XI'AN JIAOTONG UNIVERSITY

数据通信与计算机网络

第7章 运输层

主要内容与基本要求

- 主要内容
  - 运输层的基本概念、功能，TCP/IP体系中的运输层；
  - 用户数据报协议UDP；
  - 传输控制协议TCP。
- 基本要求
  - 理解运输层的基本概念、主要功能；
  - 理解TCP/IP中的运输层，掌握端口和插口（套接字）的概念；
  - 掌握用户数据报协议的基本概念、首部格式、伪首部的概念；
  - 理解TCP的基本概念，掌握首部格式、数据的编号与确认；
  - 掌握TCP的流量控制与拥塞控制、重发机制、运输连接管理。

本章目录

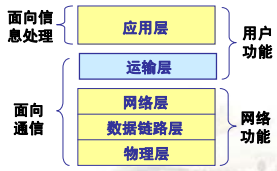
7.1 运输层协议概述

运输层的任务：向上一层进行通信的两个进程提供端到端的通信。

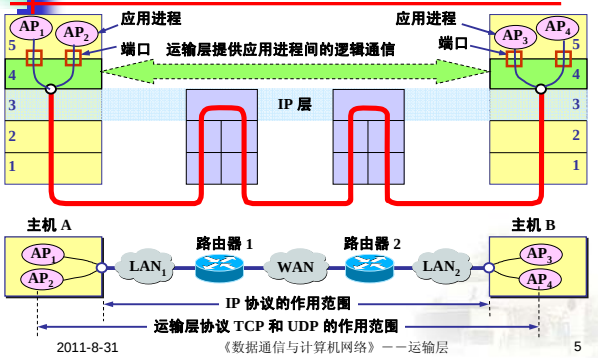
运输层只存在于通信子网以外的主机中。从通信和信息处理角度看，运输层属于面向通信的最高层；从网络功能或用户功能来划分，运输层属于用户功能的最低层。是通信子网与资源子网间的桥梁。

运输层的作用：

IP只能把分组送到目的主机，但不能交付给主机中的应用进程。运输层的一个重要功能是为应用层的多个同时通信的进程提供复用和分用服务。运输层为应用进程间提供端到端的逻辑通信，但网络层为主机之间提供逻辑通信。



7.1 运输层协议概述

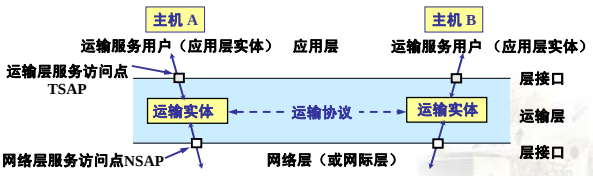


7.1 运输层协议概述

除复用和分用功能外，运输层还要对收到的报文进行差错检测。而且对不同的应用，运输层需要提供两种不同的服务，即面向连接服务和无连接服务。网络层无法同时实现这两种协议。

运输层还必须为应用进程间的通信提供端到端的流量控制。

从OSI观点看运输层与其上、下层之间的关系。



7.1 运输层协议概述

运输层中的两个运输对等实体之间的通信遵循运输协议。**运输协议的实现保证了运输层能够向应用层提供运输服务。运输层协议的实现使用了网络层向上层提供的服务。**

运输层向高层用户屏蔽了下面通信子网的细节，为应用层进程提供一条端到端的逻辑信道。**运输层采用面向连接的协议时**，向上层提供的**逻辑信道相当于一条全双工的可靠通道**；但当**运输层使用无连接的协议**时，向上层提供的**逻辑信道是一条不可靠信道**。

运输层协议与链路层协议很相似，但区别很大。运输层是两个主机通过多个网络进行通信，因此其环境比数据链路层复杂得多。

如果下面的网络或网际层不可靠（如IP），那么面向连接的运输协议就十分复杂，这个复杂的根本原因是要处理端系统之间的相对大的和可变的时延。**大的可变的时延使运输层的流量控制和差错控制复杂化。**

7.2 TCP/IP体系中的运输层

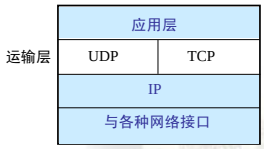
7.2.1 运输层中的两个协议

TCP/IP协议体系的运输层有两个协议：**用户数据报协议**UDP（User Datagram Protocol）和**传输控制协议**TCP（Transmission Control Protocol），分别给上层提供无连接服务和面向连接的服务。

OSI把运输层传输数据的单位叫做运输协议数据单元TPDU。但在TCP/IP体系中，把TCP的PDU称为**TCP报文段**（segment）；而把UDP的PDU称为**UDP报文或用户数据报**。

UDP数据报与IP数据报有很大区别。路由器只能看见IP数据报，而看不见UDP数据报。

TCP连接和网络层的虚电路完全不同。



7.2 TCP/IP体系中的运输层

TCP/IP的应用层协议采用**客户服务器模式**。**主动发起连接建立的进程叫客户**，而**被动等待连接建立的进程叫服务器**。

7.2.2 端口的概念

应用层的各种进程通过与运输层接口处的端口（port）与运输实体进行通信。按OSI的术语，端口就是运输层服务访问点TSAP。

端口的作用是让应用层的各种应用进程都能将其数据向下通过端口交付给运输层，并让运输层知道应当将其报文段中的数据向上通过端口交给应用层相应的进程。从此意义上讲，端口是用来标志应用层进程的。

端口用16比特表示，**只具有本地意义**，只是为了标志本计算机应用层中的各进程。

端口分为**熟知端口**和**一般端口**。熟知端口由应用层中各种不同的服务器进程使用。而一般端口分配给请求通信的客户进程。

7.2 TCP/IP体系中的运输层

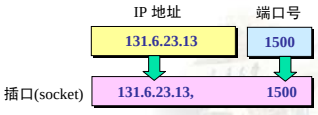
应用程序	FTP	TELNET	SMTP	DNS	TFTP	HTTP	SNMP
熟知端口	21	23	25	53	69	80	161

端口只具有本地意义，为了通信时不致发生混乱，需要把端口号和IP地址结合起来使用。

TCP使用“连接”作为最基本的抽象，一个TCP连接由它的两个端点来标志，**每个端点由IP地址和端口号决定**，又称为**插口**（socket），**或套接字、套接口**。

TCP连接由一对插口惟一确定。

UDP虽然没有连接的概念，但收发双方都有插口。



7.3 用户数据报协议UDP

7.3.1 UDP概述

用户数据报协议UDP只在IP数据报服务的基础上增加了**端口功能**（复用和分用）和**差错检测功能**。只提供不可靠的服务。其优点为：

（1）发送数据前不需要建立连接，减少了开销和平均时延。

（2）不使用拥塞控制，不保证可靠交付。

（3）只有8个字节的首部开销。

（4）网络拥塞不会使源主机的发送速率降低，这对某些实时应用很重要。

使用UDP的应用层协议有：域名服务DNS、简单文件传输协议TFTP、路由信息协议RIP、简单网络管理协议SNMP等。

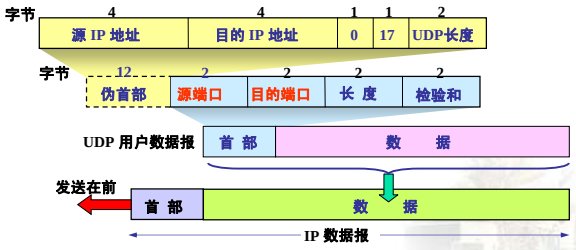
UDP不使用拥塞控制可能引起网络严重拥塞。

可通过应用进程提高可靠性。

7.3 用户数据报协议UDP

7.3.2 UDP用户数据报的首部格式

UDP数据报格式很简单，包括数据字段和首部字段两部分。



7.3 用户数据报协议UDP

UDP检验和的计算方法有些特殊。在计算检验和时要在数据报前加12字节的伪首部。

伪首部只是在计算检验和时使用，既不向下传送也不向上提交。

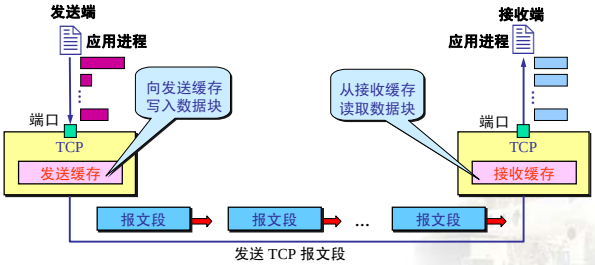
UDP计算检验和的方法和IP数据报计算首部检验和的方法相似。但IP数据报的检验和只检验IP数据报的首部，而UDP的检验和将首部和数据部分一起检验。

UDP检验和的计算

7.4 传输控制协议TCP

7.4.1 TCP概述

TCP是面向连接的运输层协议，提供全双工的和可靠交付的服务。



7.4 传输控制协议TCP

下面先介绍TCP的首部格式，然后讨论一些TCP保证数据传送可靠、按序、无丢失和不重复的机制。

7.4.2 TCP报文段的首部

TCP的全部功能都体现在其首部中各字段的作用上。要掌握TCP的工作原理，必须清楚TCP首部各字段的作用。

和IP类似，TCP的前20字节是固定首部，后面是根据需要增加的选项，长度是4字节的整数倍。最小长度20字节，最大长度60字节。

TCP首部格式

7.4.3 TCP的数据编号和确认

为了保证传输的可靠，TCP采用了对数据编号和确认的机制。

7.4 传输控制协议TCP

TCP协议是**面向字节**的。即TCP把要发送的**整个报文看成是一个个字节组成的数据流，每个字节对应一个序号**。连接建立时商定初始序号。

TCP的**确认**是对接收到的数据的最高序号表示确认。确认序号表示接收端期望收到的数据的第一个字节的序号。

由于TCP连接能提供全双工通信，所以可以用“**捎带**”的方法确认。

那么，在发送端，TCP如何决定发送一个报文段呢？TCP用三种机制控制报文段的发送：① 当缓存中数据达到最大报文段长度时发送；② 应用进程指明要求发送；③ 发送端计时器时间到了。

Nagle算法：若发送端进程将要发送的数据逐字节送到发送端的TCP缓存，则发送第一个字符，缓存后面的；在收到确认后把缓存中的字符发出去，继续缓存；还规定，当到达的字符已达到窗口大小的一半或已达到报文段的最大长度时，立即发送一个报文段。

7.4 传输控制协议TCP

糊涂窗口综合症（silly window syndrome）的解决办法，发送端不发送很小的报文段，接收端也不要在缓存刚有一点小的空位置时，就将一个很小的窗口通知给发送端。

与数据链路层的处理类似，发送方发送一个报文段后要设置计时器。若超时则重发；接收端收到出错的报文，则丢弃（不发送否认信息）；若收到重复报文段，则丢弃并发送（或捎带发回）确认信息。如果收到不按序的帧，TCP没有规定如何处理，实现者自定。

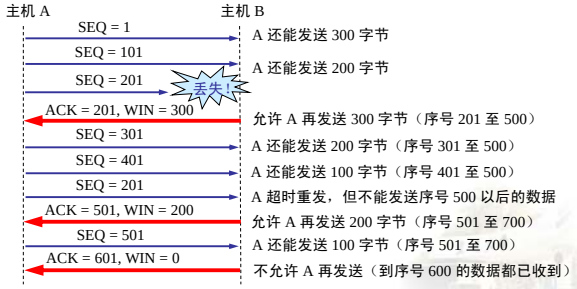
7.4.4 TCP的流量控制与拥塞控制

1. 滑动窗口的概念

TCP采用大小可变的滑动窗口进行流量控制，窗口大小的单位是字节。发送窗口在连接建立时由双方商定。通信过程中，接收端可根据自己的资源情况动态地调整发送窗口的上限值。

7.4 传输控制协议TCP

TCP中的窗口概念



7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

TCP利用发送窗口调节向网络注入分组的速率不仅是为了使接收端来得及接收，同时也为了对网络进行拥塞控制。拥塞控制的目标是将网络中的分组数量维持在一定的水平之下。

2. 慢开始和拥塞避免

发送端在发送时，既要考虑接收端的接收能力，又要从全局考虑不要使网络发生拥塞。因此发送窗口的大小受两种因素控制。

（1）**接收端窗口rwnd**（receiver window），又称**通知窗口**，是接收端根据其接收缓存大小所许诺的窗口值，是**来自接收端的流量控制**。

（2）**拥塞窗口cwnd**（congestion window），是发送端根据自己**估计**的拥塞程度而设置的窗口值，是**来自发送端的流量控制**。

发送窗口的上限值＝min[rwnd，cwnd] （7-1）

7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

慢开始算法的原理：用试探的方法，由小到大逐渐增大发送端的拥塞窗口数值。

开始时，拥塞窗口cwnd为一个MSS。每收到一个对新报文段的确认后，将拥塞窗口增加至多一个MSS的数值。

为说明原理，假定接收窗口rwnd足够大，即发送窗口只受cwnd制约。

慢开始并不是说cwnd的增长速率慢，而是指一开始cwnd较小。

为了防止cwnd的增长引起拥塞，增加一个状态变量，慢开始门限ssthresh，当cwnd超过ssthresh时，使用拥塞避免算法。

当执行拥塞避免算法时，拥塞窗口cwnd每经过一个往返时延RTT就增加一个MSS的大小。此时，拥塞窗口按线性规律缓慢增长。

无论在那个阶段，只要**发现拥塞**，则把慢开始门限ssthresh设置为发生拥塞时发送窗口值的一半，cwnd重新设置为1，并执行慢开始算法。

7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

慢开始和拥塞避免算法demo

“乘法减小”和“加法增大”。

拥塞避免是指在拥塞避免阶段让拥塞窗口按线性规律增长，使网络比较不容易出现拥塞，而并非指完全能够避免拥塞的发生。

3. 快重传和快恢复

使用慢开始和拥塞避免算法时，有时一条TCP连接会因等待重传计时器的超时而空闲较长的时间。

为了解决这个问题，快重传算法规定：发送端只要一连收到三个重复的ACK，就可断定有分组丢失了，应立即重传而不必等待超时。

快重传算法举例

与快重传配合使用的还有快恢复算法。

7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

慢开始和拥塞避免算法一旦发生拥塞就把拥塞窗口减为1并执行慢开始算法，缺点是网络不能很快地恢复到正常工作状态。快恢复算法规定：

（1）当发送端连续收到三个或三个以上（用n表示）重复的ACK，按“乘法减小”重新设置慢开始门限ssthresh。

（2）把拥塞窗口设为ssthresh+n×MSS。

（3）若发送窗口值还允许发送报文段，按拥塞避免算法继续发送报文段。

（4）若收到了确认新的报文段的ACK，将cwnd缩小到ssthresh。

采用快恢复算法时，慢开始只是在TCP连接建立时才使用。

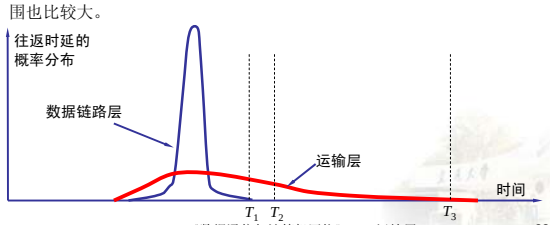
7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

7.4.5 TCP的重传机制

重传机制是TCP最重要和最复杂的问题之一。它要解决的是超时时间如何设置的问题。

TCP的重传之所以复杂是因为它所要处理的时延相对大而且变化范围也比较大。



7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

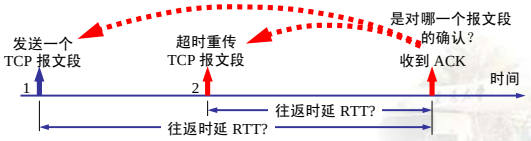
TCP采用了一种自适应算法。其思想是记录每个报文段的往返时延，在此基础上通过如下的加权平均求出报文段的平均往返时延RTT。

平均往返时延RTT＝α×(旧的RTT)＋(1－ α)(新的往返时延样本)

其中，0≤α＜1，典型取值为7/8。显然，超时重传时间RTO应略大于平均往返时延RTT，即

RTO＝β×RTT

其中，β是大于1的系数，实际上也很难确定。



7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

Karn算法：在计算平均往返时延时，只要报文段重传了，就不再采用其往返时延样本。

问题是，如果报文段的时延突然增大了很多，那么显然会引起报文段的重传。但Karn算法不考虑重传报文段的往返时延样本，使重传时间无法更新。

修正的Karn算法：报文段每重传一次，就将重传时间增大一些，如

新的重传时间＝γ×(旧的重传时间)

γ通常取2。当不再发生重传时，再根据报文段的往返时延更新平均往返时延RTT和重传时间。

7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

7.4.6 采用随机早期丢弃RED进行拥塞控制

拥塞控制和路由器对分组的处理策略有关。路由器的尾部丢弃会导致许多TCP连接在同一时间突然都进入到慢开始状态。为避免这种现象，可以让路由器采用随机早期丢弃的策略，其基本思想如下：

路由器队列维持两个参数，最小队列长度门限THmin和最大门限THmax。对每个到达的数据报计算平均队列长度Lav。若Lav小于THmin，则将新到达的数据报放入队列排队；若Lav超过THmax，将新到达的数据报丢弃；若介于二者之间，则按照某一概率p将新到达的数据报丢弃。

RED正常工作的关键是三个参数的选择要合适。最小门限必须足够大，最大门限和最小门限之差也必须足够大。

平均队列长度的计算与RTT的计算方法类似

Lav＝(1 －δ)×(旧的Lav)＋ δ×(当前队列长度样本)

7.4 传输控制协议TCP

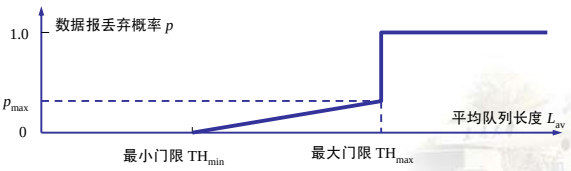
XI'AN JIAOTONG UNIVERSITY

丢弃概率p与两个门限的关系为

$$p_{temp} = p_{max} \times (L_{av} - TH_{min}) / (TH_{max} - TH_{min})$$

为了使数据报丢弃的间隔相对地更加均匀些，可以用如下的改进公式计算丢弃概率。

$$p = p_{temp} / (1 - count \times p_{temp})$$



7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

7.4.7 TCP的运输连接管理

TCP是面向连接的，其运输连接包括三个阶段：连接建立、数据传送和连接释放。

TCP建立连接时采用客户服务器方式。

TCP在连接建立过程中要解决以下三个问题：

（1）要使每一方能够知道对方的存在。

（2）要允许双发协商一些参数（如MSS，最大窗口大小等）。

（3）能对运输实体资源进行分配。

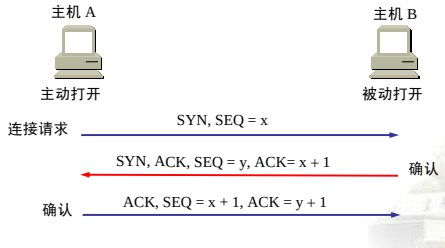
服务器进程发“**被动打开**”命令，等待客户进程的连接请求；客户进程在需要建立连接时发出“**主动打开**”命令。

7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

用三次握手建立TCP连接的过程如下：

为什么要用三次握手？



7.4 传输控制协议TCP

XI'AN JIAOTONG UNIVERSITY

数据传输结束后，通信的双发都可以发出释放连接的请求。