

2019/2/18/19

物联网19-1 付发平

No.:

Date:

1. 传输层在 TCP/IP 模型中位于网络层和应用层中间。

作用: 传输层的主要作用是为运行在网络边缘的不同主机上的各应用程序之间提供通信服务, 传输层面向的不是主机而是应用, 或者说面向进程的通信服务, 是一种真正的端到端的通信服务。

服务: 传输层通信为不同进程之间提供了端到端通信服务。

传输层的另外一个主要作用是在应用层和网络层之间充当复用器的作用, 各种各样的网络应用, 通过传输层的端口被提交给统一的网络层, 并由 IP 协议用统一的方式将它们送到网络中。

2. 端口是为了使运行不同操作系统的计算机的应用进程能够互相通信, 使应用程序能对 TCP/IP 体系的应用进程进行标识。

传输层地址为: 当两个不同主机的两个进程需要通信时, 必须指明对方是哪一个进程, 这个标记称为传输层地址。

套接字是为了使应用程序能够方便地使用协议栈软件进行通信的一种方法。

传输层地址 = 主机 IP 地址 + 端口号

套接字: { 协议, 本地地址, 本地端口, 远程地址, 远程端口 }

3. UDP 是 OSI 参考模型和 TCP/IP 模型中都有的, 一种面向无连接的传输层协议。

特点: ① UDP 是无连接协议, 在发送数据之前不需要建立连接。

② UDP 使用尽最大努力交付, 不保证可靠交付, 同时也不使用拥塞控制。

③ UDP 是面向报文, 没有拥塞控制, 很适应多媒体通信的要求。

④ UDP 支持一对一、一对多、多对一和多对多的交互通信。

⑤ UDP 的首部开销小, 只有 8 个字节。

No.: Date:

4. 我认为 TCP 首部中序列号字段和确认号字段最重要，因为这两个字段是 TCP 可靠数据传输的关键部分。通信双方在进行连接时，需要首先发送序列号字段和确认号字段。

5. 连接建立:

- ① 客户端向服务器提出连接建立请求，即发出同步请求报文。
- ② 服务器收到客户端的连接请求后，向客户端发出同意建立连接的同步确认报文。

③ 客户端在收到服务器的同步确认报文后，向服务器发出确认报文。

连接释放:

- ① 客户端向服务器发出一个连接释放报文。
- ② 服务器收到客户端的释放连接请求后，向客户端发出确认报文。
- ③ 服务器在发送完最后的数据后，向客户端发出连接释放确认报文。
- ④ 客户端在收到服务器连接释放报文后，向服务器发出确认报文。

6. ① 重传计时器：在设定时间截止之前收到确认报文，则传输成功，撤销计时器，否则传输失败，重新发送数据报文。

② 坚持计时器：当接收方发出一个零窗口确认报文，发送方立即停止发送数据报文，直到接收方发出非零窗口的确认报文。

③ 保活计时器：用于判断两个 TCP 端点之间长久的连接是否正常。

④ 时间等待计时器：是 TCP 终止连接时启动的计时器。

7. 停止等待协议是发送方收到接收方的正确的确认报文后，才继续发送下一帧数据。

无论是数据报文还是应答报文，都有可能传输过程中丢失。

① 若数据报文丢失，接收端未收到数据报文，等待发送端发送。

② 若应答报文丢失，发送端未收到应答报文，等待接收端发送。

上述两种情况，都会使发送端等待接收端发送应答报文，而接收端等待

NOTEBOOK

发送端发送数据报文，故会导致死锁。

No. Date:

解决死锁的办法: 发送端在发出报文后即启动重发计时器, 若在设置的时间 $timeout$ 内未收到接收端的应答报文, 则认为报文已丢失, 从而重发报文。

8. 连续的ARQ协议: 发送方可连续发送多个报文, 不必每发完一个报文就停止并等待接收方的确认, 也被称为流水线传输。

由于发送方在时间周期内不必每发送一个报文就等待接收方的确认, 而是连续的发送多个报文, 故连续ARQ协议可以大幅度地提高信道利用率。

9. 滑动窗口是一种流控方法, 用于约定发送方可发送报文的数量。

过程: ① 发送方维持发送窗口;

② 收到一个确认后发送窗口向前滑动。

10. RTT是确定超时重传时间的最重要的基础。

RTT是对不断动态调整的RTT的测量值进行加权平均

RTT是超时重传时间, 对TCP的性能影响有重大作用。

11. 在TCP中, 当超时即发生连续多次重传时, 在两次重传之间的超时时间值的设置遵循“指数退避”原则:

当超时第一次重传后, 第二次重传等待时间是第一次的2倍, 第三次重传等待时间是第二次的2倍, 第三次重传等待时间是第一次的4倍。2为退避因子, 直到收到重传数据包的应答, RTT退避因子回复为1。

这样为了当网络处于无法快速交付数据报文状态时减小网络负担。

12. 流量控制就是让发送方的发送速率不要太快, 既要让接收方来得及接收, 也不要使网络发生拥塞, 利用滑动窗口机制可以很方便地在TCP连接上实现流量控制。

No. Date.

1. 慢开始:

① 在主机刚刚开始发送报文段时可先设置拥塞窗口 $cwnd=1$, 即设置为一个最大报文段 MSS 的数值。

② 在每收到一个对新报文段的确认后, 将拥塞窗口加1, 即增加一个 MSS 的数值。

③ 用这样的方法逐步增大发送端的拥塞窗口 $cwnd$, 可以使分组进入网络的速率更加合理。

拥塞避免:

只要网络没有出现拥塞, 拥塞窗口就再增大一些, 以便把更多的分组发送出去。但只要网络出现拥塞, 拥塞窗口就减少一些, 以减少注入网络中的分组数。