第六章

TCP与UDP

·传输层的作用

TCP/IP中有两个具有代表性的传输协议，它们分别是TCP和UDP。

TCP提供可靠的通信传输

UDP用于广播和细节控制交给应用的通信传输

传输层的定义

传输层的TCP和UDP，为了识别自己所传输的数据部分究竟应该发给哪个应用，也设定了这样一个编号

在TCP/IP的通信当中，需要指定端口号（识别码），根据端口号就可以识别在传输层上一层的应用层中所要进行处理的具体程序

通信处理

TCP/IP的众多应用协议大多以客户端/服务端的形式运行

服务端的程序要提前启动

确认一个请求究竟发给的是哪个服务端，可以通过所收到数据包的目标端口轻松识别

TCP，UDP通过接收数据中的目标端口号识别目标处理程序

两种传输层协议TCP和UDP

在TCP/IP协议中能够实现传输层功能的，具有代表性的协议是TCP和UDP

TCP

TCP是面向连接的，可靠度流协议

流就是指不间断的数据结构

当应用程序采用TCP发送消息时，虽然可以保证发送的顺序，但还是没有任何间隔的数据流发送给接收端

UDP

不具有可靠性的数据报协议

细微的处理它会交给上层的应用去完成

虽然可以确保发送消息的大小，却不一定能保证消息一定会到达

应用有时会根据自己的需要进行重发处理

TCP与UDP区分

TCP用于在传输层有必要实现可靠传输的情况

UDP主要用于那些对高速传输和实时性由较高要求的通信或广播通信

·端口号

端口号定义

在传输层中也有类似于地址的概念，就是端口号

端口号用来识别在计算机中进行通信的不同应用程序，所以也被称作程序地址

根据端口号识别应用

传输层协议正是利用这些端口号识别本机中正在进行通信的应用程序，并准确地将数据传输

通过IP地址，端口号，协议号进行通信识别

仅凭目标端口识别某一个通信是远远不够的

TCP/IP或UDP/IP通信中通常采用5个信息来识别一个通信：

1. 源IP地址
2. 目标IP地址
3. 协议号（用来区分上层是TCP或UDP的一种编号）
4. 源端口号
5. 目标端口号

只要其中某一项不同，则被认为是其他通信

端口号如何定义

确定端口号的方法有两种：

1. 标准既定的端口

也叫静态方法

指每个应用程序都有其指定的端口号，每个端口号都有其对应的使用目的

1. 时序分配法

也叫动态（时序）fenpeifa

服务端有必要确定监听端口号，但是接受服务的客户端没必要确定端口号

在这种方法下，客户端应用程序可以完全不用自己设置端口号，而全权交给操作系统分配

操作系统可以为每个应用程序分配互不冲突的端口号

端口号与协议

端口号由其使用的传输层协议决定，因此，不同的传输层协议可以使用相同的端口号

数据到达IP层后，会先检查IP首部中的协议号，再传给相应协议的模块

如果是TCP则传给TCP模块，如果是UDP则传给UDP模块去做端口号的处理，由于传输协议是各自独立地进行处理，因此相互之间不会受到影响

那些知名端口号与传输层协议并无关系，只要端口号一致都将分配统一种程序进行处理

·UDP

UDP的特点及其目的

UDP不提供复杂的控制机制，利用IP提供面向无连接的通信服务。并且它是将应用程序发来的数据在收到的那一刻，立即按照原样发送到网络上的一种机制

UDP不负责丢包时重发，包到达时顺序乱掉也没有纠正的功能

所以这些细节控制交由采用UDP的应用程序去解决

主要用于：

包总量较少的通信（DNS.SNMP等）

视频，音频等多媒体通信（即时通信）

限定于LAN等特定网络中的应用通信

广播通信（广播，多播）

·TCP

充分实现了数据传输时的各种控制功能，可以在进行丢包时的重发控制，还可以对次序乱掉的包进行顺序控制

TCP是面向有连接的协议，只有在通信对端存在时才会发送数据，从而控制通信流量的郎芬

连接是指各种设备，线路，或网络中进行通信的两个应用程序为了相互传递消息而专有的，虚拟的通信线路，也叫做模拟电路

TCP的特点及其目的

TCP保证可靠传输

通过序列号与确认应答提高可靠性

在TCP中，当发送端的数据到达连接主机时，接收端主机会返回一个已收到消息的通知，这个消息叫做确认应答（ACK），还有否定确认应答（NACK）

TCP通过肯定的确认应答（ACK）实现可靠的数据传输

未收到确认应答并不意味着数据一定丢失，也有可能是数据对方已经收到，而确认应答丢失，这种情况下，还是要进行重发

有时确认应答会延迟，因此源发送主机会重发，目标主机会收到重复数据包。为了对上层应用提供可靠的传输，必须舍弃重发的数据包

上述的确认应答处理，重发控制以及重复控制等功能都可以通过序列号实现。

序列号是按顺序发送数据的每一个字节（8位字节）都标上号码的编号

接收端查询接收数据TCP首部中的序列号和数据的长度，将自己的下一步应该接收的序号作为确认应答返回送出去

所以，通过序列号和确认应答号，TCP可以实现可靠传输

重发超时如何确定

重发超时是指在重发数据之前，等待确认应答到来的那个特定时间间隔

在每次发包时都会计算往返时间及其偏差，将这个往返时间和偏差相加，重发超时的时间就是比这个总和要稍大一点的值

数据不会被无限，重复地重发，达到一定重发次数后，如果仍没有任何确认应答，就会判断为网络或对端主机发生了异常，强制关闭连接，并通知应用通信异常强行终止

连接管理

可以用TCP首部用于控制的字段来管理TCP连接，一个连接的建立与断开，正常过程至少需要来回发送7个包才能完成

客户端（k）服务端（f）

K：SYN包（请求建立连接）

F：ACK（针对SYN的确认应答） SYN（请求建立连接）

K：ACK（针对SYN的确认应答）

连接建立完成 发送数据

K：FIN（请求切断连接）

F：ACK（针对FIN的确认应答）

F：FIN（请求切断连接）

K：ACK（针对FIN的确认应答）

TCP以段为单位发送数据

在建立TCP的同时，也可以确定发送数据包的单位，可以称其为“最大消息长度”

最理想的情况是：最大消息长度正好是IP中不会分片处理的最大数据长度

TCP在传送大量数据时，以MSS的大小将数据进行分割发送

MSS是在三次握手的时候，在两端主机之间被计算得出

两端的主机在发出建立连接的请求时，会在TCP首部中写入MSS选项，告诉对方自己的接口能够适应的MSS的大小。然后会在两者之间选择一个较小的值投入使用

利用窗口控制提高速度

窗口大小就是指无需等待确认应答而可以继续发送数据的最大值

这一举措实现使用了大量的缓冲区，通过对多个端同时进行确认应答的功能

可以提高通信性能

窗口控制与重发控制

窗口在一定程度上较大时，即使有少部分的确认应答丢失也不会进行数据重发。可以通过下一个确认应答进行确认

接收端在没有收到自己所期望的序号的数据时，会对之前收到的数据进行确认应答，发送端则一旦收到某个确认应答后，有连续3次收到同样的确认应答，则认为数据段已经丢失，需要进行重发，这种机制比起超时机制可以提供更为快速的重发服务

流控制

TCP提供一种机制可以让发送端根据接收端的实际接受能力控制发送的数据量

机制：

接收端主机向发送端主机通知自己可以接收数据的大小，于是发送端根据接收端的实际接收能力发送不超过这个限度的数据，该大小限度就被称作窗口大小

TCP首部中，专门有一个字段用来通知窗口大小，接收主机将自己可以接收的缓冲区大小放入这个字段中通知给发送端，这个字段的值越大，说明网络的吞吐量越高

发送端主机会根据接收端主机的指示，对发送数据的量进行控制

拥塞控制

在通信一开始就会通过一个叫做慢启动的算法得出的数值，对发送数据量进行控制

提高网络利用率的规范

Nagle算法

可以提高网络利用率，但是可能会发生某种程度的延迟

延迟确认应答

收到数据以后并不立即返回确认应答，而是延长一段时间的机制

TCP文件传输中，绝大多数是每两个数据段返回一次确认应答

捎带应答

TCP的确认应答和回执数据可以通过一个包发送，可以使收发的数据量减少

能够提高网络利用率从而降低计算机处理负荷的一种较优的处理机制

使用TCP的应用

如果应用自己处理一些细节上的控制，使用UDP协议是不错的选择

如果转发数据量较多，对可靠性的要求比较高时，可以选择使用TCP

·其他传输层协议

UDP-Lite

UDP-Lite（轻量级用户数据报协议 ）是扩展UDP机制的一种传输层协议

提供和UDP几乎一样的功能，不过计算校验和的范围可以由应用自行决定

就可以只针对不允许发生错误的部分进行校验和的检查，其他部分，即使发生了错误，也会被忽略不计。而这个包也不会被丢弃，而直接传给应用继续处理

SCTP

SCTP（流控制传输协议）与TCP一样，都是对一种提供数据到达与否相关可靠性检查的传输层协议，主要特点如下：

1. 以消息为单位收发

TCP接收端并不知道发送端应用所决定的消息大小，在SCTP中可以

1. 支持多重宿主

在有多个NIC的主机中，即使其中能够使用的NIC发生变化，也仍然可以继续通信

1. 支持多数据流通信

TCP中建立多个连接以后才能进行通信的效果，在SCTP中一个连接就可以

1. 可以定义消息的生存周期

超过生存周期的消息，不会被重发

主要用于进行通信的应用之间发送众多较小消息的情况，这些较小的应用消息被称作数据块，多个数据块组成一个数据包

多重宿主是指同一台主机具备多种网络的接口

同时使用以太网和无线LAN时，各自的NIC会获取到不同的IP地址

DCCP

DCCP（数据报拥塞控制协议）是一个辅助UDP的崭新的传输层协议

主要特点：

1. 与UDP一样，不能提供发送数据的可靠性传输
2. 面向连接，具备建立与断开连接的处理，在建立和断开连接上是具有可靠性
3. 能够根据网络拥堵情况进行拥塞控制
4. 为了进行拥塞控制，接收端收到包以后，接收端收到收到包以后返回确认应答ACK，确认应答将被用于重发与否的判断

·UDP首部的格式

由源端口号，目标端口号，包长度，校验和组成

源端口号：

表示发送端口号，字段长16位。该字段是可选项，有时可能不会设置源端口号，用于不需要返回的通信中

目标端口号：

表示接收端端口，字段长16位

包长度：

该字段保存了UDP首部的长度跟数据的长度之和，单位为字节（8位字节）

校验和

是为了提供可靠的UDP首部和数据而设计

·TCP首部格式

源端口号：

表示发送端端口号，字段长16位

目标端口号：

表示接收端端口号，字段长16位

序列号：

字段长32位，序列号（也叫序列）只发送数据的位置。每发送一次数据，就累加一次该数据字节数都大小

确认应答号：

字段长度为32位

指下一次应该收到的序列号，实际上，是指已收到确认应答号减1为止的数据

发送端收到这个确认应答以后可以认为在这个序号以前的数据已经被正常接收

数据偏移：

表示TCP所传输的数据部分应该从TCP包的哪个为开始计算，也可以把它看做TCP首部的长度

字段长4位，单位为4字节（即32位）

保留：

为了以后扩展时使用，其长度为4位，长度为4位，一般设置为0，但即使收到的包在该字段不为0，此包也不会被丢弃

控制位：

字段长为8位

每一位从左至右分别为：

CWR,ECE,URC,ACK,PSH,RST,SYN,FIN

这些控制标志也叫做控制位，当他们对应位上的值为1时，具体含义略

窗口大小：

字段长为16位

用于通知从相同TCP首部的确认应答号所指位置开始能够接收的数据大小（8位字节）

TCP不允许发送超过此处所示大小的数据

不过，如果窗口为0，则表示可以发送窗口探测，以了解最新的窗口大小，但这个数据必须是一个字节

校验和：

TCP的校验和与UDP想似，但是TCP的校验和无法关闭

紧急指针：

该字段长为16位

该字段的数值表示本报文段中的指针

选项

用于提高TCP的传输性能

在Web浏览器中一般会通过同时建立4个左右连接来提高吞吐量