

第3章-运输层(1)

231880038 张国良

- R5. 在今天的因特网中，为什么语音和图像流量常常是经过 TCP 而不是经 UDP 发送。（提示：我们寻找的答案与 TCP 的拥塞控制机制没有关系。）
- R8. 假定在主机 C 端口 80 上运行的一个 Web 服务器。假定这个 Web 服务器使用持续连接，并且正在接收来自两台不同主机 A 和 B 的请求。被发送的所有请求都通过位于主机 C 的相同套接字吗？如果它们通过不同的套接字传递，这两个套接字都具有端口 80 吗？讨论和解释之。
- R14. 是非判断题：
- a. 主机 A 经过一条 TCP 连接向主机 B 发送一个大文件。假设主机 B 没有数据发往主机 A。因为主机 B 不能随数据捎带确认，所以主机 B 将不向主机 A 发送确认。
 - b. 在连接的整个过程中，TCP 的 `rwnd` 的长度决不会变化。
 - c. 假设主机 A 通过一条 TCP 连接向主机 B 发送一个大文件。主机 A 发送但未被确认的字节数不会超过接收缓存的大小。
 - d. 假设主机 A 通过一条 TCP 连接向主机 B 发送一个大文件。如果对于这条连接的一个报文段的序号为 m ，则对于后继报文段的序号将必然是 $m+1$ 。
 - e. TCP 报文段在它的首部中有一个 `rwnd` 字段。
 - f. 假定在一条 TCP 连接中最后的 `SampleRTT` 等于 1 秒，那么对于该连接的 `TimeoutInterval` 的当前值必定大于等于 1 秒。
 - g. 假设主机 A 通过一条 TCP 连接向主机 B 发送一个序号为 38 的 4 个字节的报文段。在这个相同的报文段中，确认号必定是 42。
- P1. 假设客户 A 向服务器 S 发起一个 Telnet 会话。与此同时，客户 B 也向服务器 S 发起一个 Telnet 会话。给出下面报文段的源端口号和目的端口号：
- a. 从 A 向 S 发送的报文段。
 - b. 从 B 向 S 发送的报文段。
 - c. 从 S 向 A 发送的报文段。
 - d. 从 S 向 B 发送的报文段。
 - e. 如果 A 和 B 是不同的主机，那么从 A 向 S 发送的报文段的源端口号是否可能与从 B 向 S 发送的报文段的源端口号相同？
 - f. 如果它们是同一台主机，情况会怎么样？
- P3. UDP 和 TCP 使用反码来计算它们的检验和。假设你有下面 3 个 8 比特字节：01010011, 01100110, 01110100。这些 8 比特字节和的反码是多少？（注意到尽管 UDP 和 TCP 使用 16 比特的字来计算检验和，但对于这个问题，你应该考虑 8 比特和。）写出所有工作过程。UDP 为什么要用该和的反码，即为什么不直接使用该和呢？使用该反码方案，接收方如何检测出差错？1 比特的差错将可能检测不出来吗？2 比特的差错呢？
- P5. 假定某 UDP 接收方对接收到的 UDP 报文段计算因特网检验和，并发现它与承载在检验和字段中的值相匹配。该接收方能够绝对确信没有出现过比特差错吗？试解释之。

Problem 1

解:

由于大多数防火墙都被配置为阻止UDP通信，因此使用TCP进行视频和语音通信可以让通信通过防火墙

Problem 2

解:

持续连接是TCP连接，连接套接字被标识为具有四个元组: (源IP地址、源端口号、目标IP地址、目标端口号)，源IP地址不同，所以套接字并不相同，但是目标端口号都是80，所以都具有端口80

Problem 3

解:

F F T F T F F

Problem 4

解:

设A, B随机选择端口5000, 5000

- a. 5000, 23
- b. 5000, 23
- c. 23, 5000
- d. 23, 5000
- e. 有可能相同
- f. 一定不相同

Problem 5

解:

字节和（进位加到地位上）反码: 11010001

为什么用反码：使用反码方案可以使发送方和接收方使用相同的计算方法来进行校验和的计算。发送方计算出校验和并将其放入报文段首部，接收方收到报文段后，将所有字段（包括校验和字段）相加，如果结果全为 1，则认为传输没有差错。如果直接使用和，接收方很难通过简单的相加操作来判断是否有差错

为了检测错误，接收方添加四个单词（三个原始单词和校验和）。如果和包含一个零，接收器知道有一个错误。所有的一位错误都会被检测到，但是两位错误可以不被检测到（例如，如果第一个单词的最后一个数字被转换为0，第二个单词的最后一个数字被转换为1）

Problem 6

解：

不，接收方不能完全确定没有发生任何位错误。这是因为计算数据包的校验和的方式。如果包中两个 16 位字的对应位（相加在一起）是 0 和 1，那么即使这些位分别翻转到 1 和 0，和仍然保持不变。因此，接收方计算的 1s 反码也将是相同的。这意味着，即使存在传输错误，校验和也将进行验证