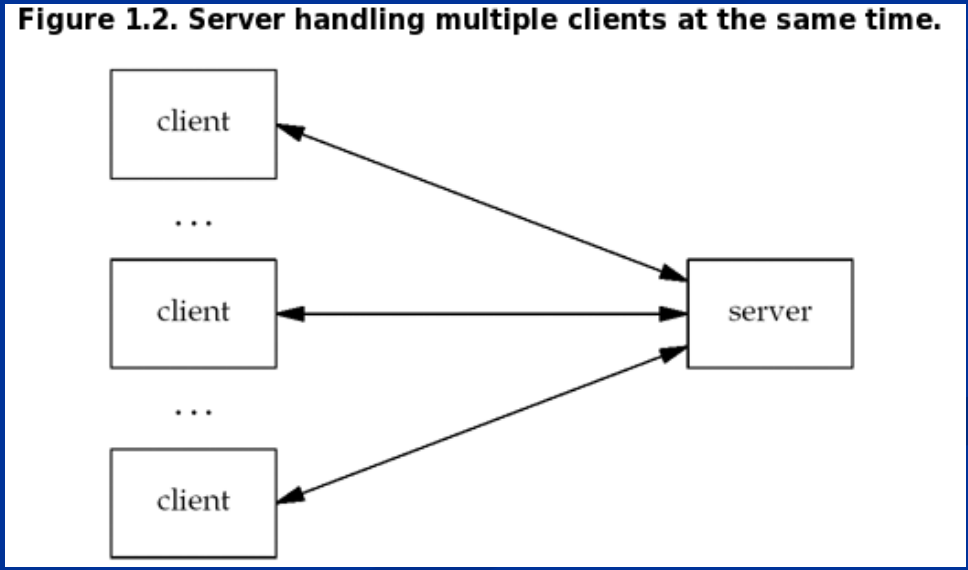
CST234
Chapter 1 - Introduction

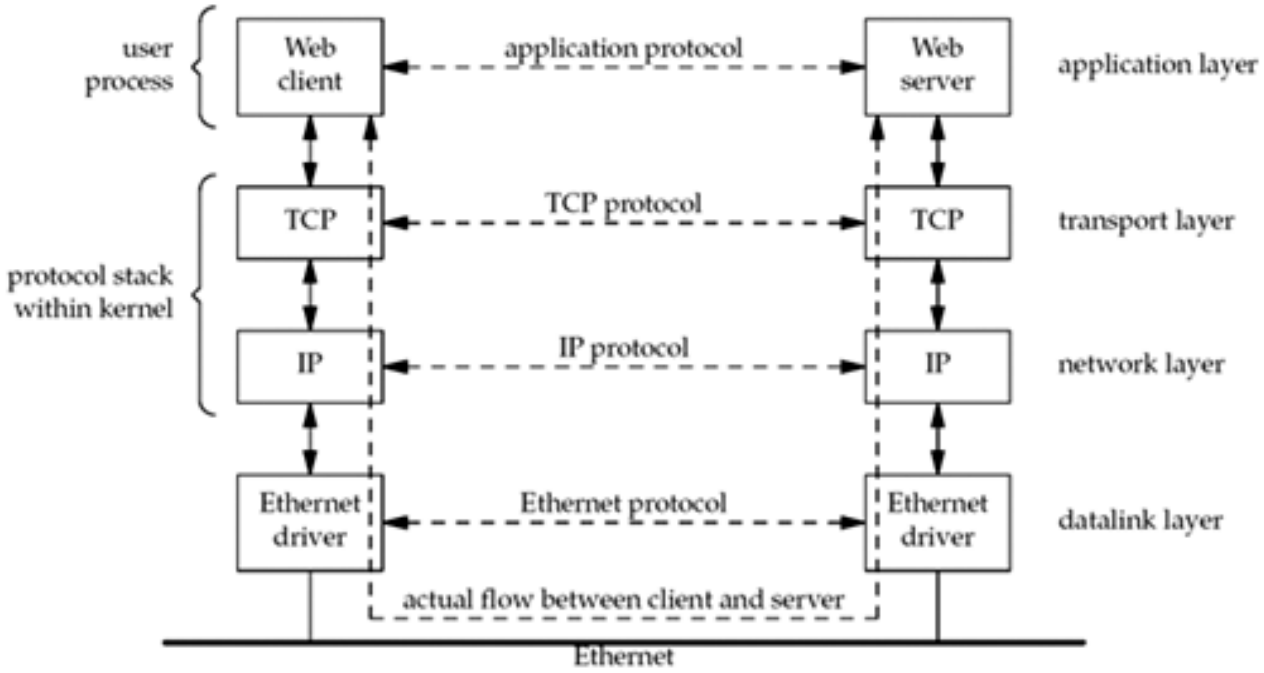
Network Application

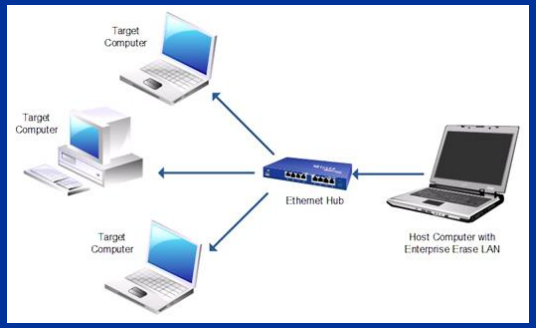
- Client application
  - seek out server to perform task
- Server application
  - long running programs (daemons) that service requests from one/more clients
- Protocol
  - an agreement on how programs communicate across a network
  - e.g. TCP/IP
- Client-server paradigm
  - multiple clients
  - single server



Client-Server on LAN

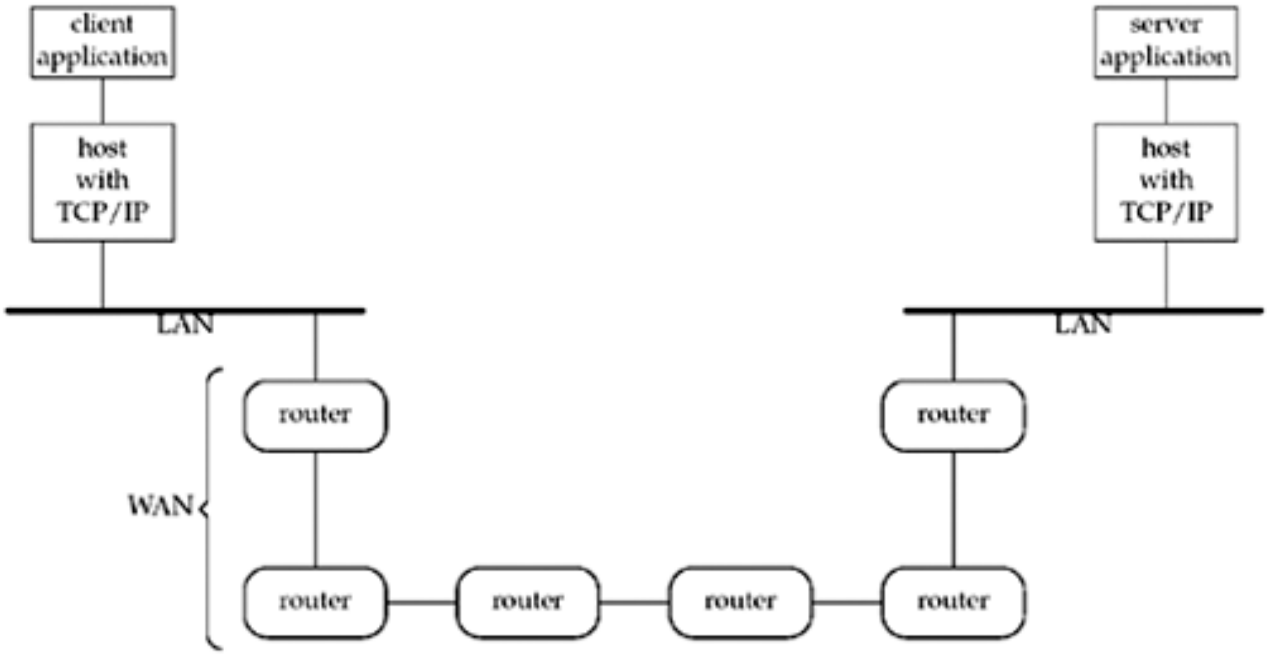
Figure 1.3. Client and server on the same Ethernet communicating using TCP.

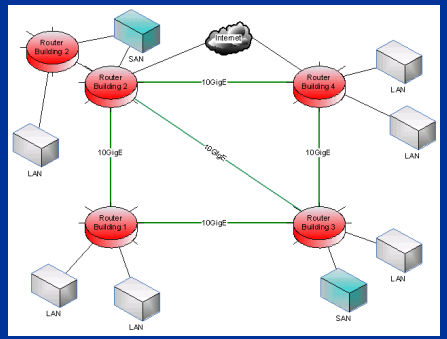




Client-Server on WAN

Figure 1.4. Client and server on different LANs connected through a WAN.





A Simple Daytime Client

- Create TCP socket (internet stream)
- Specify server's IP address and port
- Convert server address to proper format
- Establish connection with server
- Read and display server's reply

intro/daytimetcpcli.c

```
 1 #include    "unp.h"

 2 int
 3 main(int argc, char **argv)
 4 {
 5     int     sockfd, n;
 6     char    recvline[MAXLINE + 1];
 7     struct sockaddr_in servaddr;

 8     if (argc != 2)
 9         err_quit("usage: a.out <IPaddress>");

10     if ( (sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
11         err_sys("socket error");

12     bzero(&servaddr, sizeof(servaddr));
13     servaddr.sin_family = AF_INET;
14     servaddr.sin_port = htons(13);  /* daytime server */
15     if (inet_pton(AF_INET, argv[1], &servaddr.sin_addr) <= 0)
16         err_quit("inet_pton error for %s", argv[1]);

17     if (connect(sockfd, (SA *) &servaddr, sizeof(servaddr)) < 0)
18         err_sys("connect error");

19     while ( (n = read(sockfd, recvline, MAXLINE)) > 0) {
20         recvline[n] = 0;        /* null terminate */
21         if (fputs(recvline, stdout) == EOF)
22             err_sys("fputs error");
23     }
24     if (n < 0)
25         err_sys("read error");

26     exit(0);
27 }
```

intro/daytimetcpcli.c

Figure 1.5 TCP daytime client.

```
solaris% daytimetcpcli 206.168.112.96
Mon May 26 20:58:40 2003
```

Daytime client – IPv6 version

intro/daytimetcpcliv6.c

```
1 #include    "unp.h"

2 int
3 main(int argc, char **argv)
4 {
5     int     sockfd, n;
6     char    recvline[MAXLINE + 1];
7     struct sockaddr_in6 servaddr;

8     if (argc != 2)
9         err_quit("usage: a.out <IPaddress>");

10    if ( (sockfd = socket(AF_INET6, SOCK_STREAM, 0)) < 0)
11        err_sys("socket error");

12    bzero(&servaddr, sizeof(servaddr));
13    servaddr.sin6_family = AF_INET6;
14    servaddr.sin6_port = htons(13);     /* daytime server */
15    if (inet_pton(AF_INET6, argv[1], &servaddr.sin6_addr) <= 0)
16        err_quit("inet_pton error for %s", argv[1]);

17    if (connect(sockfd, (SA *) &servaddr, sizeof(servaddr)) < 0)
18        err_sys("connect error");

19    while ( (n = read(sockfd, recvline, MAXLINE)) > 0) {
20        recvline[n] = 0;    /* null terminate */
21        if (fputs(recvline, stdout) == EOF)
22            err_sys("fputs error");
23    }
24    if (n < 0)
25        err_sys("read error");

26    exit(0);
27 }
```

intro/daytimetcpcliv6.c

Figure 1.6 Version of Figure 1.5 for IP version 6.

Wrapper Functions

```
sockfd = Socket(AF_INET, SOCK_STREAM, 0);
```

lib/wrapsock.c

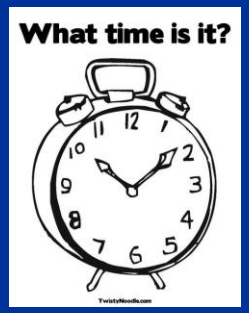
```
172 int
173 Socket(int family, int type, int protocol)
174 {
175     int     n;

176     if ( (n = socket(family, type, protocol)) < 0)
177         err_sys("socket error");
178     return (n);
179 }
```

lib/wrapsock.c

Figure 1.7 Our wrapper function for the `socket` function.

A Simple Daytime Server

- Create TCP socket
- Bind server's well-known port to socket
- Convert socket to listening socket
- Accept client connection, send reply



intro/daytimetcpsrv.c

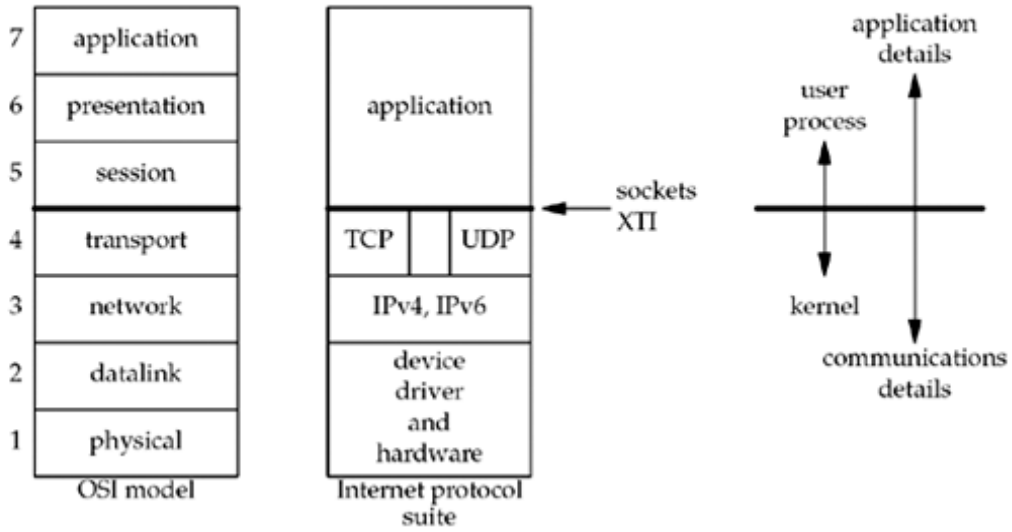
```
 1 #include    "unp.h"
 2 #include    <time.h>

 3 int
 4 main(int argc, char **argv)
 5 {
 6     int     listenfd, connfd;
 7     struct sockaddr_in servaddr;
 8     char    buff[MAXLINE];
 9     time_t  ticks;

10     listenfd = Socket(AF_INET, SOCK_STREAM, 0);

11     bzero(&servaddr, sizeof(servaddr));
12     servaddr.sin_family = AF_INET;
13     servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
14     servaddr.sin_port = htons(13);  /* daytime server */

15     Bind(listenfd, (SA *) &servaddr, sizeof(servaddr));

16     Listen(listenfd, LISTENQ);

17     for ( ; ; ) {
18         connfd = Accept(listenfd, (SA *) NULL, NULL);

19         ticks = time(NULL);
20         snprintf(buff, sizeof(buff), "%.24s\r\n", ctime(&ticks));
21         Write(connfd, buff, strlen(buff));

22         Close(connfd);
23     }
24 }
```

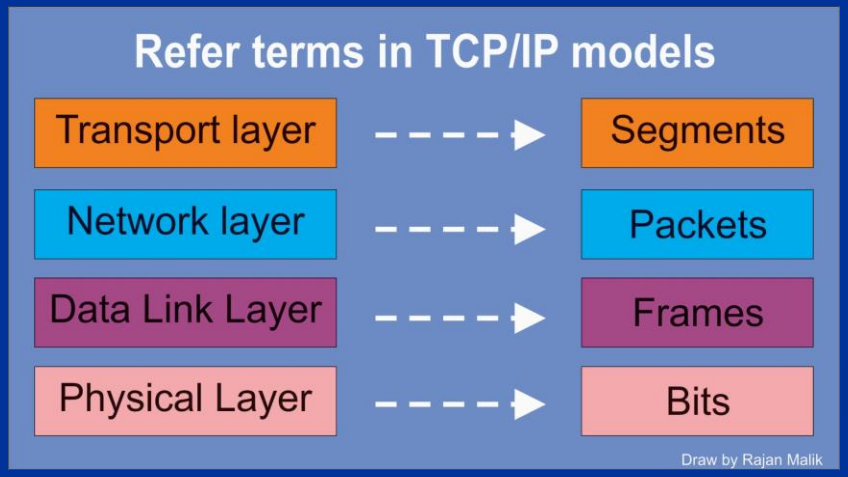
intro/daytimetcpsrv.c

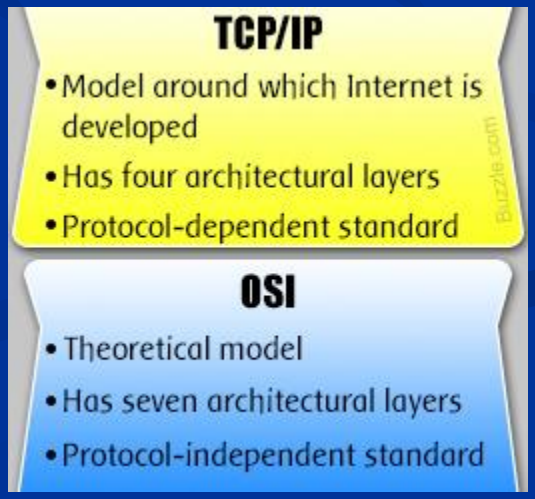
Figure 1.9 TCP daytime server.

OSI and TCP/IP

Figure 1.14. Layers in OSI model and Internet protocol suite.





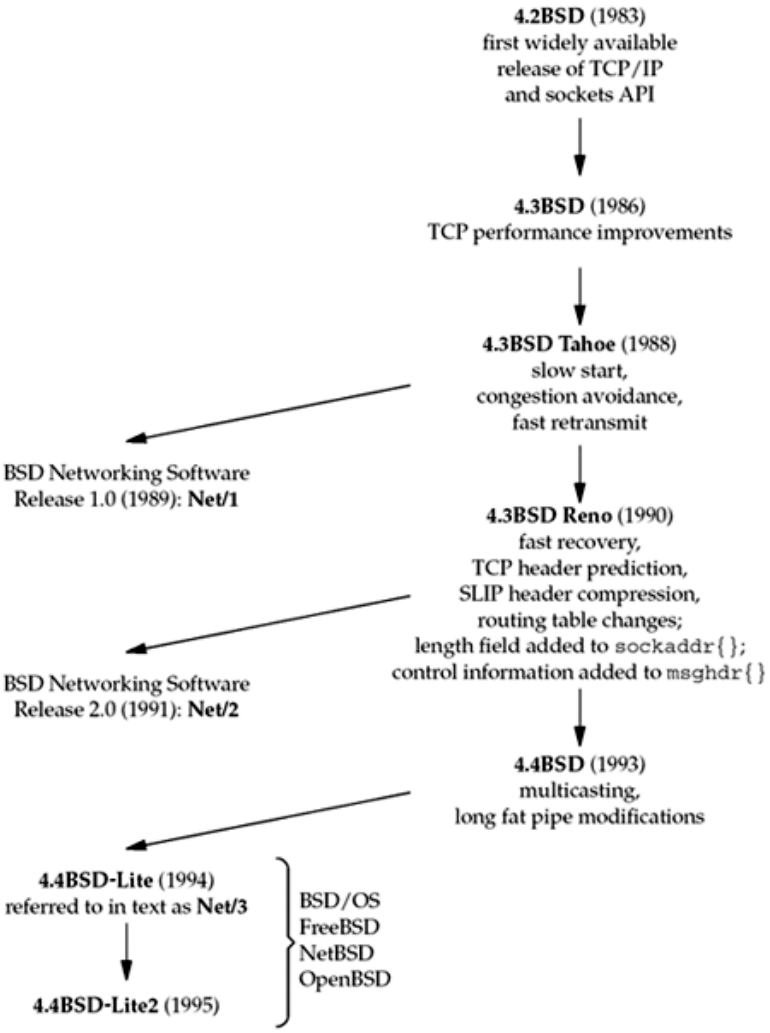


BSD Networking History

Berkeley Software Distribution (BSD)

- Unix OS
- Developed by Computer Systems Research Group (CSRG) at University of California, Berkeley
- 1970-1995

Figure 1.15. History of various BSD releases.



Unix Standards

- POSIX (Portable Operating System Interface)
  - family of standards, developed by IEEE, 1988–1996
  - POSIX.1, POSIX.2
- The Open Group
  - international consortium of vendors and end-users from industry, government, academia
  - formed by X/Open Company & Open Software Foundation (OSF), 1996
  - Single Unix Specification Version 3
    - most Unix systems today conform to this
- BSD Unix
  - first publicly available TCP/IP stack implementation
  - many commercial versions of Unix based on System V Release 4 (SVR4)
    - e.g. UnixWare, Solaris
- Linux
  - popular, free version of Unix
  - strictly not based on any standard, but has compatibility with POSIX and BSD Unix

32-bit vs 64-bit

- A network connection can be
  - from a 32-bit / 64-bit client
  - to a 32-bit / 64-bit host
- Socket API uses specifically defined data types to avoid problems with intrinsic C language data size issues
- ILP = integer, long integer, pointer

Figure 1.17. Comparison of number of bits to hold various datatypes for the ILP32 and LP64 models.

Datatype	ILP32 model	LP64 model
char	8	8
short	16	16
int	32	32
long	32	64
pointer	32	64

Exercises

- Daytime Client
  - intro/daytimetcpcli.c
- Daytime Server
  - intro/daytimetcpsrv.c