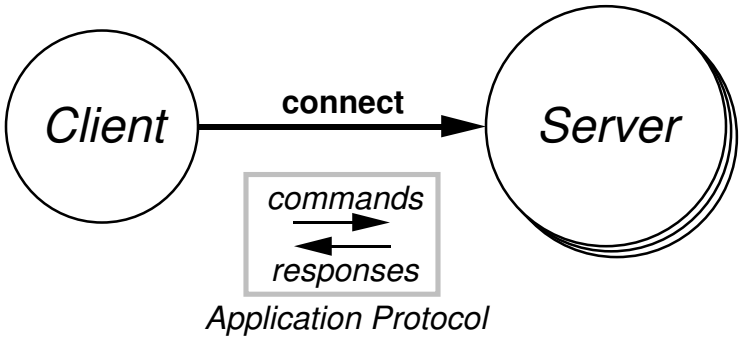
The Client–Server Model

- **Server**

  A central resource: a process which performs a specific task on request

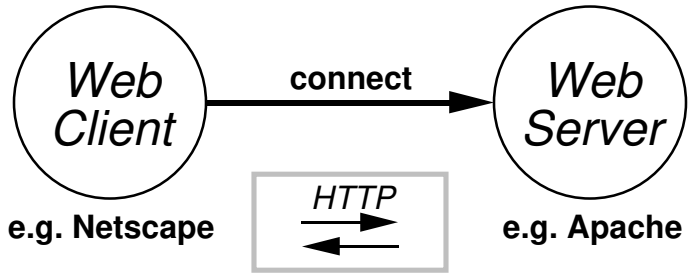
- **Client**

  A process which connects to a server and uses it

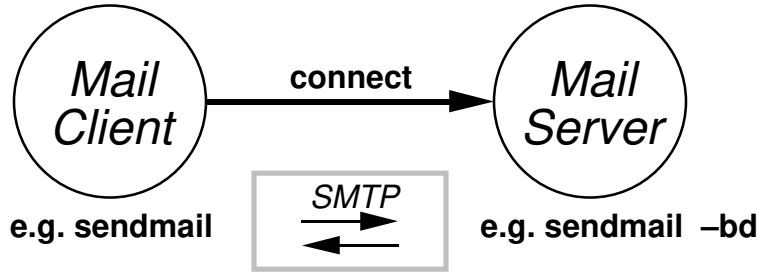


One server can appear to talk to many clients at once; in practice, separate server processes are spawned for each client

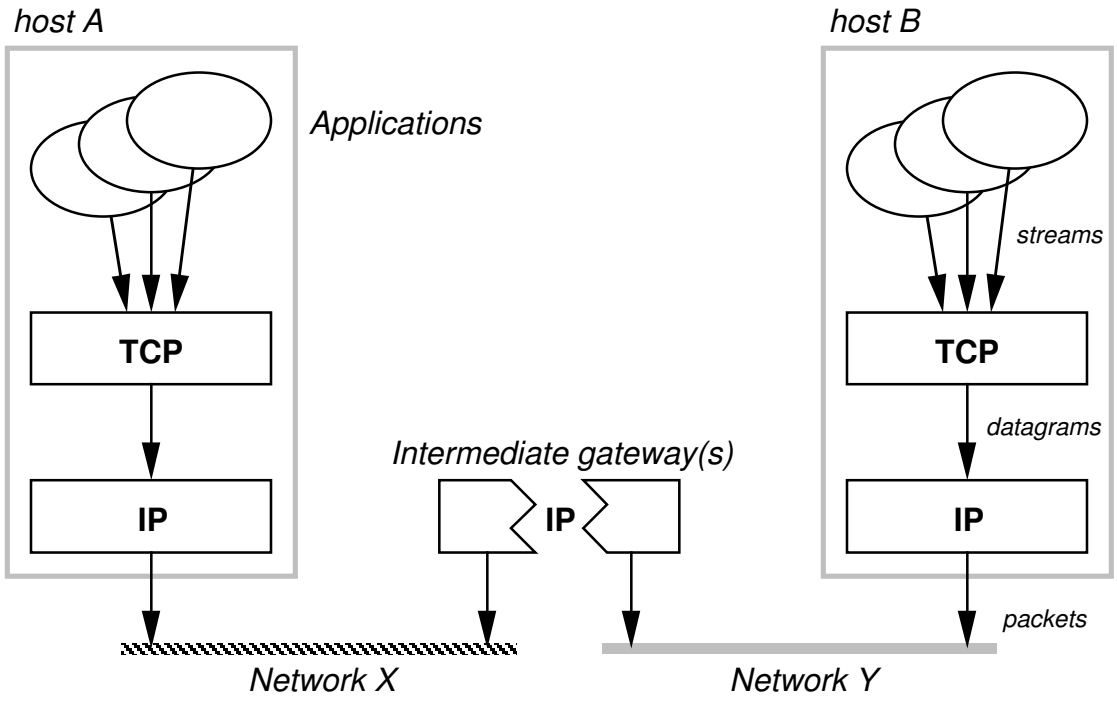
- **Example: World-Wide Web**



- **Example: E-mail delivery**



The TCP/IP protocol suite



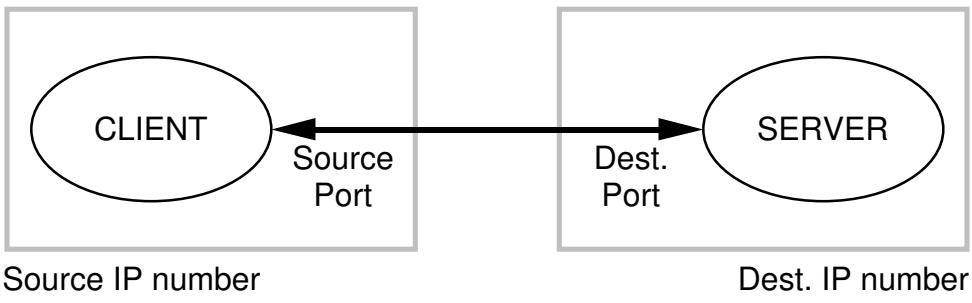
- **Hosts are identified by IP numbers** (32 bit)

  Although the user normally sees host names, these are converted to IP numbers by an application (DNS) before being used

- **Processes are identified by Port numbers** (16 bit)

  So multiple streams can be active simultaneously

  The Destination Port also identifies which server the client wishes to use. Ports 1–1023 are used for 'Well-known Services' – see /**etc**/**services** (Unix)



Examples of Well-Known Services (TCP)

	Destination Port #	Example server process
Telnet	23	*telnetd*
SMTP	25	*sendmail*
HTTP (Web)	80	*apache*
POP3	110	*pop3d*

- **How a client contacts a server**

  The client opens a TCP stream with the following parameters:

Source IP number	*Client's own IP number*
Destination IP number	*IP number of server host*
Source Port	*Any unused port (>1023)*
Destination Port	*Select server process required (see above)*

  These four numbers together uniquely identify the client-server connection.

  If there is no server 'listening' on that particular port, the connection will be refused.

- **Monitoring streams (Unix)**

netstat	*Show active sockets (≈ streams)*
netstat -n	*Ditto, without converting IP and port numbers to names*
netstat -a	*Also show "listening" ports*

Testing servers using ‘telnet’

- **The hacker’s tool**

  Telnet does little more than connect your keyboard and screen to a TCP stream. It is therefore ideal for testing TCP services.

- **Example: testing a Web server**

```
$ telnet www.isoc.org 80
```

host *port*

  If it connects, the server is responding.

  If you know the application protocol you can drive it directly. In this case it’s HTTP (the HyperText Transfer Protocol) and typing "GET /" will send you the home page

- **Example: testing a SMTP server**

```
$ telnet rg.net smtp
```

(You can use the name if it’s in /etc/services)

```
250 rg.net Smail3.1.28.1 #2 ready at [date]
HELO my.domain.com
250 rg.net Hello my.domain.com
EXPN randy
250 randy@rip.psg.com
QUIT
221 rg.net closing connection
```

  Many Internet protocols are designed for both machine use and human debugging (hence both numeric and textual results)