COMPUTER NETWORKING

CSE 3034

Evaluation Scheme

Internal Components	Marks	External Components	Marks
Mid Term Exam	15	End Sem Exam	45
Assignment	10	External Project	15
Quiz/Viva	10		
Attendance	05		
Total	**40**	**Total**	**60**

Course Format

- *3 Classes/week, 1hr/Class*
- *1 Lab/Week, 2hr/Lab*
- *4 Credits*

Textbook

Java Network Programming by Harold, O'Reilly (Shroff Publishers)

Experiment-01

BASIC NETWORKING CONCEPTS

- This Experiment covers
  - The nature of networks
  - The TCP/IP layer model
  - The IP, TCP, and UDP protocols
  - Firewalls and proxy servers
  - The Internet

Networks:

- A network is a collection of computers and other devices that can send data to and receive data from one another, more or less in real-time.
- Each machine on a network is called a node. Most nodes are computers, but printers, routers, bridges, gateways, dumb terminals, and Coca-Cola™ machines can also be nodes.
- Nodes that are fully functional computers are also called hosts.
- Every network node has an address, a sequence of bytes uniquely identifying it.
- The more bytes there are in each address, the more addresses available and the more devices that can be connected to the network simultaneously.
- Addresses are assigned differently on different kinds of networks. Ethernet addresses are attached to the physical Ethernet hardware.

Networks:

- Internet addresses are normally assigned to a computer by the organization responsible. However, the addresses an organization can choose for its computers are assigned by its Internet service provider (ISP).
- ISPs get their IP addresses from one of four regional organization Internet registries (the registry for North America is ARIN, the American Registry for Internet Numbers), which are assigned IP addresses by the Internet Corporation for Assigned Names and Numbers (ICANN).
- On some kinds of networks, nodes also have text names that help human beings identify them, such as “www.elharo.com” or “Beth Harold’s Computer.” A particular name normally refers to exactly one address at a specific time. However, names are not locked to addresses. Names can change while addresses stay the same; likewise, addresses can change while the names stay the same. One address can have several names and refer to several addresses.

- All modern computer networks are **packet-switched networks**: data traveling on the network is broken into chunks called packets, and each packet is handled separately. Each packet contains information about who sent it and where it’s going.
- The most crucial advantage of breaking data into individually addressed packets is that packets from many ongoing exchanges can travel on one wire, making building a network much cheaper: many computers can share the same wire without interfering. Another advantage of packets is that checksums can detect whether a packet was damaged in transit.

- A **protocol** is a precise set of rules defining how computers communicate: the format of addresses, how data is split into packets, and so on. There are many different protocols defining different aspects of network communication.
- For example, the Hypertext Transfer Protocol (HTTP) defines how web browsers and servers communicate; at the other end of the spectrum, the IEEE 802.3 standard defines a protocol for how bits are encoded as electrical signals on a particular type of wire.
- A web server doesn’t care whether the client is a Unix workstation, an Android phone, or an iPad because all clients speak the same HTTP protocol regardless of platform.

The Layers of a Network:

- Sending data across a network is a complex operation that must be carefully tuned to the physical characteristics of the network as well as the logical character of the data being sent.
- Software that sends data across a network must understand how to avoid collisions between packets, convert digital data to analog signals, detect and correct errors, route packets from one host to another, and more.
- The different aspects of network communication are separated into multiple layers to hide most of this complexity from the application developer and end-user.
- Each layer represents a different level of abstraction between the physical hardware (i.e., the wires and electricity) and the information being transmitted.
- Theoretically, each layer only talks to the layers immediately above and below it. Separating the network into layers lets you modify or even replace the software in one layer without affecting the others, as long as the interfaces between the layers stay the same.

- Figure 1-1 shows a stack of possible protocols in a network.
- While the middle layer protocols are fairly consistent across most of the Internet today, the top and the bottom vary a lot.
- Some hosts use Ethernet; some use Wi-Fi; some use PPP; some use something else.
- Similarly, what's on the top of the stack will depend entirely on which programs a host is running.
- The key is that from the top of the stack, it doesn't matter what's on the bottom and vice versa.
- The layer model decouples the application protocols from the network hardware's physics and the network connections' topology.

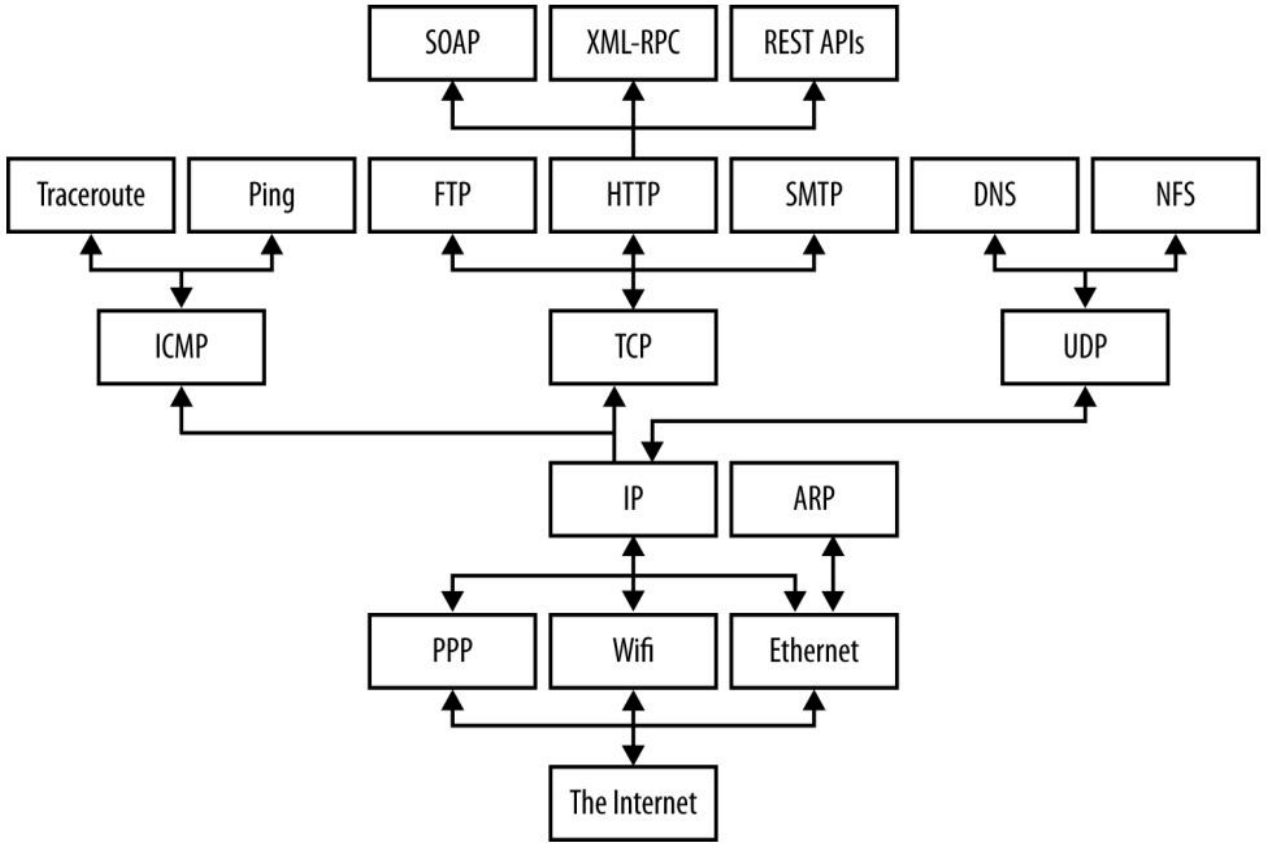

Figure 1-1. Protocols in different layers of a network

- There are several different layer models, each organized to fit the needs of a particular kind of network.
- This uses the standard TCP/IP four-layer model appropriate for the Internet, shown in Figure 1-2.
- In this model, applications like Firefox and Warcraft run in the application layer and talk only to the transport layer.
- The transport layer talks only to the application layer and the Internet layer.
- The Internet layer talks only to the host-to-network layer and the transport layer, never directly to the application layer.
- The host-to-network layer moves the data across the wires, fiber-optic cables, or other mediums to the host-to-network layer on the remote system, which then moves the data up the layers to the application on the remote system.

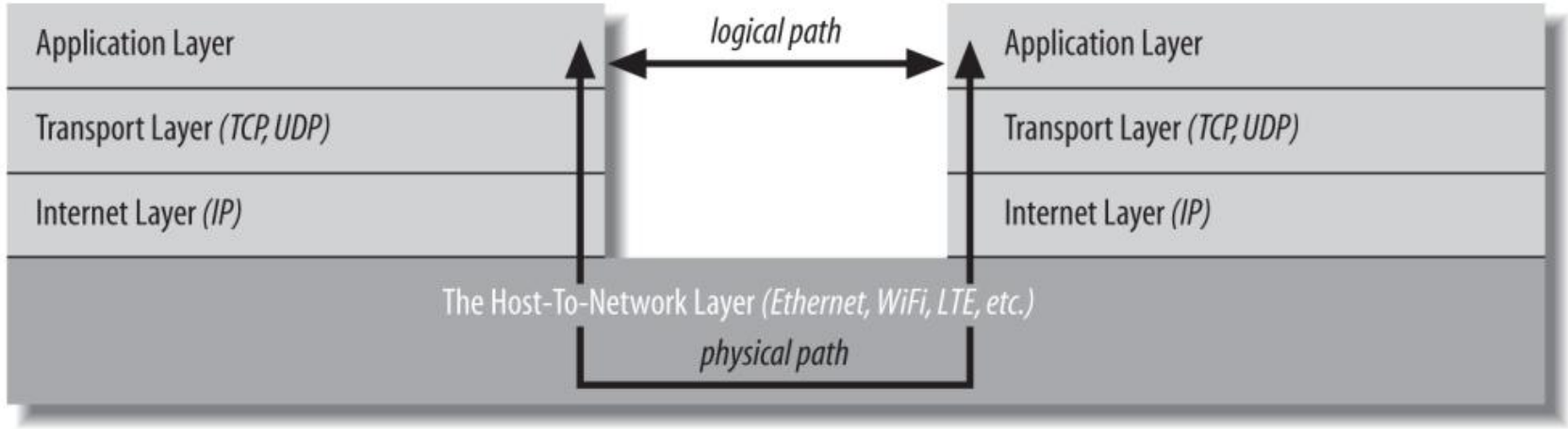

Figure 1-2. The layers of a network

- A seven-layer model is called the Open Systems Interconnection (OSI) Reference Model.
- The most significant difference between the OSI model and the TCP/IP model is that the OSI model splits the host-to-network layer into data link and physical layers and inserts presentation and session layers in between the application and transport layers.
- The OSI model is more general and better suited for non-TCP/ IP networks.
- Java’s network classes only work on TCP/IP networks and always in the application or transport layers.
- The application layer is talking directly to the application layer on the other system; the network creates a logical path between the two application layers.
- Everything more than one layer deep is effectively invisible.

The Host-to-Network Layer:

- In the standard reference model for IP-based Internet, the hidden parts of the network belong to the host-to-network layer (also known as the link layer, data link layer, or network interface layer).
- The host-to-network layer defines how a network interface—such as an Ethernet card or a WiFi antenna—sends IP datagrams over its physical connection to the local network and the world.
- The part of the host-to-network layer made up of the hardware that connects different computers (wires, fiber-optic cables, radio waves, or smoke signals) is sometimes called the network's physical layer.

The Internet Layer :

- In the OSI model, the internet layer goes by the generic network layer.
- A network layer protocol defines how bits and bytes of data are organized into larger groups called packets and the addressing scheme by which different machines find one another.
- The Internet Protocol (IP) is the world's most widely used network layer protocol, and the only one Java understands. It has two protocols: IPv4, which uses 32-bit addresses, and IPv6, which uses 128-bit addresses and adds a few other technical features to assist with routing.
- In IPv4 and IPv6, data is sent across the internet layer in packets called data-grams.
- Each IPv4 datagram contains a header between 20 and 60 bytes long and a payload that contains up to 65,515 bytes of data. (In practice, most IPv4 datagrams are much smaller, ranging from a few dozen bytes to a little more than eight kilobytes.) An IPv6 datagram contains a larger header and up to four gigabytes of data.

- Figure 1-3 shows how the different quantities are arranged in an IPv4 datagram.
- All bits and bytes are big-endian; most significant to least significant runs left to right.
- Besides routing and addressing, the second purpose of the Internet layer is to enable different types of Host-to-Network layers to talk to each other.
- Internet routers translate between WiFi and Ethernet, Ethernet and DSL, DSL and fiber-optic backhaul protocols, etc.
- Without the internet layer or something like it, each computer could only talk to other computers that shared its particular type of network.
- The internet layer is responsible for connecting heterogeneous networks using homogeneous protocols.

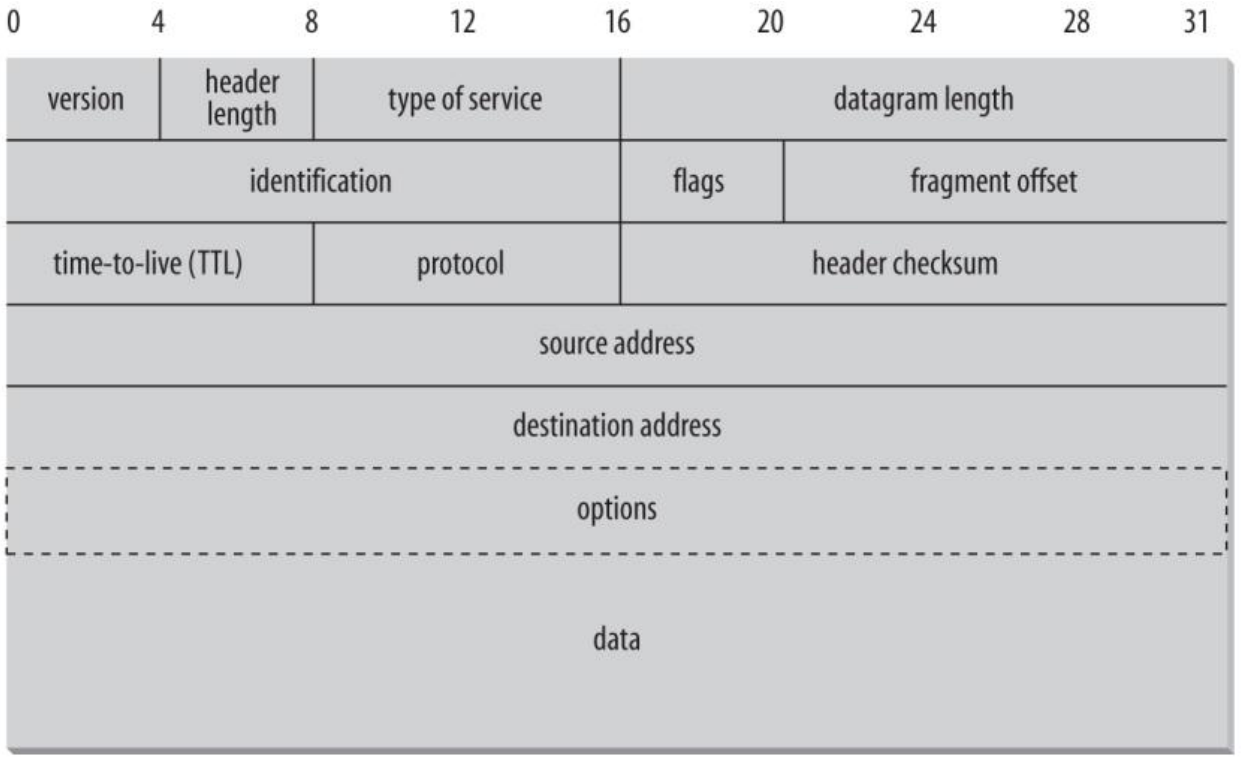

Figure 1-3. The structure of an IPv4 datagram

The Transport Layer:

- The transport layer ensures that packets are received in the order they were sent and that no data is lost or corrupted.
- If a packet is lost, the transport layer can ask the sender to retransmit the packet.
- IP networks implement this by adding an additional header to each datagram that contains more information.
- There are two primary protocols at this level. The first, the Transmission Control Protocol (TCP), is a high-overhead protocol that allows for the retransmission of lost or corrupted data and the delivery of bytes in the order they were sent. The second protocol, the User Datagram Protocol (UDP), allows the receiver to detect corrupted packets but only guarantees that packets are delivered in the correct order (or at all).
- However, UDP is often much faster than TCP. TCP is called a reliable protocol; UDP is an unreliable protocol. Later, you'll see that unreliable protocols are much more helpful than they sound.

The Application Layer :

- The layer that delivers data to the user is called the application layer.
- The three lower layers define how data is transferred from one computer to another.
- The application layer decides what to do with the data after it's transferred. For example, an application protocol like HTTP (for the World Wide Web) ensures that your web browser displays a graphic image as a picture, not a long stream of numbers.
- The application layer is where most of the network parts of your programs spend their time.
- There is an entire alphabet soup of application layer protocols: in addition to HTTP for the Web, there are SMTP, POP, and IMAP for email; FTP, FSP, and TFTP for file transfer; NFS for file access; Gnutella and Bit-Torrent for file sharing; the Session Initiation Protocol (SIP) and Skype for voice communication; and many, many more.

IP, TCP, and UDP:

- IP, the Internet protocol, was developed with military sponsorship during the Cold War and ended up with many features that the military was interested in.
- First, it had to be robust. Therefore, IP was designed to allow multiple routes between two points and route data packets around damaged routers. Second, the military had many different kinds of computers, and all of them had to be able to talk to one another.
- Therefore, IP had to be open and platform-independent. Because there are multiple routes between two points and the quickest path between two points may change over time as a function of network traffic and other factors, the packets that make up a particular data stream may not all take the same route.
- Furthermore, they may not arrive in the order they were sent, if they even arrive at all.
- To improve the basic scheme, TCP was layered on top of the IP to give each connection end the ability to acknowledge receipt of IP packets and request retransmission of lost or corrupted packets. Furthermore, TCP allows the packets to be put back together on the receiving end in the same order they were sent.

- TCP, however, carries a fair amount of overhead. Therefore, if the order of the data isn't significant and the loss of individual packets won't completely corrupt the data stream, packets are sometimes sent without the guarantees TCP provides using the UDP protocol.
- UDP is an unreliable protocol that does not guarantee that packets will arrive at their destination or that they will arrive in the same order they were sent. Although this would be a problem for uses such as file transfer, it is perfectly acceptable for applications where the loss of some data would go unnoticed by the end user.
- Error-correcting codes can be built into UDP data streams at the application level to account for missing data.
- Several other protocols can run on top of IP. The most commonly requested is ICMP, the Internet Control Message Protocol, which uses raw IP datagrams to relay error messages between hosts. The best-known use of this protocol is in the ping program.
- Java does not support ICMP, nor does it allow the sending of raw IP datagrams (as opposed to TCP segments or UDP datagrams).
- The only protocols Java supports are TCP, UDP, and application layer protocols built on top of these. All other transport layer, internet layer, and lower layer protocols such as ICMP, IGMP, ARP, RARP, RSVP, and others can only be implemented in Java programs by linking to native code

IP Addresses and Domain Names:

- A four-byte number identifies every computer on an IPv4 network.
- This is usually written in a dotted quad format like 199.1.32.90, where each of the four numbers is one unsigned byte ranging in value from 0 to 255.
- Every computer attached to an IPv4 network has a unique four-byte address.
- When data is transmitted across the network, the packet's header includes the address of the machine for which the packet is intended (the destination address) and the address of the machine that sent the packet (the source address).
- Along the way, Routers choose the best route to send the packet by inspecting the destination address.
- The source address is included so the recipient will know who to reply to. There are more than four billion possible IP addresses, not even one for every person on the planet, much less for every computer.

- A slow transition is underway to IPv6, which will use 16-byte addresses.
- This provides enough IP addresses to identify every person, every computer, and device on the planet. IPv6 addresses are customarily written in eight blocks of four hexadecimal digits separated by colons, such as FEDC:BA98:7654:3210:FEDC: BA98:7654:3210.
- Leading zeros do not need to be written. A double colon, at most one of which may appear in any address, indicates multiple zero blocks. For example, FEDC: 0000:0000:0000:00DC:0000:7076:0010 could be written more compactly as FEDC::DC: 0:7076:10.
- In mixed networks of IPv6 and IPv4, the last four bytes of the IPv6 address are sometimes written as an IPv4 dotted quad address. For example, FEDC:BA98:7654:3210:FEDC: BA98:7654:3210 could be written as FEDC:BA98:7654:3210:FEDC: BA98:118.84.50.16.
- Although computers are very comfortable with numbers, humans aren't good at remembering them. Therefore, the Domain Name System (DNS) was developed to translate hostnames humans can remember, such as "www.oreilly.com," into numeric Internet addresses such as 208.201.239.101.
- When Java programs access the network, they must process these numeric addresses and their host names.

Table 1-1. Well-known port assignments

Protocol	Port	Protocol	Purpose
echo	7	TCP/UDP	Echo is a test protocol used to verify that two machines are able to connect by having one echo back the other's input.
discard	9	TCP/UDP	Discard is a less useful test protocol in which all data received by the server is ignored.
daytime	13	TCP/UDP	Provides an ASCII representation of the current time on the server.
FTP data	20	TCP	FTP uses two well-known ports. This port is used to transfer files.
FTP	21	TCP	This port is used to send FTP commands like `put` and `get`.
SSH	22	TCP	Used for encrypted, remote logins.
Telnet	23	TCP	Used for interactive, remote command-line sessions.
smtp	25	TCP	The Simple Mail Transfer Protocol is used to send email between machines.
time	37	TCP/UDP	A time server returns the number of seconds that have elapsed on the server since midnight, January 1, 1900, as a four-byte, unsigned, big-endian integer.
whois	43	TCP	A simple directory service for Internet network administrators.
finger	79	TCP	A service that returns information about a user or users on the local system.
HTTP	80	TCP	The underlying protocol of the World Wide Web.
POP3	110	TCP	Post Office Protocol version 3 is a protocol for the transfer of accumulated email from the host to sporadically connected clients.
NNTP	119	TCP	Usenet news transfer; more formally known as the "Network News Transfer Protocol."
IMAP	143	TCP	Internet Message Access Protocol is a protocol for accessing mailboxes stored on a server.
dict	2628	TCP	A UTF-8 encoded dictionary service that provides definitions of words.

The Internet :

- The Internet is the world's largest IP-based network.
- It is an amorphous group of computers in many different countries on all seven continents (Antarctica included) that talk to one another using IP protocols.
- Each computer on the Internet has at least one IP address by which it can be identified.
- Many of them also have at least one name that maps to that IP address.
- It is simply a large collection of computers that have agreed to talk to one another in a standard way. The Internet is not the only IP-based network but the largest one. Other IP networks are called internets with a bit of i: for example, a high-security internal network that is not connected to the global Internet.
- Intranet loosely describes corporate practices of putting lots of data on internal web servers that are not visible to users outside the local network.
- Unless you're working in a high-security environment that's physically disconnected from the broader network, it's likely that the internet you'll be using is the Internet.

Internet Address:

- Blocks of IPv4 addresses are assigned to Internet service providers (ISPs) by their regional Internet registry.
- When a company or an organization wants to set up an IP-based network connected to the Internet, their ISP assigns them a block of addresses.
- Each block has a fixed prefix.
- For instance, if the prefix is 216.254.85, the local network can use addresses from 216.254.85.0 to 216.254.85.255. Because this block fixes the first 24 bits, it's called a /24. A /23 specifies the first 23 bits, leaving 9 bits for 29 or 512 local IP addresses.
- A /30 subnet (the smallest possible) specifies the first 30 bits of the IP addresses within the subnetwork, leaving 2 bits for 22 or 4 total local IP addresses.
- However, the lowest address in all blocks is used to identify the network itself, and the most significant address is a broadcast address for the network, so you have two fewer available addresses than you might first expect.

Network Address Translation:

- Because of the increasing scarcity of and demand for raw IP addresses, most networks today use Network Address Translation (NAT).
- In NAT-based networks, most nodes only have local, non-routable addresses selected from 10. x.x.x, 172.16.x.x to 172.31.x.x, or 192.168.x.x.
- The routers that connect the local networks to the ISP translate these local addresses to a much smaller set of routable addresses.

Firewalls :

- There are some naughty people on the Internet. To keep them out, it’s often helpful to set up one point of access to a local network and check all traffic into or out of that access.
- A firewall is called the hardware and software that sit between the Internet and the local network, checking all the data that comes in or out to ensure it’s kosher.
- The firewall is often part of the router that connects the local network to the broader Internet and may perform other tasks, such as network address translation.
- More intelligent firewalls look at the contents of the packets to determine whether to accept or reject them.
- The exact configuration of a firewall—which packets of data are and to pass through and which are not—depends on the security needs of an individual site.
- Java doesn’t have much to do with firewalls—except insofar as they often get in your way.

Proxy Servers:

- Proxy servers are related to firewalls. If a firewall prevents hosts on a network from making direct connections to the outside world, a proxy server can act as a go-between.
- Thus, a machine that is prevented from connecting to the external network by a firewall would request a web page from the local proxy server instead of asking for the web page directly from the remote web server.
- The proxy server would then request the page from the web server and forward the response to the original requester. Proxies can also be used for FTP services and other connections. One of the security advantages of using a proxy server is that external hosts only find out about the proxy server.
- They need to learn the names and IP addresses of the internal machines, making it more difficult to hack into internal systems.
- Whereas firewalls generally operate at the transport or internet layer level, proxy servers typically operate at the application layer.
- A proxy server has a detailed understanding of some application-level protocols, such as HTTP and FTP. (The notable exception is SOCKS proxy servers that operate at the transport layer and can proxy for all TCP and UDP connections regardless of application layer protocol.)
- Packets that pass through the proxy server can be examined to ensure they contain data appropriate for their type. For instance, FTP packets with Telnet data can be rejected. Figure 1-4 shows how proxy servers fit into the layer model.

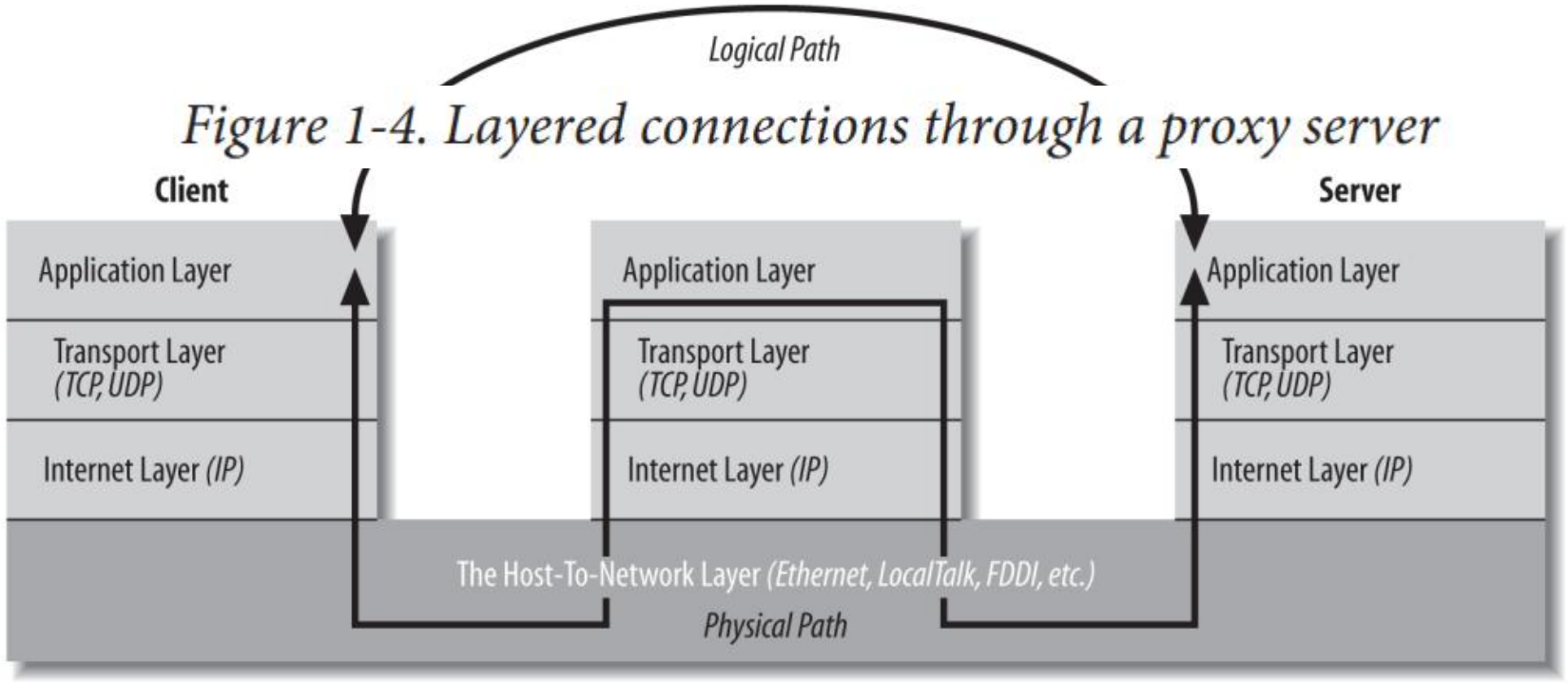

Figure 1-4. Layered connections through a proxy server

- As long as all access to the Internet is forwarded through the proxy server, access can be tightly controlled.
- Proxy servers can also be used to implement local caching. When a file is requested from a web server, the proxy server checks to see if it is in its cache. If the file is in the cache, the proxy serves the file from the cache rather than the Internet. If the file is not in the cache, the proxy server retrieves it, forwards it to the requester, and stores it in the cache for the next time it is requested. This scheme can significantly reduce the load on an Internet connection and greatly improve response time.
- America Online runs one of the world's largest farms of proxy servers to speed up user data transfer.
- The biggest problem with proxy servers is their inability to cope with all but a few protocols.

- Generally established protocols like HTTP, FTP, and SMTP can pass through, while newer protocols like BitTorrent are not.This is a significant disadvantage in the rapidly changing world of the Internet.
- It’s a disadvantage for Java programmers because it limits the effectiveness of custom protocols. Creating a new protocol optimized for your application in Java is accessible and often helpful. However, no proxy server will ever understand these one-of-a-kind protocols.
- Consequently, some developers have taken to tunneling their protocols through HTTP, most notably with SOAP. However, this has a significant negative impact on security.
- The firewall usually is there for a reason, not just to annoy Java programmers. Applets that run in web browsers typically use the proxy server settings of the web browser itself, though these can be overridden in the Java Control Panel.
- Standalone Java applications can indicate the proxy server to use by setting the socksProxyHost and socksProxyPort properties (if you’re using a SOCKS proxy server) or http.proxySet, http.proxyHost, http.proxyPort, https.proxySet, https.proxy Host, https.proxyPort, ftpProxySet, ftpProxyHost, ftpProxyPort, gopherProxy Set, gopherProxyHost, and gopherProxyPort system properties (if you’re using protocol-specific proxies).
- You can set system properties from the command line using the -D flag, like this: java -DsocksProxyHost=socks.cloud9.net -DsocksProxyPort=1080 MyClass

The Client/Server Model :

- Most modern network programming is based on a client/server model.
- A client/server application typically stores large quantities of data on an expensive, high-powered server or cloud of servers. At the same time, most of the program logic and the user interface are handled by client software running on relatively cheap personal computers.
- In most cases, a server sends data while a client primarily receives it, but it is rare for one program to send or receive exclusively.
- A more reliable distinction is that a client initiates a conversation while a server waits for clients to start conversations with it. Sometimes, the same program may be a client and a server.

- Not all applications fit easily into a client/server model. For instance, in networked games, both players will likely send data back and forth roughly equally (at least in a fair game). These sorts of connections are called peer-to-peer. The telephone system is a classic example of a peer-to-peer network.
- Each phone can either call another phone or be called by another phone. You don't have to buy one phone to send calls and another to receive them. Java does not have explicit peer-to-peer communication in its core networking API. However, applications can easily offer peer-to-peer communications in several ways, most commonly by acting as both a server and a client.
- Alternatively, the peers can communicate through an intermediate server program that forwards data from one peer to the other. This neatly solves the discovery problem of how two peers find each other.

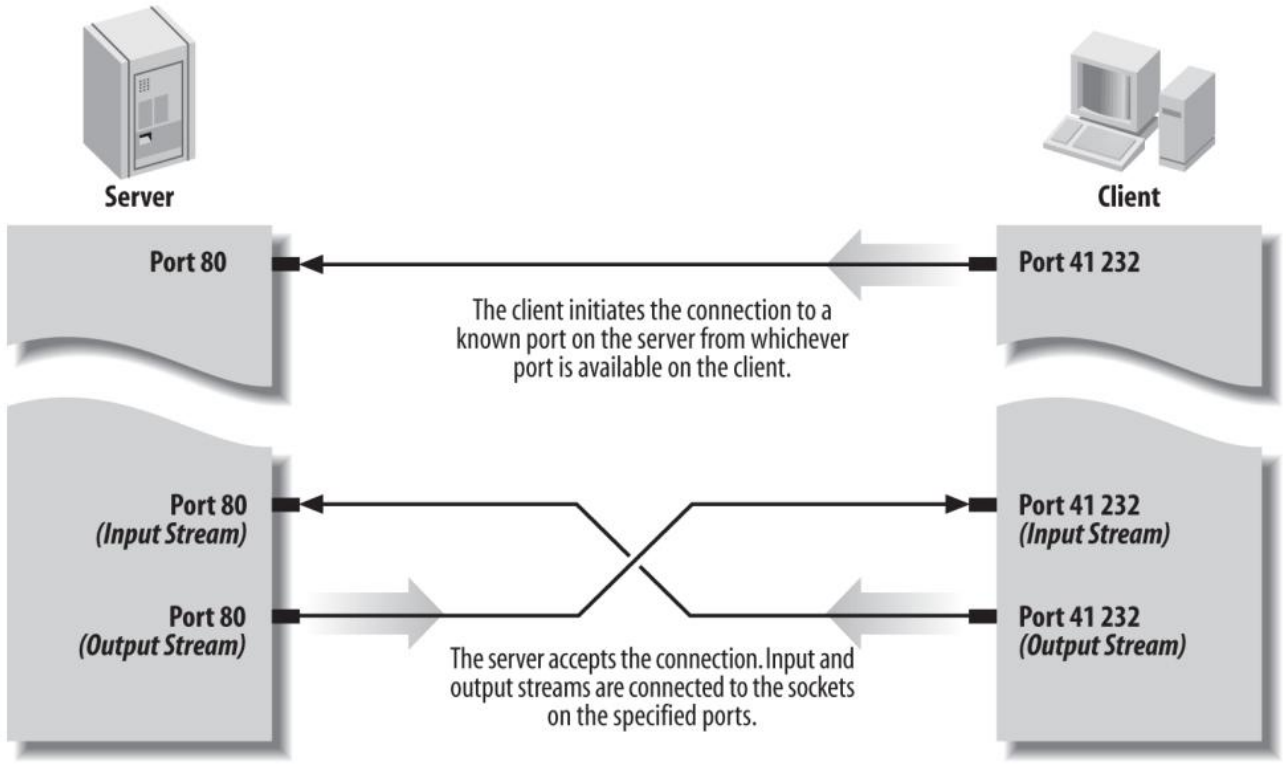

Figure 1-5. A client/server connection

Internet Standards:

- Although there are many standards organizations worldwide, the two that produce most of the standards relevant to application layer network programming and protocols are the Internet Engineering Task Force (IETF) and the World Wide Web Consortium (W3C).
- The IETF is a relatively informal, democratic body open to participation by any interested party. Its standards are based on “rough consensus and running code” and tend to follow rather than lead implementations.
- IETF standards include TCP/IP, MIME, and SMTP.
- The W3C, by contrast, is a vendor organization controlled by dues-paying member corporations that explicitly excludes individual participation.
- For the most part, the W3C tries to define standards before implementation.
- W3C standards include HTTP, HTML, and XML

IETF RFCs :

- IETF standards and near-standards are published as Requests for Comments (RFCs). Despite the name, a published RFC is a finished work. It may be obsoleted or replaced by a new RFC, but it will not be changed.
- IETF working documents subject to revision and open for development are called “Internet drafts.” RFCs range from informational documents of general interest to detailed specifications of standard Internet protocols such as FTP.
- RFCs are available from many online locations, including http://www.faqs.org/rfc/ and http://www.ietf.org/rfc.html. For the most part, RFCs (particularly standards-oriented RFCs) are very technical, turgid, and nearly incomprehensible.
- Nonetheless, they are often the only complete and reliable source of information about a particular protocol. Most proposals for an RFC begin when a person or group gets an idea and builds a prototype.
- The prototype is significant. Before something can become an IETF standard, it must exist and work. This requirement ensures that IETF standards are at least feasible, unlike those promulgated by other organizations.

Table 1-2 lists the RFCs that provide formal documentation for the protocols discussed in this book.

Table 1-2. Selected Internet RFCs

RFC	Title	Description
RFC 5000	Internet Official Protocol Standards	Describes the standardization process and the current status of the different Internet protocols. Periodically updated in new RFCs.
RFC 1122 RFC 1123	Host Requirements	Documents the protocols that must be supported by all Internet hosts at different layers (data link layer, IP layer, transport layer, and application layer).
RFC 791, RFC 919, RFC 922, RFC 950	Internet Protocol	The IP internet layer protocol.
RFC 768	User Datagram Protocol	An unreliable, connectionless transport layer protocol.
RFC 792	Internet Control Message Protocol (ICMP)	An internet layer protocol that uses raw IP datagrams but is not supported by Java. Its most familiar uses are *ping* and *traceroute*.
RFC 793	Transmission Control Protocol	A reliable, connection-oriented, streaming transport layer protocol.

RFC	Title	Description
RFC 2821	Simple Mail Transfer Protocol	The application layer protocol by which one host transfers email to another host. This standard doesn't say anything about email user interfaces; it covers the mechanism for passing email from one computer to another.
RFC 822	Format of Electronic Mail Messages	The basic syntax for ASCII text email messages. MIME is designed to extend this to support binary data while ensuring that the messages transferred still conform to this standard.
RFC 854, RFC 855	Telnet Protocol	An application layer remote login service for command-line environments based around an abstract network virtual terminal (NVT) and TCP.
RFC 862	Echo Protocol	An application layer protocol that echoes back all data it receives over both TCP and UDP; useful as a debugging tool.
RFC 863	Discard Protocol	An application layer protocol that receives packets of data over both TCP and UDP and sends no response to the client; useful as a debugging tool.
RFC 864	Character Generator Protocol	An application layer protocol that sends an indefinite sequence of ASCII characters to any client that connects over either TCP or UDP; also useful as a debugging tool.
RFC 865	Quote of the Day	An application layer protocol that returns a quotation to any user who connects over either TCP or UDP and then closes the connection.
RFC 867	Daytime Protocol	An application layer protocol that sends a human-readable ASCII string indicating the current date and time at the server to any client that connects over TCP or UDP. This contrasts with the various NTP and Time Server protocols, which do not return data that can be easily read by humans.
RFC 868	Time Protocol	An application layer protocol that sends the time in seconds since midnight, January 1, 1900, to a client connecting over TCP or UDP. The time is sent as a machine-readable, 32-bit unsigned integer. The standard is incomplete in that it does not specify how the integer is encoded in 32 bits, but in practice a big-endian integer is used.

RFC 959	File Transfer Protocol	An optionally authenticated, two-socket application layer protocol for file transfer that uses TCP.
RFC 977	Network News Transfer Protocol	The application layer protocol by which Usenet news is transferred from machine to machine over TCP; used by both news clients talking to news servers and news servers talking to each other.
RFC 1034, RFC 1035	Domain Name System	The collection of distributed software by which hostnames that human beings can remember, like www.oreilly.com, are translated into numbers that computers can understand, like 198.112.208.11. This RFC defines how domain name servers on different hosts communicate with each other using UDP.
RFC 1112	Host Extensions for IP Multicasting	The internet layer methods by which conforming systems can direct a single packet of data to multiple hosts. This is called multicasting. Java's support for multicasting is discussed in Chapter 13.
RFC 1288	Finger Protocol	An application layer protocol for requesting information about a user at a remote site. It can be a security risk.
RFC 1305	Network Time Protocol (Version 3)	A more precise application layer protocol for synchronizing clocks between systems that attempts to account for network latency.
RFC 1939	Post Office Protocol, Version 3	An application layer protocol used by sporadically connected email clients such as Eudora to retrieve mail from a server over TCP.

RFC	Title	Description
RFC 1945	Hypertext Transfer Protocol (HTTP 1.0)	Version 1.0 of the application layer protocol used by web browsers talking to web servers over TCP; developed by the W3C rather than the IETF.
RFC 2045, RFC 2046, RFC 2047	Multipurpose Internet Mail Extensions	A means of encoding binary data and non-ASCII text for transmission through Internet email and other ASCII-oriented protocols.
RFC 2141	Uniform Resource Names (URN) Syntax	Similar to URLs but intended to refer to actual resources in a persistent fashion rather than the transient location of those resources.
RFC 2616	Hypertext Transfer Protocol (HTTP 1.1)	Version 1.1 of the application layer protocol used by web browsers talking to web servers over TCP.
RFC 2373	IP Version 6 Addressing Architecture	The format and meaning of IPv6 addresses.
RFC 3501	Internet Message Access Protocol Version 4rev1	A protocol for remotely accessing a mailbox stored on a server including downloading messages, deleting messages, and moving messages into and out of different folders.
RFC 3986	Uniform Resource Identifiers (URI): Generic Syntax	Similar to URLs but cut a broader path. For instance, ISBN numbers may be URIs even if the book cannot be retrieved over the Internet.
RFC 3987	Internationalized Resource Identifiers (IRIs)	URIs that can contain non-ASCII characters.