

Computer Science for Internauts—How E-mail and Other Internet Services Actually Work

By Naomi James Sutcliffe de Moraes

Just Right Communications Ltda.

I will begin with a theoretical introduction to computer networks and how they function. I will then use this framework to describe how an e-mail system works. The information presented here should help both the technical translator of networking texts and all other translators, especially when their e-mail and web services are down or when trying to decide which service provider to contract with.

The OSI Reference Model

The Open Systems Interconnection (OSI) reference model roughly describes most networks (a schematic is shown in Figure 1). The message to be sent from PC1 to PC2 begins at the application layer. It is then compressed, broken up into smaller pieces, or encapsulated (put into an electronic envelope) at each subsequent layer until it arrives at the physical layer. After being transferred to the other computer, the layers of data comprising the message are peeled (like an onion) and converted back into a format the receiving application can understand.

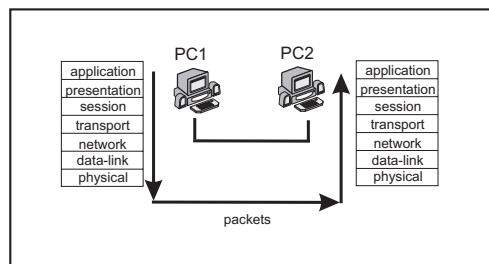


Figure 1: The OSI Reference Model

Let us look at an example. PC1 is being used to send e-mail. The e-mail client (Eudora, for example) creates a unit of data called an e-mail packet (**application layer**), compresses it (**presentation layer**), and requests a connection with the Simple Mail Transport Protocol (SMTP) server at the Internet provider (**session layer**). The session layer establishes connections between applications on different computers (like when a web page is requested or e-mail is up/downloaded). The **transport layer** is where a specific port is set up (a port is the channel through which data is sent). Each standard application has a well-known port: HTTP, 80; POP3, 110; FTP, 21. The destination Internet Provider is recognized at the application layer, and the topology of the network is recognized at the **network layer**. This is where routing decisions are made. Routing is how packets get from one Local Area Network (LAN) to another. The packet is sent to a gateway (usually a router) if the destination is outside the LAN (see Figure 2), or directly to the destination PC if the destination is on the same LAN. At the **data-link layer**, only the sending and receiving PCs on a LAN are involved, regardless of how many devices are actually present. At this layer, the Media Access Control (MAC) address is used. The MAC address is a number assigned to each device, valid only *inside* a local network, and is different from the IP address (discussed later). Routers are the key here, as they are officially on two or more LANs simultaneously and serve as stepping stones for the packets to hop from one LAN to another. The **physical layer** deals with the type of medium used (the type of cable, for example).

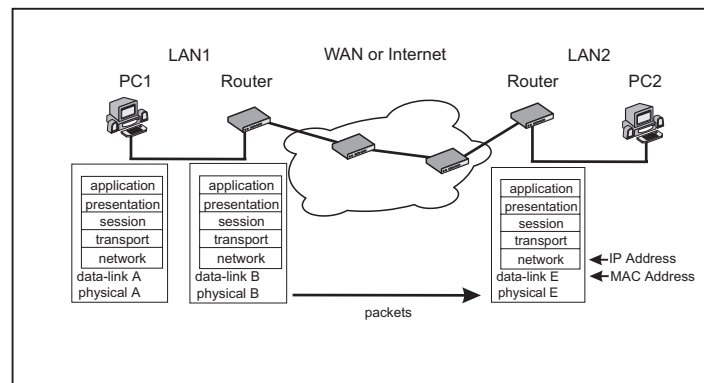


Figure 2: Routing and the OSI Reference Model

As mentioned before, the OSI reference model is like an onion, containing many layers of data constructed by the sender. During the message delivery process, these layers are peeled away as necessary and converted back into a format the receiving application can understand. This process ensures that a packet will reach its destination. For example, the MAC address is valid only within a local network and is used by the data-link layer. Each time a packet moves from one LAN to another, it must have its first two layers changed by the equipment operating on it (a router, for example). In Figure 2, this conversion process can be seen taking place as the packet passes from PC1 to the first router. To get the packet to the router (in this case, a gateway), the packet contains the router's MAC address in the data-link layer envelope. The router peels off the lowest two layers to read the destination IP address (network layer information). It then determines the next router on the path to the destination and places the MAC address of this next router in the packet and reconstructs it. Why are two layers necessary? One determines the type of cable and standard used (V.35, X.21, etc.), while the other determines the type of data-link (HDLC, PPP, Ethernet, Frame Relay, X.25, etc.). LANs normally use Ethernet, whereas Wide Area Networks use the Point-to-Point Protocol (PPP) or other options. Some protocol combinations do not fit exactly into the OSI reference model (Asynchronous Transfer Mode, for example) and either more or fewer layers must be peeled off at each step.

I will not discuss routing here except to say that there are various protocols (RIP, OSPF, and BGP-4) in use, and each router keeps its own table of neighboring routers. Each time a packet comes through for a new destination, the router asks around (kind of like gossip). If it receives an answer from a neighboring router, it updates its table and sends the packet to the indicated next router; if not, it sends the packet to the default gateway. Run *winipcfg* on your IBM-compatible PC (while connected to a network) to see your computer's IP address and the default gateway router at your access provider. Any packets not destined to the access provider itself (an email to technical support, for example) will be forwarded to this gateway router.

OSI Layer	Typical Protocols
Application/Presentation/ Session	ftp (file transfer), telnet, ssh, SMTP (e-mail), http (web)
Transport	TCP, UDP, SPX
Network	IP, IPX

Data-link	HDLC, PPP, Frame Relay, X.25, Ethernet
Physical	Ethernet 802.3, RS-232, V.35, X.21

Figure 3: Protocols Associated with Each OSI Layer

Figure 3 gives a sampling of acronyms you may already have seen, along with the OSI layer they represent. I will not indicate what each acronym stands for, as the names are unnecessary for this discussion. Please see the section on “Further Reading” at the end of this article for more information on acronyms.

Some protocols are interoperable and some are not. When transferring a packet from a TCP/IP network to a SPX/IPX (Novell) network, all the layers up to the network layer will have to be stripped off the packet and replaced with values appropriate to the receiving network. It is amazing the Internet even works!

Routing and the Domain Name System

Routing was discussed briefly above. On a network using the IP protocol, routing is performed using an IP address. An IP address is a 32-bit number generally written as four 8-bit numbers (for example, 200.241.120.25). IP addresses are geographically distributed, though I do not know if there is a rule for their distribution. My web site (IP address 216.234.189.104) is based in Edmonton, Alberta, Canada. The site <http://where-is.info/> claims to be able to identify the country for any IP address, and correctly identified that my site is in Canada.

So, when someone types www.justrightcommunications.com in their browser, how does the request packet reach its destination? First, the application looks in a local file to see if some application has requested this domain name recently. If so, the IP address will still be registered in the file (similar to the routing table mentioned above). Most entries in these types of tables have a self-destruct time delay. If a site is not accessed for a certain amount of time, its entry will be removed from the local table. Next, the application will send a request to the Domain Name System (DNS) server (also visible by running *winipcfg* and choosing “additional options”). DNS is the largest distributed database in the world. If the access provider’s DNS server does not know the IP address for the domain, it will send the request forward. After an IP address is determined and the request packet zipped up in all its layers, routing proceeds as described in the previous section.

To be registered, every site must have two DNS servers always ready to respond to requests about its domain name/IP address. These three DNS servers are indicated in the “whois” information (see www.whois.net), which is also useful to discover who owns a site. For my site, for example, they are: ns1.tera-byte.com (IP address 216.234.161.11); ns2.tera-byte.com (IP address 216.234.161.12); and a spare ns3.tera-byte.com (IP address 204.209.56.2). A bit of investigation shows that www.tera-byte.com is the web site of my Internet service provider (my web hosting company).

Internet Access and Service Providers

There are two kinds of Internet providers: access providers and service providers. Access providers allow you to dial in to a network (either with a modem or other device), and service providers provide hosting, e-mail, and perhaps content (a newspaper or other text, for example).

Access providers frequently offer e-mail services as well, though they are not necessarily the best option for translators. If you change access providers (due to poor service, elevated prices, or other reasons), the e-mail address is lost. Few providers forward mail, even for a fee. This is a good argument for purchasing your own domain name! That way, you have a permanent e-mail address and control over how you send and receive e-mail.

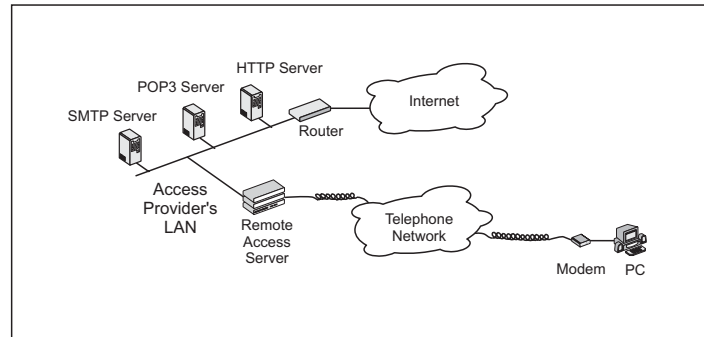


Figure 4: Local Area Network of an Internet Access Provider with Some Services

In Figure 4, a minimal Internet access provider is shown. The paying user calls in using a modem and is connected to the provider's LAN. From there, the Internet can be reached through the provider's gateway router. The provider may or may not have SMTP, POP3, and HTTP servers. Let me explain what these are for.

E-mail and other Internet applications use the client/server model. Your e-mail programs (Eudora, Pegasus, Netscape Mail, Outlook Express) are e-mail clients (also known as User Agents) and are located on your PC. They interact with SMTP servers (also called Mail Transfer Agents) to send mail and POP3 servers (also called Mail Box Managers) to receive mail. These two servers are not related in any way. The SMTP and POP3 server programs may be located on the same physical server (a very small service provider) or in different countries. In my case, I use my Canadian service provider's POP3 server to receive mail and my Brazilian access provider's SMTP server to send mail (using an outgoing email address using the domain name I purchased).

An HTTP (HyperText Transfer Protocol) server provides content. My web site is on a computer with an HTTP server program. When someone wants to see my site, the HTTP server accepts the connection and responds with the appropriate HTML file.

This division of services into separate servers means that you do not need a site in order to have an e-mail address with your own domain name. Some service providers offer e-mail-only packages. They will host your domain (including a POP3 server and two DNS servers) without providing HTTP server services. Simple! If you later decide to create a site, just upgrade.

A drawback of the division of services is that e-mail may not actually be coming from its apparent destination. Returning to the OSI model, the recipient's e-mail address is converted into an IP address when the e-mail packet is constructed (in the application layer). The sender's e-mail address just goes along for the ride. It is (usually) read only when the packet is peeled by the e-mail client at the destination. An exception is my ex-access provider. I sent e-mails with my domain e-mail address (nmoraes@justrightcommunications.com) as the return address for over a year, until one day they started peeling the packet layers off (up to the application level!) and let through only those e-mail packets with their domain e-mail address (xxx@uol.com.br).

No warning, no error messages. My e-mail just disappeared! Next, I moved to an access provider who bases outgoing e-mail acceptance on my IP address (I have a dedicated connection and therefore a dedicated IP address). They peel the packet to the network level and make sure my IP address is on their network. These safeguards are necessary (though not always used) to avoid spam. The return e-mail addresses on spam are almost always false.

For those who are worried about the security problems related to having 24-hour access to the Internet, keep in mind that any client/server interaction must be initiated on the client's side. Most PCs (Linux and Windows NT excluded) have no server programs! A virus that acts like a client program would have to install itself on your computer and connect (without your permission) to a server somewhere else to transfer your data/passwords through the connection. For added protection, you could install a firewall, a hardware device or software program that protects networked computers from intentional hostile intrusion that could compromise confidentiality or result in data corruption or denial of service. The term "firewall" comes from the fact that by segmenting a network into different physical sub-networks, firewalls limit the damage that could spread from one sub-net to another—just like firewalls. To learn more about this, see the information about firewalls and client security at www.symantec.com.

E-mail: Options and Client Programs

Now that I have described how e-mail is routed (or not!), I will describe the options of various popular e-mail programs. The two most common are Netscape Mail and Outlook Express. Many people become unsatisfied with these and end up purchasing Eudora. I use Pegasus (www.pmail.com), a free alternative to Eudora which seems to have the same options. I could not get Eudora to install on my computer, so I tried Pegasus and have never looked back. If you are still using Outlook, try Eudora or Pegasus for a week to see the benefits. Each e-mail client implements a different subset of the available functions. They each have different sorting capabilities, address book features, and spell-check options. Some even allow you to delete e-mails without having to read them first.

When sending e-mail, most of the options (blind carbon copy (BCC), return receipt, priority) are coded into the packet at the application level. (The priority, for example, is ignored at the network level where packets are routed). If you look at the text of an e-mail (hidden by most e-mail clients), the header will contain the information in Figure 5. To see this information, choose "raw view" in Pegasus, "view->message source" in Netscape Mail, or something similar in Outlook Express or Eudora.

```

Return-Path: <email address of sender>
Received: from intermediate router1 (server/domain name [IP address])
by POP3 server (server/domain name [IP address])
for <email address of recipient>; Date and Time
Received: from intermediate router1(server/domain name [IP address])
by SMTP server (server/domain name [IP address])
for <email address of recipient>; Date and Time
From: = name of sender (according to his/her email client, could be false)
Subject: text provided by sender
X-Confirm-Reading-To: email address of sender
X-Priority: Level
Date: Date and Time
User-Agent: Name of email client used by the sender
To: email address of recipient
●●●

```

Figure 5: Raw View of an E-mail Message

The received fields indicate the path which the e-mail took from the sender's SMTP server to the recipient's POP3 server. These are not the routers, but rather intermediate e-mail servers. The field "X-Priority" (the priority of the e-mail: Urgent, Normal, or Bulk) is not used by all client programs. The "X-Confirm-Reading-To:" is a nice feature of Pegasus that, unfortunately, is not recognized by all receiving e-mail clients. When an e-mail with this option is received, the user is asked if they would mind sending a return receipt. The sender can then find out if the recipient (a translation client?) received the e-mail, even if no formal reply has been sent. This option could save you from having to make an international phone call to confirm the receipt of your message. Pegasus can send return-receipt e-mail and process it when it is sent by others. I believe Outlook Express and Netscape can process incoming return-receipt messages. The return-receipt is just an automatic e-mail stating that the e-mail was opened, with a date and time.

Let me end with a quick discussion of BCC (blind carbon copy) etiquette. There are three ways to address an e-mail to a recipient: TO, CC (carbon copy), and BCC. CC is normally used when the e-mail is directed at the person designated in the TO field (and that person is expected to reply), and the CCed person is included for information purposes only. BCC is used when the people to whom the e-mail is directed are not supposed to know that the e-mail is being sent to another person. For example, suppose I receive a really nice job offer and respond saying, "Yes, I'm available." Instead of writing another e-mail to my husband to tell him we have to cancel our romantic dinner plans, I can just put him in the BCC field. That way, he has the news and my perspective client does not know he is receiving a copy of the message. Another use for BCC is when various recipients should not have each other's addresses. An example is when a translation agency sends out a message to several translators to see who is available, or sends out a job for a quote. BCC is very appropriate here, and the sender's address should be used in the TO field. For those who like to send Internet jokes to all their friends, BCC should also be used in case the message is forwarded five times and the addresses fall into the wrong hands. And for all the translators who have sent me their CVs as a CC and CCed 100 other agencies in the same e-mail: stop! This gives a very bad impression!

Further Reading

When confronted with an unknown networking term, your first stop should be <http://whatis.techtarget.com/>. There are no translations, but many explanations and links. For a nuts-and-bolts lesson, try www.howstuffworks.com/web-server.htm. Technical translators who wish to keep on top of where the Internet is going can visit www.isoc.org and www.w3.org/Consortium/.

References

Kurose, James F. and Keith W. Ross. 2000. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley.

Cyclades Brasil. 1999. *Guia Internet de Conectividade*. Cyclades Brasil, São Paulo.