

Introduction to HTTP and network services

Dan Wright

September 30, 2008

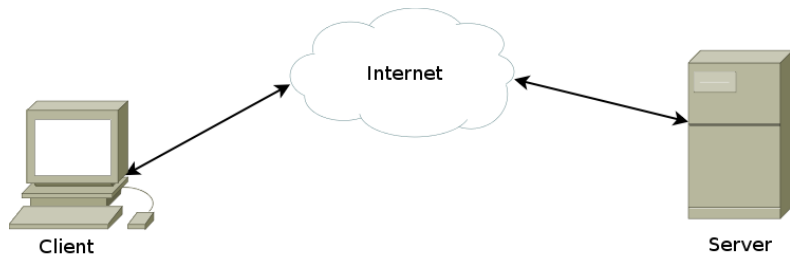
Network services architecture

The HTTP protocol

An example web session

Network services architecture

The client-server model



Most Internet services built using this model. The client may be any computer running client software (e.g., a web browser), and the server may be any computer running server software (e.g., a web server).

Layered architecture

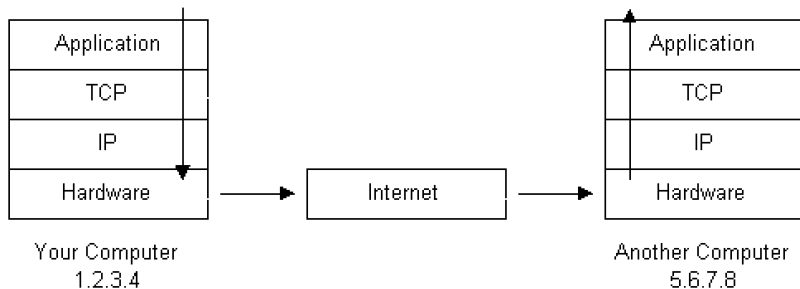


Image from http://www.theshulers.com/whitepapers/internet_whitepaper/

Addressing

- ▶ Every computer (client, server, mobile device, etc.) on the public Internet has a unique IP address
- ▶ Every network service runs on a standard port; this is what allows the client to contact services running on some unfamiliar server
- ▶ The combination of IP address + port number uniquely identifies a particular service on a particular machine
- ▶ Some common examples: SSH: 22, HTTP: 80, HTTPS: 443, SMTP: 25

Addressing cont'd.

- ▶ A very common network configuration has a number of IP devices “hidden” behind a firewall device that routes their traffic so that it all *appears* to come from a single address – this is how home wireless routers work, for example
- ▶ How does this potentially break the client/server model?

The HTTP protocol

The basics

- ▶ HTTP is a simple, text-based protocol designed to efficiently transfer the objects needed for rendering HTML hypertext documents
- ▶ Most other Internet application protocols are text-based (e.g., SMTP, IMAP, POP, and NNTP); this simplifies implementation and debugging

The details

- ▶ Unlike most other protocols, HTTP is connection-oriented; that is, it is designed to have each connection serve a single transaction, and then close (though note that HTTP 1.1 allows multiple transactions per connection)
- ▶ HTTP can be used to serve any data, not just HTML documents; examples include images, PDF files, Flash content, and any number of other file data types
- ▶ HTTP is essentially a data encapsulation protocol – it provides a minimal framework for a client to request and receive data, with as little overhead as possible

Security concerns

- ▶ Note the security implications of a plain text protocol: anyone who can see your traffic with a remote server can see exactly what you are doing – this includes passwords and any other sensitive information
- ▶ This concern is not unique to HTTP; it also applies to Telnet, SMTP, and any other of the plain text protocols
- ▶ The most common way to address this problem is with SSL, an application-level encryption wrapper

SSL/TLS

- ▶ Secure Sockets Layer / Transport Layer Security
- ▶ All “secure” web traffic is carried over SSL
- ▶ The security provided by SSL is of two distinct, and separate, types:
 - ▶ Identity verification: “proof” that the remote server is who they claim to be via public-key certificates
 - ▶ Transport encryption: wrapping HTTP traffic in robust encryption so sensitive data (like passwords or credit card numbers) can’t be grabbed off the wire by a 3rd party
- ▶ Note that while this protocol covers both, these are actually *separate* security concerns

An example web session

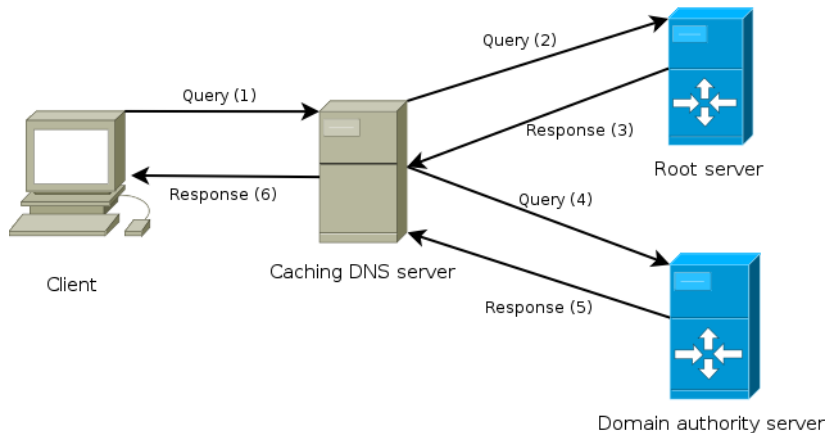
URL parsing

http://www.google.com/search?q=test

The diagram illustrates the parsing of the URL `http://www.google.com/search?q=test`. Four arrows point from labels below to the corresponding parts of the URL: 'Protocol' points to 'http://', 'Domain' points to 'www.google.com', 'Path' points to '/search', and 'Query parameters' points to '?q=test'.

Protocol Domain Path Query parameters

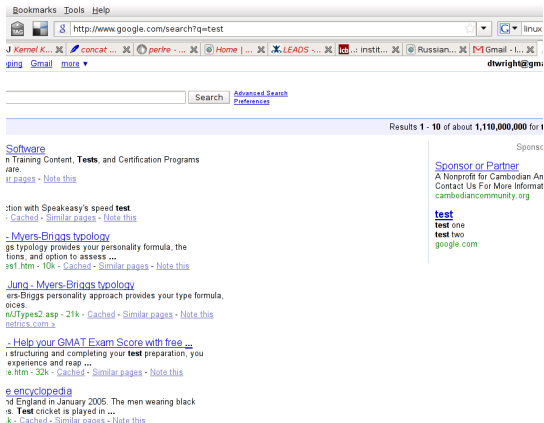
DNS lookup



Document retrieval, parsing, & rendering

GET /search?q=test HTTP/1.1

Connection: close



The screenshot shows a web browser window with the address bar containing 'http://www.google.com/search?q=test'. The browser's title bar includes 'Bookmarks Tools Help'. The search results page displays 'Results 1 - 10 of about 1,110,000,000 for test'. The first result is 'Software', which includes links to 'Training Content, Tests, and Certification Programs', 'Sponsor or Partner', and 'test one test two google.com'. Other results include 'Myers-Briggs typology' and 'Help your GMAT Exam Score with free ...'.

- ▶ The process of fetching a displaying a web page is a complex, multi-step process, potentially involving dozens of network transactions and several remote servers
- ▶ Thanks to the design of web client software and the architecture of Internet protocols, this complexity is almost completely hidden from end users

