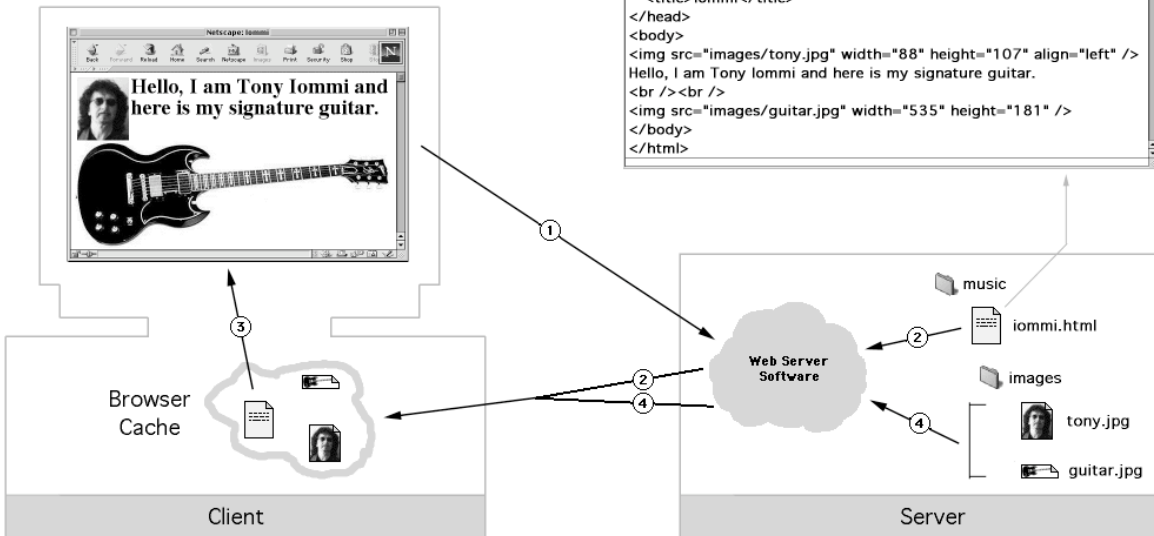
10-07

based on notes on the <http://www.cknuckles.com/webapps/> companion

30

A basic HTTP transaction.

- ① URL request `http://music.uweb.edu/iommi.html`
- ② A copy of the HTML document is transferred into the browser's cache.
- ③ The browser starts parsing the HTML code and asks the Web server software to send the two image files. (The "kept alive" socket is still being used.)
- ④ The image files are transferred into the cache, completing the Web page.



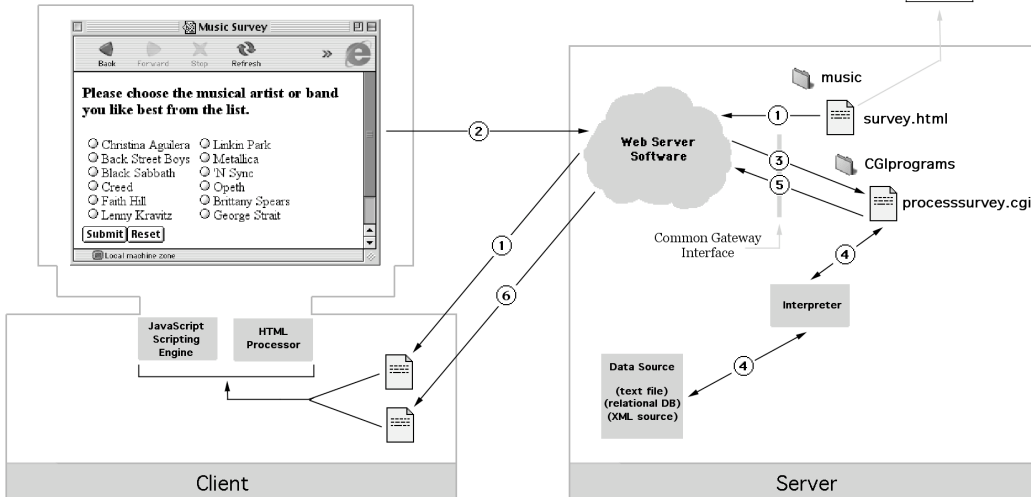
10-07

based on notes on the <http://www.cknuckles.com/webapps/> companion

31

A more complex HTTP transaction

- ① The front end of the Web application is passed to the client as an HTML file that contains some JavaScript code. The JavaScript validates the form data on the client.
- ② The user submits the form to a program named `processsurvey.cgi` on the Web server. Fundamentally, this is an HTTP transaction between the browser and the server software, but the URL points to an executable gateway program.
- ③ The server software executes the gateway program and passes it the form data. This is the first part of the gateway interface.
- ④ The gateway program processes the form data and interacts with a back-end data source.
- ⑤ The gateway program passes its output back to the Web server software formatted as an HTML document. This completes the gateway interface.
- ⑥ The server software passes the HTML document to the client, thereby completing the HTTP transaction.



10-07

based on notes on the <http://www.cknuckles.com/webapps/> companion

32

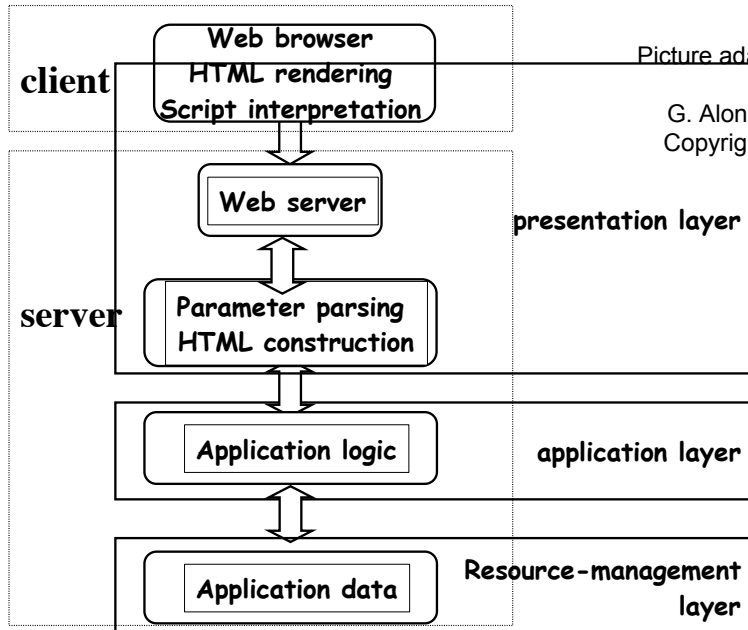
What happens on the client's browser

- Rendering HTML
- Frequent functions for Javascript:
 - Browser sniffing
 - Dynamic HTML (pop ups, animations, recoloring of HTML elements)
 - Client-side form validation

What happens on the server's back-end

- The web server passes the request to an executable program
 - Locates the program (there may be special directories set-up for that purpose)
 - Passes the query-string parameters to the target

The Software Architecture of Web-based Information systems



Picture adapted from "Web Services: Concepts, Architecture and Applications"
G. Alonso, F. Casati, H. Kuno, V. Machiraju
Copyright Springer Verlag Berlin Heidelberg
2004

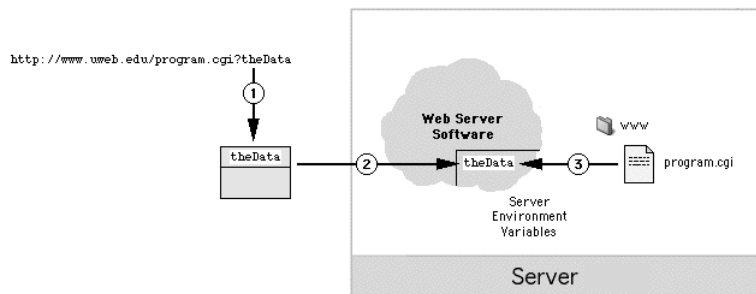
10-07

based on notes on the <http://www.cknuckles.com/webapps/> companion

35

GET Transaction

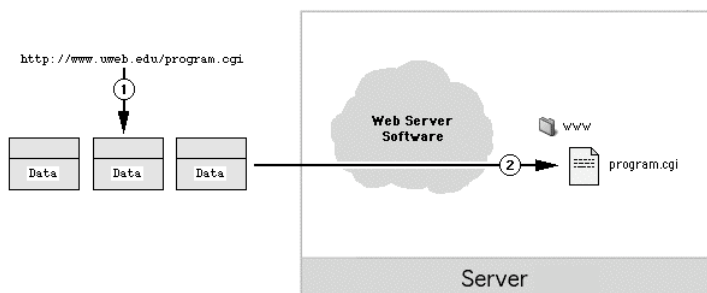
- ① The query string is sent to the web server in the application layer header.
- ② The Web server places the query string in one of its environment variables.
- ③ The CGI program accesses the environment variables and stores the query string into a variable within the program.



- HTTP GET -- Data in query string sent in packet header

POST Transaction

- ① The data is included in the bodies of the application layer packets and sent to the Web server.
- ② The Web server passes the data to the CGI program as the program's standard input stream.

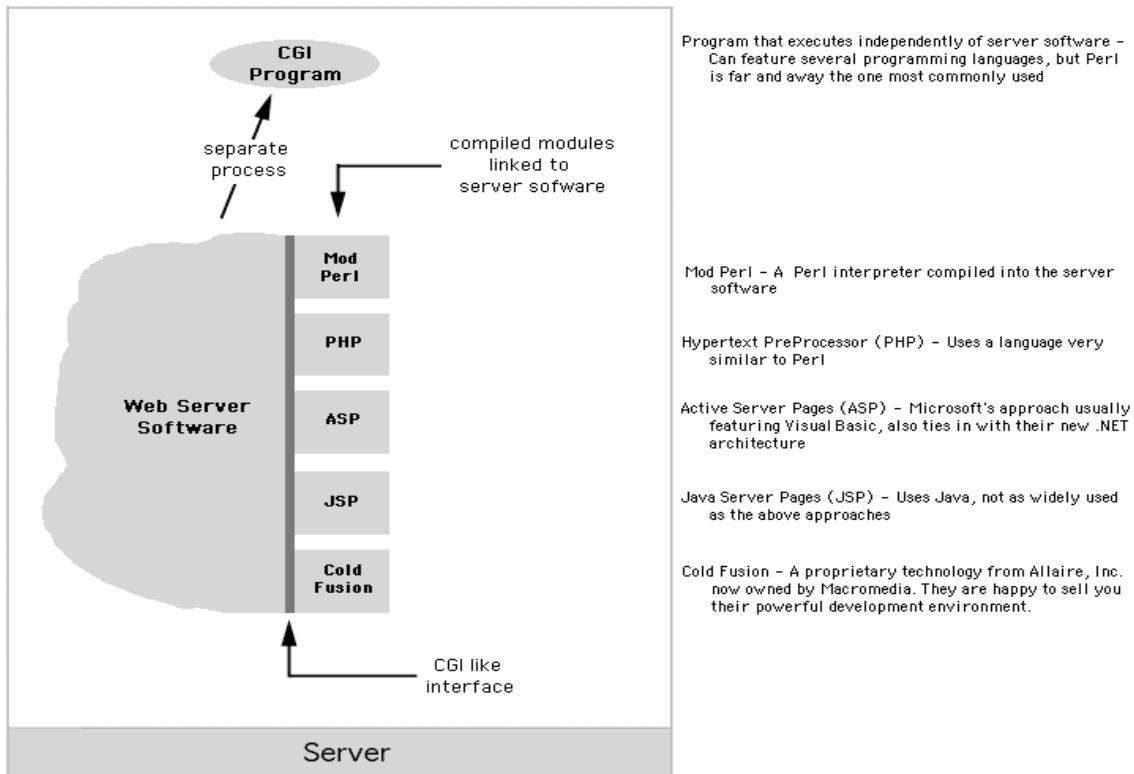


- HTTP POST -- Data is part of message payload.

s/ companion

36

Web technologies



Other middleware

- In addition to CGI
- Java servlets
- Web services

References

- A brief history of the internet:
<http://www.w3c.rl.ac.uk/primers/history/origins.htm>
- Information from an Internet guru:
<http://navigators.com/>
- Theodor Holm Nelson: Xanalogical Structure,
Needed Now More than Ever
<http://www.xanadu.com.au/ted/XUsurvey/xuDation.html>
- Tools
 - tcpmon
 - traceroute