Read Tanenbaum, Bos: Chapter 8.1, 8.3, 10.5-10.5.2

Exercises

1. Suppose a network stack has a TCP layer, an IP layer, and an Ethernet layer.
   Which of the three layers provides each of these functionalities:
   (a) Packet routing across the entire internet; (b) Reliable data delivery; (c) Process-to-process communication; (d) Unreliable packets on the local network; (e) Synchronous byte streams.

2. What are "well known ports"?

3. (a) SLU's SMTP (send mail) server, `slumailrelay.slu.edu` will not accept connections from computers outside the slu domain. Explain why.

   (b) When using SLUguest wireless, the firewall blocks all connections to port 25, the SMTP port. Why?

4. When a TCP connection arrives, how does the OS know which process to notify?

5. A web browser generated the following request:

```
GET / HTTP/1.1[CRLF]
Host: slashdot.org[CRLF]
Connection: close[CRLF]
Accept-Encoding: gzip[CRLF]
Accept: */*[CRLF]
Accept-Language: en[CRLF]
User-Agent: Mozilla/5.0[CRLF]
[CRLF]
```

   (a) What is the URL of the page the browser is requesting?

   (b) What version of HTTP is being used?

   (c) What is the purpose of the final `[CRLF]`

6. IPv4 addresses of the form 10.x.x.x are reserved for private networks (for example, your home WiFi network might use these). How many addresses are in this space?

7. Suppose an HTTP server creates a new thread to handle every new connection. It's running slowly, and you discover that most of the CPU time is spent in calls to `pthread_create()`.

   What could you do to improve performance?

8. What is the purpose of the short program on the back of this page?

```
int main(int argc, char *argv[]) {
  if (argc != 4) {
    cerr << "usage: " << argv[0] << " hostname start end" << endl;
    exit(1);
  }

  int pstart = atoi(argv[2]);
  int pend = atoi(argv[3]);

  struct addrinfo *host;
  int err = getaddrinfo(argv[1],NULL,NULL,&host);
  if (err) {
    cerr << argv[1] << " : " << gai_strerror(err) << endl;
    exit(err);
  }

  int sock = socket(host->ai_family,SOCK_STREAM,0);
  if (sock < 0) { perror("socket"); exit(errno); }

  char p[6];
  for (int i = pstart; i <= pend; i++) {
    sprintf(p,"%d",i);
    getaddrinfo(argv[1],p,NULL,&host);
    if (connect(sock,host->ai_addr,host->ai_addrlen)) {
      ; // fail - do nothing
    } else {
      cout << p << endl;
      close(sock);
    }
  }
}
```