Understanding TCP/IP

We conclude our four-part article looking in depth at the TCP/IP protocol. Here, we examine the difference between the SMTP and POP3 email protocols.

By Julian Moss

In this series of articles we have looked at the TCP / IP suite of protocols, beginning with the link layer and progressing by stages to the application layer. We have seen how each layer relies upon the layers below it, so that network applications can be written without needing to take account of considerations such as how the network is constructed or what type of hardware or cabling is used.

A striking point about many of the application layer protocols is how simple they are. The protocols based on TCP mostly use commands and responses in plain ASCII text, making them easier for a user to understand and for a programmer to implement. For further illustration we shall look at the two protocols that you may use every day to send and receive Internet email: SMTP and POP3.

SMTP

Simple Mail Transfer Protocol (SMTP) is one of the most venerable of the Internet protocols. Designed in the early 1980s, its function is purely and simply to transfer electronic mail across and between networks and other transport systems. As such, its use need not be restricted to systems that use TCP / IP. Any communications system capable of handling lines of up to 1,000 7-bit ASCII characters could be used to carry messages using SMTP. On a TCP / IP network, however, TCP provides the transport mechanism.

In SMTP the sender is the client, but a client may communicate with many different servers. Mail can be sent directly from the sending host to the receiving host, requiring a separate TCP connection to be made for each copy of each message. However, few mail recipients run their own SMTP servers.

It is more usual for the destination of an SMTP message to be a server that serves a group of users such as all those int one domain. The server receives all mail intended for its users and then allows them to collect it using POP3 (Post Office Protocol version 3) or some other mail protocol. Similarly, most SMTP clients send messages to a single server, whose job it is to relay those messages on to their eventual recipients.

An SMTP transaction begins when the sender client opens a TCP connection with the receiver using the well-known port number 25. The server acknowledges the connection by sending back a message of the form “220 SMTP Server Ready”. SMTP uses a similar format of replies to ftp, which we looked at previously. The three-digit code is all the client software needs to tell if everything is going OK. The text is there to help the humans who might be troubleshooting a problem by analysing a log of the transaction. The box “Application Protocol Reply Codes” provides more information about message reply codes.

An SMTP relay server might refuse a connection by sending back a message with a “421 Service not available” reply code. For example, an Internet Service Provider’s SMTP server provided for use by its subscribers to relay outgoing mail might refuse a connection from a host whose IP address indicates that it is not a subscriber to that ISP. SMTP has no form of access control - the way it can be used to relay messages would make this impractical - so this is about the only way ISPs can prevent non-subscribers such as spammers from using their mail servers to send out messages.

Having received the correct acknowledgement the sender signs on to the server by sending the string “HELO hostname”. HELO is the sign-on command and hostname is the name of the host. As we will see, the hostname is used in the Received: header which the server adds to the message when it sends it on its way. This information allows the recipient to trace the path taken by the message.

Sending

Once the sender gets a “250 OK” acknowledgement it can start sending messages. The protocol is extremely simple. All the sender has to do is say who the message is from, who it is to, and supply the contents of the message.

Who a message is from is specified with the command “MAIL FROM: <address>”. This command also tells the receiver that it is about to receive a new message, so it knows to clear out its list of recipients. The address in the angle brackets (which are required) is the return path for the message. The return path is the address that any error report - such as would be generated if the message is undeliverable - is sent to.

“SMTP uses a similar format of replies to ftp, which we looked at previously. The three- digit code is all the client software needs.”

It is valid for the return path to be null, as in "MAIL FROM: <>". This is typically used when sending an error report. A null return path means that no delivery failure report is required. Its main purpose is to avoid getting into the situation in which delivery failure messages continually shuttle back and forth because both sender and recipient addresses are unreachable.

The recipients of a message are defined using the command "RCPT TO: <address>". Each address is enclosed in angle brackets. A message may have many recipients, and an RCPT TO: command is sent for each one. It is the RCPT TO: command, not anything in the message headers, that results in a message arriving at its destination. In the case of blind carbon copies or list server messages the recipient address will not appear in the headers at all.

"The return path is the address that any error report - such as would be generated if the message is undeliverable - is sent to."

Each recipient is acknowledged with a "250 OK" reply. A recipient may also be rejected using a reply with a 550 reply code. This depends on how the server has been configured. Dial-up ISP SMTP relay servers may accept every RCPT TO: command, even if the address specified is invalid, because the server doesn't know that the address is invalid until it does a DNS lookup on it. However, a mail server intended to receive messages for local users only would reject recipients that aren't at that domain.

Other replies may be received in response to RCPT TO: messages as a result of the SMTP server being helpful. If an address is incorrect but the server knows the correct address it could respond with "251 User not local; will forward to <address>" or "551 User not local; please try <address>". Note the different reply codes signifying whether the server has routed the message or not. These replies aren't common, and a mail client may simply treat the 551 response as an error, rather than try to parse the alternative address out of the reply text.

For the sake of completeness it should be pointed out that RCPT TO: commands may specify routes, not merely addresses. A route would be expressed in the form "RCPT TO: <server1,server2:someone@server3>". Today this capability is rarely needed.

Application Protocol Reply Codes

Many Internet application layer protocols which are based on ASCII text commands use a system of replies in which an initial three-digit code provides the essential status information. Each digit has a particular meaning, as shown below.

First Digit

1xx: Positive Preliminary Reply. Command accepted but held awaiting a further confirmation command (continue or abort).

2xx: Positive Completion Reply. Command completed. Awaiting next command.

3xx: Positive Intermediate Reply. Command accepted but held awaiting further information (such as a password).

4xx: Transient Negative Completion Reply. Command not accepted due to a temporary error condition (such as an HTTP server busy). The command may be tried again later.

5xx: Permanent Negative Completion Reply. Command not accepted due to a permanent error condition. The command is unlikely to be accepted if repeated later.

Second Digit

x0x: Syntax Error. For example, command unimplemented or valid but incorrect in the circumstances.

x1x: Information. The text following the code contains the answer to an information request.

x2x: Connections. Message reply relates to the communications channel.

x5x: Server. Message reply relates to the state of the server.

Third Digit

Used to distinguish individual messages.

Message Text

Once all the recipients have been specified, all that remains is for the sender to send the message itself. First it sends the command "DATA", and then waits for a reply like: "354 Start mail input; end with <CRLF>.<CRLF>". The message is then sent as a succession of lines of text. No acknowledgement is received for each line, though the sender needs to watch for a reply that indicates an error condition.

The end of the message is, as indicated by the reply shown above, a period (full stop) on a line of its own. Thus, one of the simplest but most essential things that a mail client must do is ensure that a line containing a single period does not appear in the actual text.

TCP/IP

The end of the message is acknowledged with "250 OK".

It's worth noting that SMTP isn't in the least bit interested in the content of the message. It could be absolutely anything, though strictly speaking it should not contain any characters with ASCII values in the range 128 to 255, and lines of text may not exceed 1,000 characters. There is no requirement for the headers to show the same sender and recipient addresses that were used in the SMTP commands, which makes it easy to make a message appear to have come from someone other than the true sender.

Tracking

When a message is relayed by the server it inserts a "Received:" header at the start of the message showing the identity of the host that sent the message, its own host name, and a time stamp. Each SMTP server that a message passes through adds its own "Received:" header. Thus it is possible to track the path taken by a message. Although this won't identify the sender it may shed some light on whether or not the address the message is apparently from is in fact the true one.

After the "250 OK" that acknowledges the end of the message, the sender can start again with a new message by sending a new "MAIL FROM:" command or it can sign off from the server using "QUIT". A 221 reply will be received in response to the QUIT command.

SMTP servers should support two further commands for a minimum implementation. NOOP does nothing, but should provoke a "250 OK" reply. RSET aborts the current message transaction. There are other commands such as HELP which are really only of interest to those trying to communicate with SMTP servers interactively and are therefore not really relevant to understanding how the protocol works in day-to-day use.

POP3

SMTP is capable of delivering mail direct to the recipient's desktop, but in practice it isn't the ideal protocol for this. If an SMTP relay is unable to deliver a message to the next (or final) host in the chain, it will try at ever-lengthening intervals over a period of a few days before giving up and sending a delivery failure notification to the return path address.

SMTP offers no way for the recipient to prompt a server into sending mail that it is trying to deliver. If a recipient connects to the Internet infrequently their server may never be active at the right time. In this case the mail will eventually bounce.

SMTP is rather like a courier delivery service. If you aren't in when it calls then, after a couple of re-delivery attempts, the message is returned to the sender. Post Office Protocol version 3 (POP3) - as the name suggests - lets you have your mail held at the post office so you can collect it at a time of your own choosing.

POP3 is another TCP application, and uses the well-known port number 110. As with the other text-based application protocols you can connect with a POP3 server using a Telnet terminal emulator and interact with it using POP3 commands. This can sometimes be useful, as for example to manually delete a corrupt message that crashes a mail client whenever it is downloaded. (However, don't try connecting to your ISP's port 110 and sending random commands without permission. Their automatic hacker detection systems might spring into operation and you may well be asked to explain what you're doing.)

On connecting to the server, the server should respond with the message "+OK POP3 server ready". POP3 uses "+OK" and "-ERR" at the start of replies to indicate acceptance or rejection of commands. This is simpler than the numeric codes used by SMTP and other protocols: software need only check the first character for a plus or a minus. The text that may appear after a "+OK" is a prompt for what to do next. After "-ERR" it is an error description. The exact content of the text may vary between server implementations.

"As with the other text-based application protocols you can connect with a POP3 server using a Telnet terminal emulator and interact with it using POP3 commands."

To Access The Server

A POP3 server holds people's personal mail, so unsurprisingly you need to enter a user name and a matching password before you can gain access to it. To log in you must send "USER username". A "+OK" response shows that the user name is valid. You must then send "PASS password". If the password is correct you will receive another positive acknowledgement in a reply like "+OK username has two message(s) (914 octets)". "-ERR" replies may be received if the user name is not known, the password is incorrect or the server is for some reason unable to open a user's mailbox.

Once a client is successfully logged in it can issue several different commands which allow it to find out how many messages are waiting and how big they are, and to download the messages and delete them from the server.

The "STAT" command returns the number of messages waiting (mw) and their total size in bytes (sb), as a response in the form "+OK mw sb". Note that this is the same information given in the login acknowledgement, but in a form (two numbers separated by a single space) that is easier for the client software to process.

"SMTP is rather like a courier delivery service. If you aren't in when it calls then, after a couple of re-delivery attempts, the message is returned to the sender."

The command "LIST" can be used to determine the size of each message. After the "+OK" the server sends, on separate lines, the message numbers (mn) and the message sizes (ms) separated by a space. Waiting messages are numbered sequentially, starting from 1. The command "LIST mn" can be used to find out the size of a specific message. The LIST command is typically used by mail clients that implement a user-defined restriction on the size of messages that will be downloaded, or those that want to display a progress indicator that shows how much of each message has been downloaded.

POP3 provides no commands that enable a client to find out the subject of a message or who it is from. However, the TOP command lets the client download a message's headers and a specified number of lines from the message body, from which this information may be obtained. TOP is an optional POP3 command but its implementation is strongly recommended.

The format of the command is "TOP mn nl" where mn is the message number and nl the number of lines required. The response is "+OK" (if mn is valid) followed by a partial download of the message. The end of the download is indicated by a line containing a single period (full stop).

Some spam filtering software - which kills unwanted messages without downloading them - uses the TOP command to determine whether a message meets the criteria for being killed or not. However, the time taken to get this information for every message may exceed the time it would have taken simply to download the spam and delete it later.

The command "RETR mn" is used to retrieve messages from the server. The command must include a message number (mn). After an "+OK" acknowledgement the server sends the whole message. Again, the end of the message is indicated by a line containing just a period.

Wiping

The command "DELE mn" is used to delete a message. In fact, the DELE command only marks messages for deletion. Any messages marked for deletion during a session may be undeleted by issuing an "RSET" command. The messages are only deleted once the client has closed the POP3 session by issuing a "QUIT" command. If a client never gets to close a session properly because the connection is lost or timed out then you may find some messages being downloaded again the next time you connect to the server.

In order to avoid downloading messages twice, a POP3 client can use the command "UIDL" or "UIDL mn" to obtain unique, server-generated IDs for each message. By storing the UIDLs of downloaded messages in a file, a client can easily determine whether a message on the server has been previously retrieved or not.

Implementation of the UIDL command is optional, but most POP3 servers seem to support it and most mail clients use it.

Benefits

SMTP and POP3 are two of the most commonly-used Internet protocols, which is why we have devoted this article to looking at them in some detail. Their text-based nature, which makes it possible to send and receive messages by communicating with a server interactively using a simple Telnet client, also makes it easy to write client software using just about any programming language that can send and receive text using TCP.

This simplicity is in stark contrast to many other network architectures which require the use of proprietary APIs and languages that support complex data structures.

Conclusion

In this article it has only been possible to give an overview of the most important protocols used on the Internet. The full specifications of these and other Internet protocols can be found in Requests For Comments (RFCs) published by the Network Working Group. RFCs are freely available for download from the Internet. Anyone interested in finding out more about TCP/IP, and particularly in implementing their own TCP/IP applications, should obtain and study the RFCs for the protocols concerned.

However, even if you never have to write your own Internet software it is hoped that this article has piqued your interest, and contributed to a better understanding of how TCP/IP and the Internet really work.

PCNA

The Author

Julian Moss is a freelance writer and software developer with experience of developing TCP/IP client software. He can be contacted as jmoss@cix.co.uk.

Additional Resources

- IPv6 Explained
- The OSI 7 Layer Model Explained
- Understanding Frame Relay
- Understanding DHCP
- Virtual Private Networking Explained

All these articles are available free online now at
www.pcnetworkadvisor.com

PCNA

Recent Reviews from Tech Support Alert

Reviews of the Best Windows Backup Software

In this detailed comparative review, we checked out eighteen backup software utilities designed for home or SOHO use. Many of the products reviewed were disappointing. However 6 products passed our tests with flying colors and 2 of these were so impressive, they were awarded our "Editor's Choice."

Suppliers of Cheap Inkjet Printer Cartridges Reviewed and Rated

With hundreds of companies all claiming to have the *"cheapest and best inkjet printer cartridges*," our editors decided to put their claims to the test. Not unexpectedly, many suppliers flunked but we did manage to come up with a number of web sites that sell good quality inkjet printer cartridges at heavily discounted prices.

The Best Anti Trojan Software

Our editors took a close look at the 6 leading anti-trojan/trojan remover software utilities. Unfortunately, they found only 2 products that were effective in their ability to detect and remove dangerous modern polymorphic and process injecting trojans.

The 46 Best Ever Freeware Utilities

This is our Editor, Ian "Gizmo" Richards, personal selection of the best freeware utilities. He's hunted down some real gems, many of which perform better than expensive commercial products.