

# How the Internet works: A (detailed) overview

J. Philip East<sup>1</sup>

This document describes my general understanding of how Internet communication occurs in general and for accessing a web page in particular. There are probably a couple of layers of detail beneath the one presented before one reaches the level of electricity and physics and a level or two above at which the process can be described. I believe an understanding of *about* this level is useful if one wishes to teach students about the Internet or to have them consider security and privacy issues of the Internet.

A glossary of terms is included at the end of this document. It includes acronyms and terms appearing in bold print (for their first appearance) in the body of the document. The terms are generally not defined in the body of the document, so you may wish to look the terms up when they are first encountered. I made up the definitions to correspond to this document and to communicate my understanding. Please send suggestions for improvement of the glossary (and document in general) to east@cs.uni.edu. The combination of scenario and glossary are meant to provide an understanding of Internet communication but are recognized to be incomplete.

I believe a scenario or case-study approach is useful to understanding general ideas. This document includes a scenario involving the loading of web pages. Students should/would be encouraged to develop their own statement of a similar scenario (e.g., mail, ftp, ssh, etc.).

## The Web Page Scenario

Suppose you are using a web browser and click on a link to some distant web page.

To you as a user, the computer displays the **URL** for page in the location box on the browser toolbar and sits for a moment or two perhaps showing some sign of activity, and then begins displaying the contents of the new page. Underneath this simple change a number of computers have been performing many actions that lead to accomplishing the task.

There are several major steps involved in getting and displaying that web page. First, the **IP address** of the textual URL must be ascertained. Then, a connection must be established between your computer's browser (the **client**) and the computer that hosts the web site (the **server**). Next, the client requests the desired web page and has the request evaluated by the server. The requested file is sent from the server to the client. Finally, the client begins rendering and displaying the file. In this final step it is likely that additional files (CSS, images, scripts, etc.) are required and must be requested using the same process.

### Step 1

The first step is to determine the IP address of the server hosting the web page that is being sought.

- ✗ the client connects to the local **DNS server** (supplied by your local organization or **ISP**) asking for an IP address for the remote server
- ✗ your local DNS server connects to the **whois** database server seeking the IP address for the primary **nameserver** of the remote server's domain
- ✗ your local DNS server connects to the domain's nameserver to find the IP address for the web server in question
- ✗ your DNS server reports back with the IP address of the desired server

It is at this point that you might get a "server not found" error message. The message is generated by your browser locally when either there was no entry for the requested server found in the DNS server hierarchy or some component of the DNS server hierarchy was not available.

---

<sup>1</sup> A colleague, Paul Gray, graciously answered a number of questions for me and fixed some of my errors. He is **not** responsible for any continued misunderstanding or miscommunication on my part.

## Step 2

Establishing the connection between the client and server involves tasks similar to the transfer of the data. This process is described in detail below and then referred to later on. Your browser has a component that implements **TCP** actions. The connection establishment begins with that component. (The process described below is a **three-way handshake**.)

- ✖ the TCP component built into browser (client) software creates a TCP synchronization (**SYN**) **packet** consisting of a header (source port, destination port, sequence number, some other overhead information, and a checksum) and no data (though the packet has a slot for data)
  - + an **ethernet** (or other local network protocol) packet is created with
    - packet header—source IP address, destination IP address, some other overhead information)
    - packet data—most of the TCP packet information
  - + the packet is sent out on **LAN**; the **hub/switch/router** notes that destination is beyond LAN and sends packet to next higher level within the local area. Eventually, the router to the **IP** (Internet Protocol) world is reached
  - + an IP packet is created by the outside-world connector/router
    - the relevant ethernet packet info is copied into the IP packet header (e.g., source & destination IP addresses, TCP packet info, ...)
    - additional IP packet information is included (e.g., checksum)
  - + the IP packet is sent to your **ISP** which will forward it (perhaps through a network of routers) to its **NAP** for an **NSP**
  - + the IP packet is sent through the NSP's router network to the NAP of the web server's ISP
  - + the IP packet is sent through the server's ISP router network to the web server's NAP
  - + the IP packet information is discarded and (if necessary) an ethernet (or other local network protocol) packet is created (as above). if the server is on a LAN, the packet is sent out on; a hub/switch/router notes that destination is inside its LAN and sends packet to next lower level within the local area. Eventually, the LAN containing the server is reached
  - + the web server receives the SYN packet
- ✖ the web server software acknowledges the connection request
  - + it creates an synchronization and acknowledgment (**SYN-ACK**) packet provides containing a header with source port, destination port, sequence/synchronization number, acknowledgment number, some other overhead information and a checksum, and no data
  - + the SYN-ACK packet is sent to client making the request—this is essentially the same process as above but in reverse
- ✖ the client acknowledges the acknowledgment—again, using essentially the same process as above. Thus, both ends have acknowledged the connection. Note that it is possible for this packet to serve double duty, i.e., to include both the acknowledgment and (as data in the packet) the requested URL.

There are several additional points about this process that are worth knowing.

- ✖ The "local" communication could have been a single step on both ends, i.e., the client communicating directly with its ISP and the server communicating directly with its ISP. Alternatively, the server could have been within the local collection of LAN or even on the same LAN as the client and no IP packet would have been needed.
- ✖ The packet is received, stored temporarily, examined for transmission (checksum) correctness and to determine the addressee, and then passed along at each router encountered along the path. Routers beyond the LAN will be communicating with their neighbors to maximize network efficiency.
- ✖ Packets have a limited life span. If not delivered before its time to live counter is counted down, a packet will be dropped (not sent on).
- ✖ Most Internet communication uses the concept of timing-out. If an anticipated action does not occur within the expected amount of time, the process is begun again, an error message is displayed, or nothing further happens (with respect to that particular action).

- ✖ Much Internet traffic uses clear-text and therefore can be read by anyone who intercepts it.

### Step 3

The next major step involves your browser requesting a web page/file. This process is very similar to that above except that the TCP packet will now have data—the URL of the requested page. A single packet will be created and sent to the server.

- ✖ application software creates a TCP "segment"/packet for the request with a header (as above) and data (the HTTP request for a particular web page).
- ✖ request sent to web server
  - + an ethernet (or other local network protocol) packet is created
    - packet header—source IP address, destination IP address, some other overhead information)
    - packet data—most of the TCP packet information
  - + the packet is sent out on the LAN; the hub/switch/router notes that destination is beyond LAN and sends packet to next higher level within the local area. Eventually, the router to the IP world is reached.
  - + an IP packet is created by the outside-world connector/router
    - the ethernet packet info is copied into IP packet header (e.g., source & destination IP addresses, TCP packet info, ...)
    - additional IP packet information included (e.g., new header checksum)
  - + the IP packet is sent to the ISP which will forward it (perhaps through a network of routers) to its NAP for a NSP
  - + the IP packet continues through the NSP router network to the NAP of the web server's ISP
  - + the IP packet is sent through ISP router network to web server's NAP
  - + the IP packet information is discarded and (if necessary) ethernet (or other local network protocol) packet is created (as above)
  - + if the server is on a LAN, the packet is sent out on; a hub/switch/router notes that destination is inside its LAN and sends packet to next lower level within the local area. Eventually, the LAN containing the server is reached
  - + web server receives the request

### Step 4

The server examines its file system and if the requested file exists and is available it will be prepared for transmission to the client. If it does not exist (or some part of the URL/path has been mistyped) or has not been made publicly readable by the owner of the file, an error message will be sent instead of the file. To transmit the file, the web server TCP software subdivides the file and creates TCP packets and:

- ✖ for each packet:
  - + prepares a header (source port, destination port, sequence number, acknowledgment number, some other overhead information and a checksum) and includes data (a chunk of the file being sent)
  - + sets a time check/alarm (if the packet is not acknowledged within a specified time, it will automatically be resent)
  - + passes it along to/through the LAN or router software for appropriate packaging and routing to the ISP and NSP (as described above)
- ✖ when/if a receipt acknowledgment is received from the client, the time check/alarm associated with a packet is disabled/deleted
- ✖ when/if a time check/alarm is recognized, the associated packet is resent [Packets will be lost when they are sent to a router that does not have sufficient buffer space to store it. This will typically be due to the busyness of the net. Packets will also be lost if it is determined that garbling has occurred during transmission which is determined by calculating a checksum on the received data and comparing it to the transmitted checksum.]

## Step 5

The client browser receives the packets and combines them into a file. The sequence numbers in the headers are used to tell the order in which they should appear. The browser then begins to render the HTML file and display it appropriately. While doing so, it is likely that additional files will be needed, i.e., CSS, images, scripts, etc. That will cause additional requests to the server. These additional requests would involve all the actions noted above for each file needed.

There is actually a sixth step for most connections—a graceful ending. This is accomplished with another three-way handshake in which the server sends a packet with the **FIN** flag set. The client acknowledges the FIN and the server acknowledges the FIN-ACK.

## Some Web References

<http://computer.howstuffworks.com/internet-infrastructure.htm/printable>

<http://www.securityfocus.com/infocus/1180>

[http://ocportal.com/site/pg/how\\_internet\\_works](http://ocportal.com/site/pg/how_internet_works)

<http://computer.howstuffworks.com/question525.htm/printable>

<http://www.answers.com/topic/proxy-server>

[http://en.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](http://en.wikipedia.org/wiki/Transmission_Control_Protocol)

<http://vlaurie.com/computers2/Articles/Name.htm>

<http://vlaurie.com/index.html>

<http://www.techsupportalert.com/c04100.htm>

[http://mvollo.com/blogs/serverside/archive/2006/10/16/Where-did-my-IIS7-server-go\\_3F00\\_-Troubleshooting-guide-for-\\_2200\\_server-not-found\\_2200\\_-errors.aspx](http://mvollo.com/blogs/serverside/archive/2006/10/16/Where-did-my-IIS7-server-go_3F00_-Troubleshooting-guide-for-_2200_server-not-found_2200_-errors.aspx)

<http://www.strum.co.uk/webbery/intranet.htm>

<http://computer.howstuffworks.com/vpn.htm/printable>

<http://support.microsoft.com/kb/172983>

<http://www.netbook.cs.purdue.edu/index.htm>

<http://www.netbook.cs.purdue.edu/othrpags/page15.htm>

# Glossary of Internet Terms

- ACK** — refers to the acknowledgment flag of IP packets and to packets in which the ACK flag has been set. See FIN and SYN.
- client** — a computer or application that is making a request. Of used in combination with “server”, the computer or application that fulfills requests. Both roles are action-based in that the same application or computer can be a client in one case and a server in a different case.
- DNS server** — acronym for “Domain Name Service” server. There is actually a hierarchy of DNS servers—the 13 root servers that know the primary names servers for each of the domains (domains such as edu, org, com), those actual primary names servers, and local domain name servers. (I also believe there are additional names servers.) These servers interact with each other and the “whois” database to provide IP addresses for textual names such as uni.edu.
- ethernet** — communication protocol now used mostly on local area networks (LANs). File/message is divided into packets and broadcast to all parties on the network.
- FIN** — a flag setting in IP packets that indicates communication is complete. A FIN packet will typically be acknowledged by the recipient in a packet with both the FIN and ACK flags set (FIN+ACK). And, the acknowledgement will be acknowledged by the originator (an ACK). This process is referred to as a three-way handshake.
- FIN-ACK** — refers to the acknowledgment of IP FIN packets in which the FIN and ACK flags have been set. See FIN.
- hub** — a device providing connections to a LAN. Computers, printers, etc. are connected to the hub. When messages are broadcast, all devices on the LAN will “hear” them.
- IP** — acronym for “Internet protocol”. It governs the actual transfer of information over the Internet from one router to the next, presumably getting closer to the destination with each transfer. Files being communicated will have been divided into packets and submitted to the IP system which sends them individually without knowledge or concern for the packet contents or connection to other packets.
- IP address** — “internet protocol address”. A number uniquely identifying each machine communicating on the Internet. The number is typically written in the form ###.###.###.### where each ### is a value in the range from 1 to 256 (or 0 to 255). IP addresses typically have textual equivalents that are useful for human consumption, e.g., east.cs.uni.edu is the textual version of 134.161.242.253. The numbers and text components match in reverse order 134-edu, 161-uni, cs-242, east-253 (this is not precisely correct but it illustrates the point reasonably).
- ISP** — acronym for “Internet Service Provider”. These companies connect individuals and organizations to network service providers (NSPs).
- LAN** — acronym for “local area network”. A set of machines (computers, printers, routers, etc. connected together. Typically, there is a router included that keeps local traffic local, sends externally bound traffic on, and accepts inbound traffic addressed to member of the LAN.
- nameserver** — see DNS server. Also, individual companies are assigned to administer domains such as .edu, .com, and .org, They assign and keep track of who has which name and provide that information to people and servers requesting it.
- NAP** — acronym for “Network Access Point”. This is merely a connect from a user or organization to an ISP or for an ISP to and NSP (and, of course, going the other way, an NSP to an ISP and an ISP to an individual or organization).
- NSP** — acronym for “Network Service Provider”. These companies provide the primary infrastructure for the Internet. If you compare the Internet to the highway system in the US, the NSPs provide the interstate highway system that have interchanges (NAPs, network access points) for the lesser roads (ISPs) which connect to city streets (LANs) or (in the country, to) driveways of individuals. Some of the NSPs are AT&T, MCI, and Sprint.

- packet** — a two-part (header and data) collection of information that is the basis for Internet communication. Breaking large files into pieces ensures that all users have reasonable chance to communicate and that difficulties in communication are minimized. Many communication protocols use packets. Common protocols are Ethernet in which packets are broadcast (essentially to the entire LAN or net) and TCP/IP in which packets are communicated from client to server but with individual packets being sent over possibly different paths. Packet headers provide information about source and destination, as well as flag values that affect or control the communication process.
- router** — a communication device that receives Internet traffic and passes it along to an appropriate destination. A router may separate two parts of a network in which case they recognize local traffic and keep it local and pass along outbound traffic. In this case, the router will also listen on the non-local side for traffic addressed to the local net. An alternative placement of the router is just as a node on the net in which case the router receives traffic, notes its destination, determines the currently best way to forward the traffic, and forwards it. In both cases the router ensures (to the extent possible) that header information was correctly transmitted and that further transmission is appropriate.
- server** — a computer or application that is fulfilling a request. Of used in combination with “client”, the computer or application that makes requests. Both roles are action-based in that the same application or computer can be a client in one case and a server in a different case.
- switch** — a device providing connections to a LAN. Computers, printers, etc. are connected to the switch. Messages arrive at the switch and are directed to the addressee of the message—others on the LAN do not “hear” them. The switch will pass externally-bound messages to a router and receive in-bound messages from the outside via the router.
- SYN** — a flag setting in IP packets that indicates request to open and synchronize a communication session. It is typically sent by a client process requesting some service. The SYN will be acknowledged with a packet which has the ACK flag set and contains a synchronization number (SYN+ACK). The SYN+ACK will be acknowledged (by the originator) with an ACK packet. This process is referred to as a three-way handshake.
- SYN-ACK** — refers to the acknowledgment of IP SYN packets in which the ACK flag has been set and a synchronization number has been included. See SYN.
- TCP** — acronym for “transfer control protocol”. It provides the guidelines/mechanisms for sequencing packets when they are sent over the Internet, checking to see that packets are received and sending them again when transmission was not successful, and putting the packets back together when received. Sequence numbers (beginning with a random number but then being incremented by one) provide continuity in communication. TCP sessions typically begin and end with three-way handshakes.
- TCP-IP** — the principle Internet communication protocol suite. It indicates and governs how communication will occur on the Internet. See the separate entries for TCP and IP.
- three-way handshake** — a set of communications that establishes or severs a TCP connection between a client and server. The process begins with a SYNchronization request (from a client) or FINalized communication (from a server), is followed by an annotated acknowledgment (SYN+ACK from server or FIN+ACK from client), and is terminated by a simple acknowledgment from the originating (client or server). Sequence numbers (beginning with a random number but then being incremented by one) provide continuity in communication.
- URL** — acronym for “Universal Resource Locator”. It includes the host ID and a path/file name indicating the name and location of the desired file. An example is `www.cs.uni.edu/east/web/index.html`. The host id is “www.cs.uni.edu”, the file is “index.html” and it is located in the “web” subdirectory of the “east” directory.
- whois** — a registry of internet domain names. The database identifies an organization and/or person(s) associated with a particular domain. It works much like a phone book. (Apparently multiple copies of the database exist and are hosted by different companies/entities.)