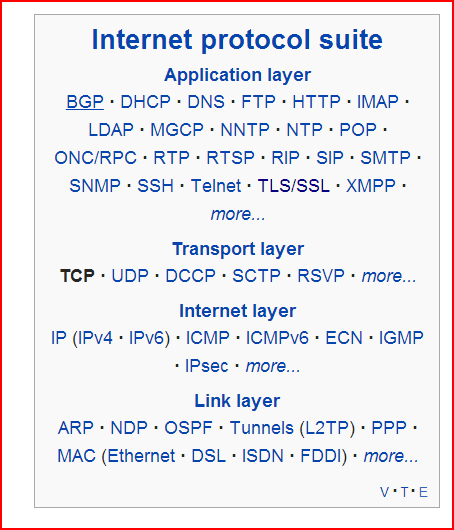
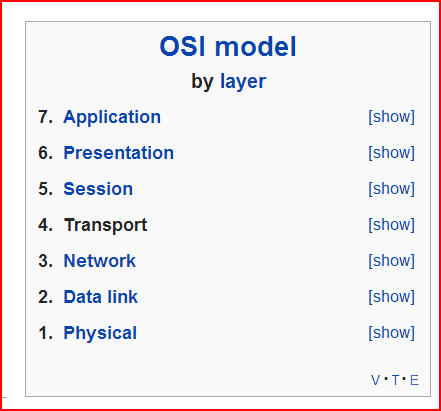
The protocol stack used on the Internet is refered to as the TCP/IP protocol stack because of the two major communication protocols used. The TCP/IP stack looks like this: 





|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **How Does the Internet Work?**  *© 2002*[*Rus Shuler*](mailto:russhuler@yahoo.com)*@*[*Pomeroy IT Solutions*](http://www.pomeroy.com/)*, all rights reserved*  **Contents**   1. [Introduction](http://www.theshulers.com/whitepapers/internet_whitepaper.html#intro) 2. [Where to Begin? Internet Addresses](http://www.theshulers.com/whitepapers/internet_whitepaper.html#addr) 3. [Protocol Stacks and Packets](http://www.theshulers.com/whitepapers/internet_whitepaper.html#stack) 4. [Networking Infrastructure](http://www.theshulers.com/whitepapers/internet_whitepaper.html#net_infra) 5. [Internet Infrastructure](http://www.theshulers.com/whitepapers/internet_whitepaper.html#int_infra) 6. [The Internet Routing Hierarchy](http://www.theshulers.com/whitepapers/internet_whitepaper.html#route) 7. [Domain Names and Address Resolution](http://www.theshulers.com/whitepapers/internet_whitepaper.html#dns) 8. [Internet Protocols Revisited](http://www.theshulers.com/whitepapers/internet_whitepaper.html#prot) 9. [Application Protocols: HTTP and the World Wide Web](http://www.theshulers.com/whitepapers/internet_whitepaper.html#http) 10. [Application Protocols: SMTP and Electronic Mail](http://www.theshulers.com/whitepapers/internet_whitepaper.html#smtp) 11. [Transmission Control Protocol](http://www.theshulers.com/whitepapers/internet_whitepaper.html#tcp) 12. [Internet Protocol](http://www.theshulers.com/whitepapers/internet_whitepaper.html#ip) 13. [Wrap Up](http://www.theshulers.com/whitepapers/internet_whitepaper.html#wrap_up) 14. [Resources](http://www.theshulers.com/whitepapers/internet_whitepaper.html#resources) 15. [Bibliography](http://www.theshulers.com/whitepapers/internet_whitepaper.html#bib)   **Introduction**  How does the Internet work? Good question! The Internet's growth has become explosive and it seems impossible to escape the bombardment of *www.com*'s seen constantly on television, heard on radio, and seen in magazines. Because the Internet has become such a large part of our lives, a good understanding is needed to use this new tool most effectively.  This whitepaper explains the underlying infrastructure and technologies that make the Internet work. It does not go into great depth, but covers enough of each area to give a basic understanding of the concepts involved. For any unanswered questions, a list of resources is provided at the end of the paper. Any comments, suggestions, questions, etc. are encouraged and may be directed to the author at rshuler@gobcg.com.  **Where to Begin? Internet Addresses**  Because the Internet is a global network of computers each computer connected to the Internet **must**have a unique address. Internet addresses are in the form **nnn.nnn.nnn.nnn** where nnn must be a number from 0 - 255. This address is known as an IP address. (IP stands for Internet Protocol; more on this later.)  The picture below illustrates two computers connected to the Internet; your computer with IP address 1.2.3.4 and another computer with IP address 5.6.7.8. The Internet is represented as an abstract object in-between. (As this paper progresses, the Internet portion of Diagram 1 will be explained and redrawn several times as the details of the Internet are exposed.)   |  | | --- | | Diagram 1 | | Diagram 1 |   If you connect to the Internet through an Internet Service Provider (ISP), you are usually assigned a temporary IP address for the duration of your dial-in session. If you connect to the Internet from a local area network (LAN) your computer might have a permanent IP address or it might obtain a temporary one from a DHCP (Dynamic Host Configuration Protocol) server. In any case, if you are connected to the Internet, your computer has a unique IP address.   |  | | --- | | **Check It Out - The Ping Program** | | If you're using Microsoft Windows or a flavor of Unix and have a connection to the Internet, there is a handy program to see if a computer on the Internet is alive. It's called **ping**, probably after the sound made by older submarine sonar systems.1 If you are using Windows, start a command prompt window. If you're using a flavor of Unix, get to a command prompt. Type ping www.yahoo.com. The ping program will send a 'ping' (actually an ICMP (Internet Control Message Protocol) echo request message) to the named computer. The pinged computer will respond with a reply. The ping program will count the time expired until the reply comes back (if it does). Also, if you enter a domain name (i.e. www.yahoo.com) instead of an IP address, ping will resolve the domain name and display the computer's IP address. More on domain names and address resolution later. |     **Protocol Stacks and Packets**  So your computer is connected to the Internet and has a unique address. How does it 'talk' to other computers connected to the Internet? An example should serve here: Let's say your IP address is 1.2.3.4 and you want to send a message to the computer 5.6.7.8. The message you want to send is "Hello computer 5.6.7.8!". Obviously, the message must be transmitted over whatever kind of wire connects your computer to the Internet. Let's say you've dialed into your ISP from home and the message must be transmitted over the phone line. Therefore the message must be translated from alphabetic text into electronic signals, transmitted over the Internet, then translated back into alphabetic text. How is this accomplished? Through the use of a **protocol stack**. Every computer needs one to communicate on the Internet and it is usually built into the computer's operating system (i.e. Windows, Unix, etc.). The protocol stack used on the Internet is refered to as the TCP/IP protocol stack because of the two major communication protocols used. The TCP/IP stack looks like this:    |  |  | | --- | --- | | **Protocol Layer** | **Comments** | | Application Protocols Layer | Protocols specific to applications such as WWW, e-mail, FTP, etc. | | Transmission Control Protocol Layer | TCP directs packets to a specific application on a computer using a port number. | | Internet Protocol Layer | IP directs packets to a specific computer using an IP address. | | Hardware Layer | Converts binary packet data to network signals and back. (E.g. ethernet network card, modem for phone lines, etc.) |   If we were to follow the path that the message "Hello computer 5.6.7.8!" took from our computer to the computer with IP address 5.6.7.8, it would happen something like this:   |  | | --- | | Diagram 2 | | Diagram 2 |  1. The message would start at the top of the protocol stack on your computer and work it's way downward. 2. If the message to be sent is long, each stack layer that the message passes through may break the message up into smaller chunks of data. This is because data sent over the Internet (and most computer networks) are sent in manageable chunks. On the Internet, these chunks of data are known as **packets**. 3. The packets would go through the Application Layer and continue to the TCP layer. Each packet is assigned a **port number**. Ports will be explained later, but suffice to say that many programs may be using the TCP/IP stack and sending messages. We need to know which program on the destination computer needs to receive the message because it will be listening on a specific port. 4. After going through the TCP layer, the packets proceed to the IP layer. This is where each packet receives it's destination address, 5.6.7.8. 5. Now that our message packets have a port number and an IP address, they are ready to be sent over the Internet. The hardware layer takes care of turning our packets containing the alphabetic text of our message into electronic signals and transmitting them over the phone line. 6. On the other end of the phone line your ISP has a direct connection to the Internet. The ISPs **router**examines the destination address in each packet and determines where to send it. Often, the packet's next stop is another router. More on routers and Internet infrastructure later. 7. Eventually, the packets reach computer 5.6.7.8. Here, the packets start at the bottom of the destination computer's TCP/IP stack and work upwards. 8. As the packets go upwards through the stack, all routing data that the sending computer's stack added (such as IP address and port number) is stripped from the packets. 9. When the data reaches the top of the stack, the packets have been re-assembled into their original form, "Hello computer 5.6.7.8!"   **Networking Infrastructure**  So now you know how packets travel from one computer to another over the Internet. But what's in-between? What actually makes up the Internet? Let's look at another diagram:   |  | | --- | | Diagram 3 | | Diagram 3 |   Here we see Diagram 1 redrawn with more detail. The physical connection through the phone network to the Internet Service Provider might have been easy to guess, but beyond that might bear some explanation.  The ISP maintains a pool of modems for their dial-in customers. This is managed by some form of computer (usually a dedicated one) which controls data flow from the modem pool to a backbone or dedicated line router. This setup may be refered to as a port server, as it 'serves' access to the network. Billing and usage information is usually collected here as well.  After your packets traverse the phone network and your ISP's local equipment, they are routed onto the ISP's backbone or a backbone the ISP buys bandwidth from. From here the packets will usually journey through several routers and over several backbones, dedicated lines, and other networks until they find their destination, the computer with address 5.6.7.8. But wouldn't it would be nice if we knew the exact route our packets were taking over the Internet? As it turns out, there is a way...   |  | | --- | | **Check It Out - The Traceroute Program** | |

When you type a URL into a web browser, this is what happens:

1. If the URL contains a domain name, the browser first connects to a domain name server and retrieves the corresponding IP address for the web server.
2. The web browser connects to the web server and sends an HTTP request (via the protocol stack) for the desired web page.
3. The web server receives the request and checks for the desired page. If the page exists, the web server sends it. If the server cannot find the requested page, it will send an HTTP 404 error message. (404 means 'Page Not Found' as anyone who has surfed the web probably knows.)
4. The web browser receives the page back and the connection is closed.
5. The browser then parses through the page and looks for other page elements it needs to complete the web page. These usually include images, applets, etc.
6. For each element needed, the browser makes additional connections and HTTP requests to the server for each element.
7. When the browser has finished loading all images, applets, etc. the page will be completely loaded in the browser window.