



石家庄铁道大学
SHIJIAZHUANG TIEDAO UNIVERSITY

计算机网络

第 15 讲 运输层 (I)



上讲内容回顾

- ◆ 虚拟专用网络VPN
- ◆ 网络地址转换NAT
- ◆ 路由器的基本命令



本讲内容

◆ 运输层协议概述

运输层的两个主要协议，进程之间的通信，
运输层的端口

◆ 用户数据报协议UDP

UDP概述和首部格式

◆ 传输控制协议TCP

TCP的特点和TCP的连接

运输层协议概述

进程之间的通信

- 从通信和信息处理的角度看，**运输层向它上面的应用层提供通信服务**，它属于面向通信部分的最高层，同时也是用户功能中的最低层。

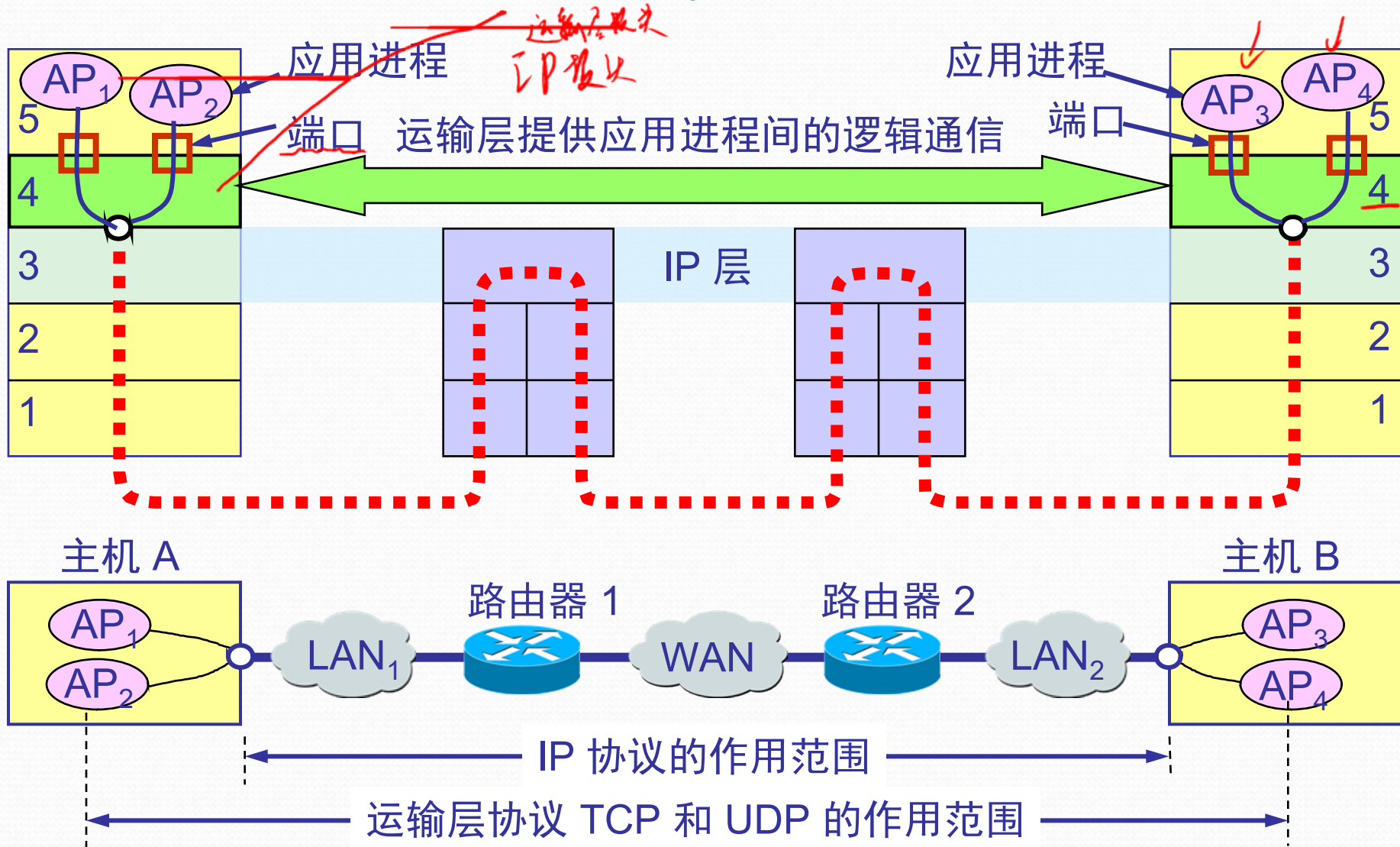




- 当网络的边缘部分中的两个主机使用网络的核心部分的功能进行端到端的通信时，只有位于网络边缘部分的主机的协议栈才有运输层，而网络核心部分中的路由器在转发分组时都只用到下三层的功能。

运输层为相互通信的应用进程提供了逻辑通信

目的编号





石家莊鐵道大學
SHIJIAZHUANG TIEDAO UNIVERSITY



应用进程之间的通信

- 两个主机进行通信实际上就是两个主机中的**应用进程互相通信**。
- 应用进程之间的通信又称为**端到端的通信**。
- 运输层的一个很重要的功能就是**复用**和**分用**。应用层不同进程的报文通过不同的端口向下交到运输层，再往下就共用网络层提供的服务。
- “**运输层提供应用进程间的逻辑通信**”。“逻辑通信”的意思是：运输层之间的通信**好像**是沿水平方向传送数据。但事实上这两个运输层之间并没有一条水平方向的物理连接。

运输层协议和网络层协议的主要区别





运输层的主要功能

边界功能

- 运输层为应用进程之间提供端到端的逻辑通信（但网络层是为主机之间提供逻辑通信）。
 - 运输层对收到的报文进行差错检测和流量控制，以改善服务质量。
可靠，流控，拥塞
 - 运输层需要有两种不同的运输协议
- 面向连接的 TCP：向应用层提供可靠服务。✓
- 和无连接的 UDP：向应用层提供不可靠的服务。尽力而为



两种不同的运输协议

- 运输层向高层用户屏蔽了下面网络核心的细节（如网络拓扑、所采用的路由选择协议等），它使应用进程看见的就是好像在两个运输层实体之间有一条端到端的逻辑通信信道。
- ✓ ● 当运输层采用面向连接的 TCP 协议时，尽管下面的网络是不可靠的（只提供尽最大努力服务），但这种逻辑通信信道就相当于一条全双工的可靠信道。—— 时延长，更复杂，更可靠
- ✓ ● 当运输层采用无连接的 UDP 协议时，这种逻辑通信信道是一条不可靠信道。



运输层的两个主要协议

TCP/IP 的运输层有两个不同的协议：

(1) 用户数据报协议 UDP

(User Datagram Protocol)

(2) 传输控制协议 TCP

(Transmission Control Protocol)



TCP 与 UDP

- 两个对等运输实体在通信时传送的数据单位叫作运输协议数据单元 TPDU (Transport Protocol Data Unit)。
- TCP 传送的数据单元是 **TCP 报文段**(segment)
- UDP 传送的数据单元是 **UDP 报文**或**用户数据报**。

frame

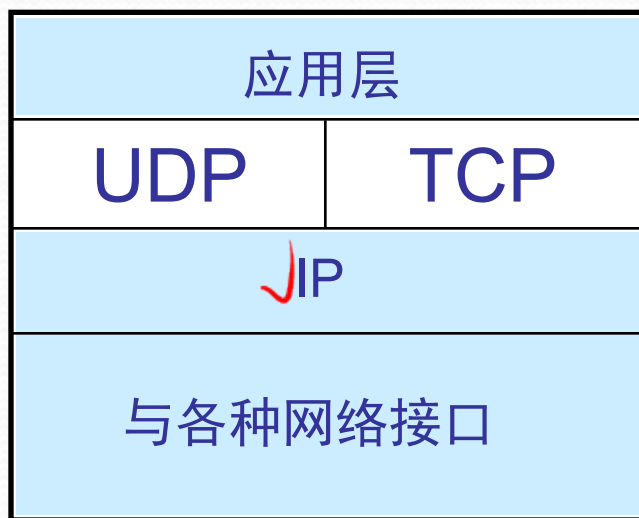
packet

segment



TCP/IP 体系中的运输层协议

运输层





TCP 与 UDP

- UDP 在传送数据之前不需要先建立连接。对方的运输层在收到 UDP 报文后，不需要给出任何确认。虽然 UDP 不提供可靠交付，但在某些情况下 UDP 是一种最有效的工作方式。
- TCP 则提供面向连接的服务。TCP 不提供广播或多播服务。由于 TCP 要提供可靠的、面向连接的运输服务，因此不可避免地增加了许多的开销。这不仅使协议数据单元的首部增大很多，还要占用许多的处理机资源。



还要强调两点

- 运输层的UDP用户数据报与网际层的IP数据报有很大区别。IP数据报要经过互连网中许多路由器的存储转发，但UDP用户数据报是在运输层的端到端抽象的逻辑信道中传送的。
- TCP 报文段是在运输层抽象的端到端逻辑信道中传送，这种信道是可靠的全双工信道。但这样的信道却不知道究竟经过了哪些路由器，而这些路由器也根本不知道上面的运输层是否建立了 TCP 连接。

编



运输层的端口

- 运行在计算机中的进程是用进程标识符来标志的。
- 运行在应用层的各种应用进程却不应当让计算机操作系统指派它的进程标识符。这是因为在因特网上使用的计算机的操作系统种类很多，而不同的操作系统又使用不同格式的进程标识符。
- 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，就必须用统一的方法对 TCP/IP 体系的应用进程进行标志。



需要解决的问题

- 由于进程的创建和撤销都是动态的，发送方几乎无法识别其他机器上的进程。
- 我们往往需要利用目的主机提供的功能来识别终点，而不需要知道实现这个功能的进程。



端口号(protocol port number) 简称为端口(port)

- 解决这个问题的方法就是在运输层使用协议端口号(protocol port number)，或通常简称为端口(port)。
- UDP和TCP使用应用层接口处的端口 (port)与上层的应用进程进行通信。
- 端口就是运输层服务访问点 TSAP。
- 端口的作用就是让应用层的各种应用进程都能将其数据通过端口向下交付给运输层，以及让运输层知道应当将其报文段中的数据向上通过端口交付给应用层相应的进程。
- 端口是用来标志应用层的进程。

应用层
运输层
网络层

发送方

应用进程



端口

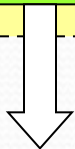
TCP 复用

UDP 复用

TCP 报文段

UDP
用户数据报

IP 复用



IP 数据报



接收方

应用进程



端口

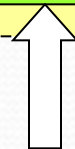
TCP 分用

UDP 分用

TCP 报文段

UDP
用户数据报

IP 分用



IP 数据报





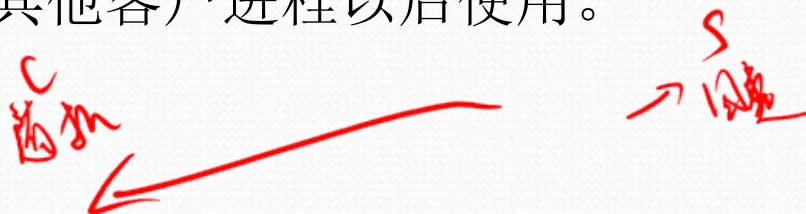
TCP 的端口

- 端口用一个 16 位端口号进行标志。
- 端口号只具有本地意义，即端口号只是为了标志本计算机应用层中的各进程。在因特网中不同计算机的相同端口号是没有联系的。



三类端口

- 熟知端口，数值一般为 0~1023。
- 登记端口号，数值为1024~49151，为没有熟知端口号的应用程序使用的。使用这个范围的端口号必须在 IANA 登记，以防止重复。
- 客户端口号或短暂端口号，数值为49152~65535，留给客户进程选择暂时使用。当服务器进程收到客户进程的报文时，就知道了客户进程所使用的动态端口号。通信结束后，这个端口号可供其他客户进程以后使用。





用户数据报协议 UDP

UDP 概述

- UDP 只在 IP 的数据报服务之上增加了很少一点的功能，即端口的功能和差错检测的功能。
- 虽然 UDP 用户数据报只能提供不可靠的交付，但 UDP 在某些方面有其特殊的优点。



UDP 的主要特点

- UDP 是无连接的，即发送数据之前不需要建立连接。
- UDP 使用尽最大努力交付，不保证可靠交付。
- UDP 是面向报文的。
- UDP 没有拥塞控制，适合多媒体通信的要求。
- UDP 支持一对一、一对多、多对一和多对多的交互通信。
- UDP 的首部开销小，只有 8 个字节。

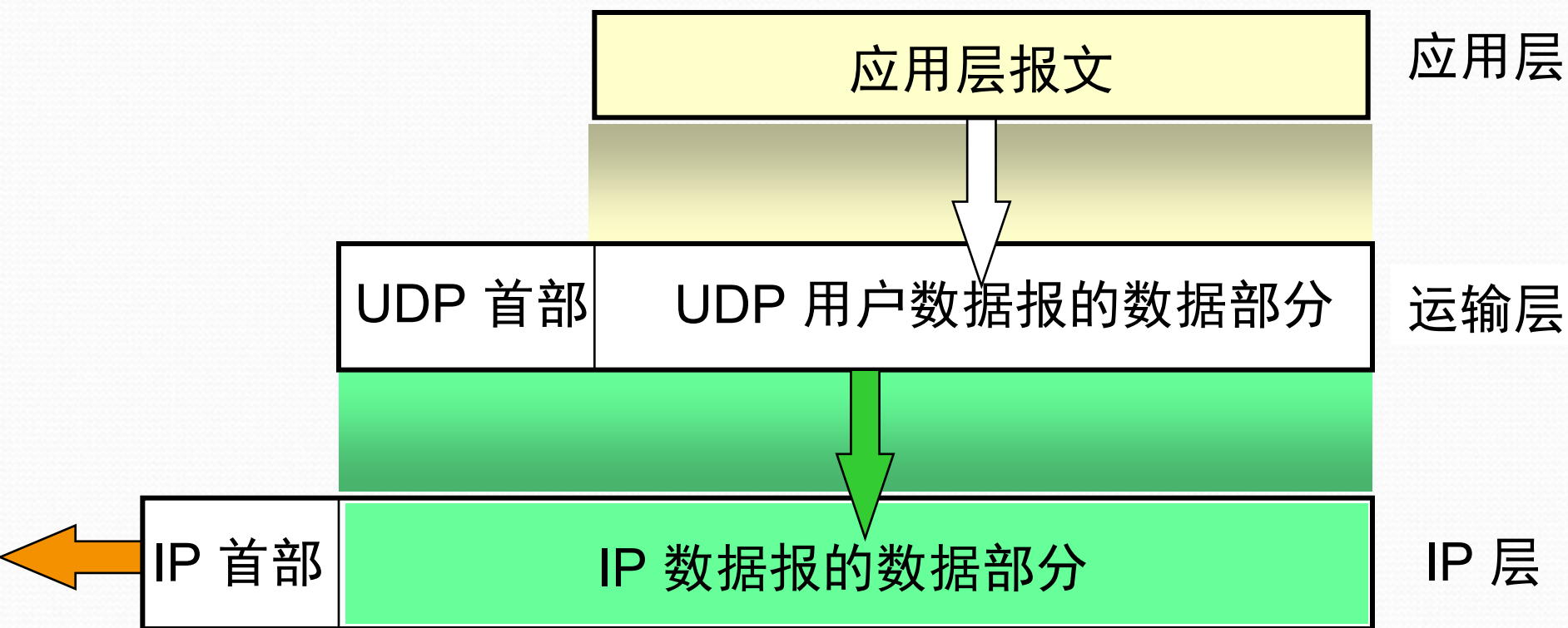
2018.3.9



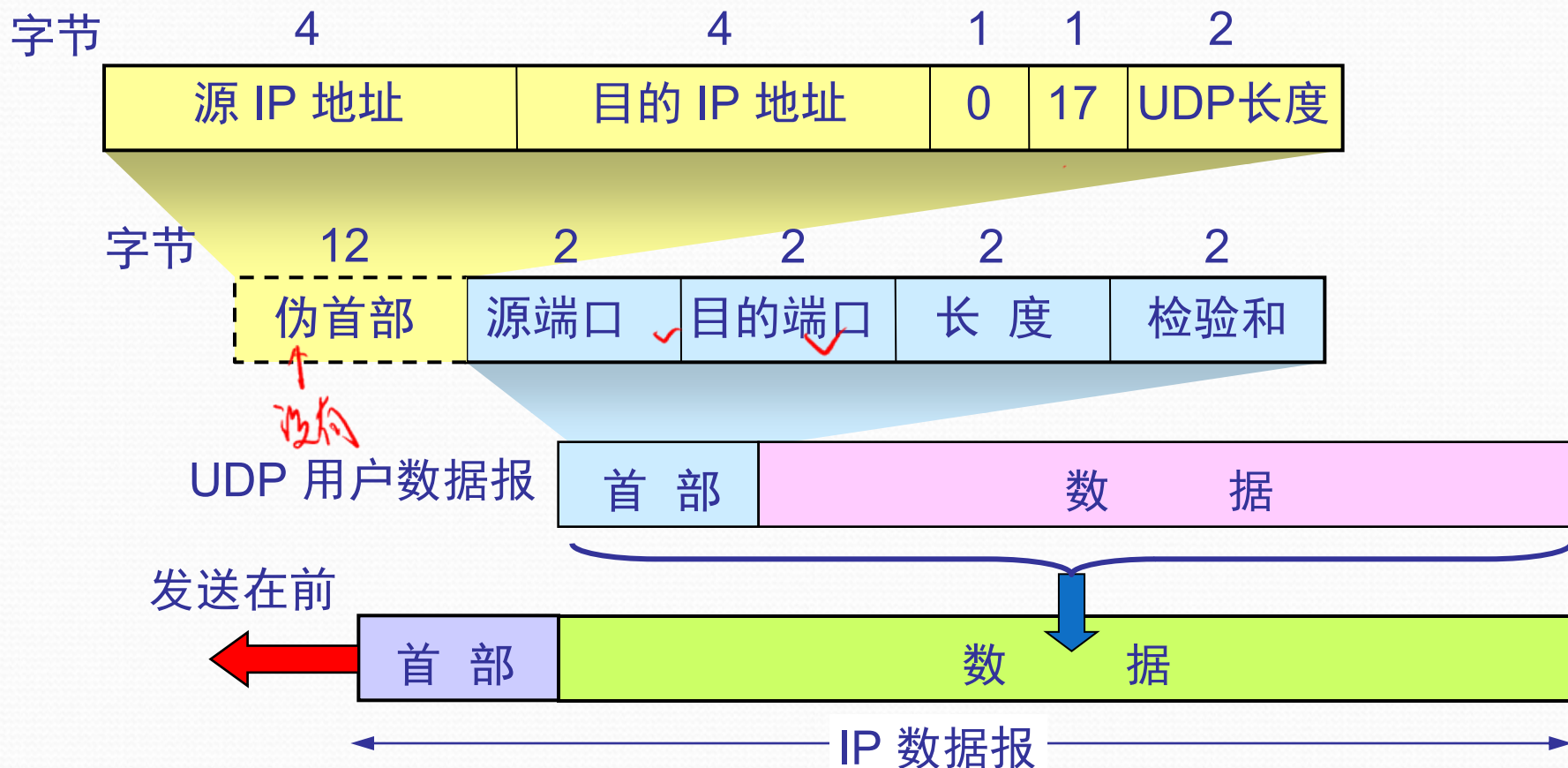
面向报文的 UDP

- 发送方 UDP 对应用程序交下来的报文，在添加首部后就向下交付 IP 层。UDP 对应用层交下来的报文，既不合并，也不拆分，而是保留这些报文的边界。
- 应用层交给 UDP 多长的报文，UDP 就照样发送，即一次发送一个报文。
- 接收方 UDP 对 IP 层交上来的 UDP 用户数据报，在去除首部后就原封不动地交付上层的应用进程，一次交付一个完整的报文。
- 应用程序必须选择合适大小的报文。

UDP 是面向报文的

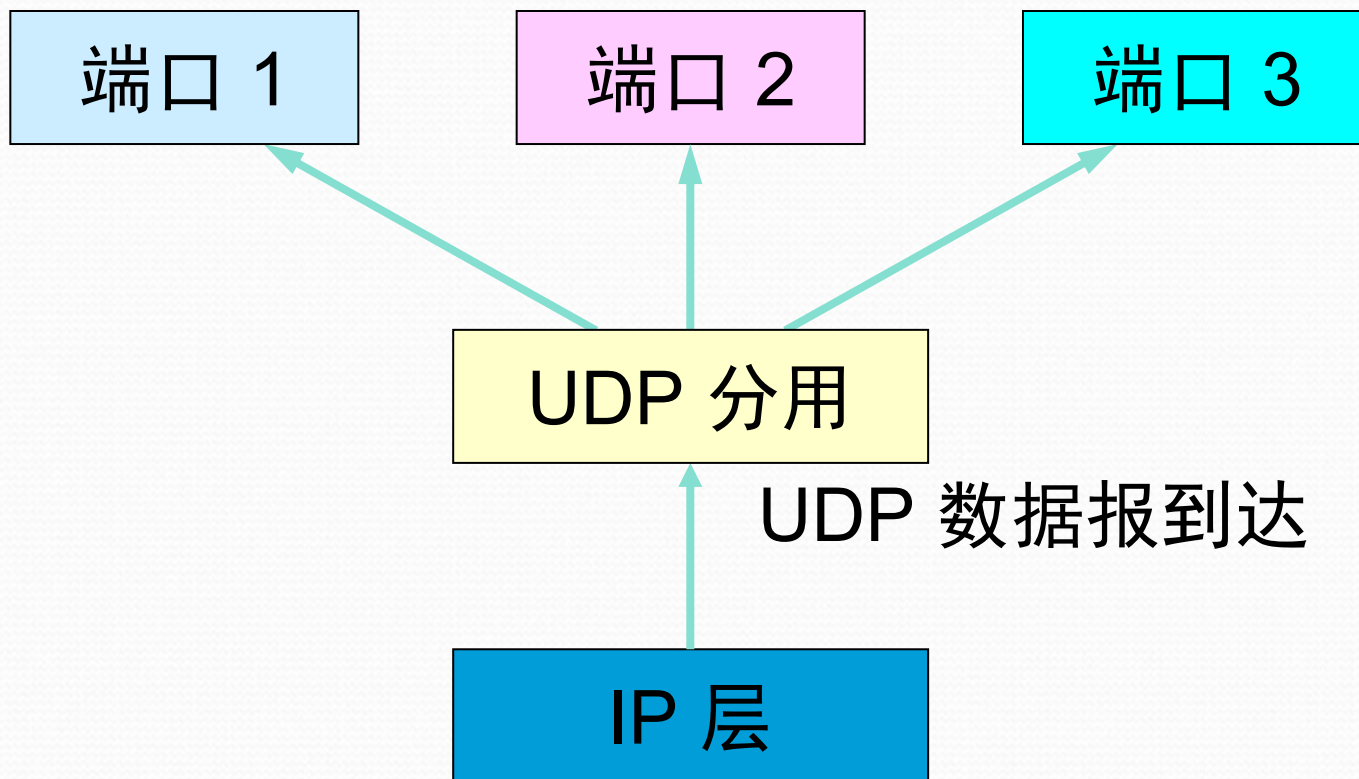


UDP 的首部格式

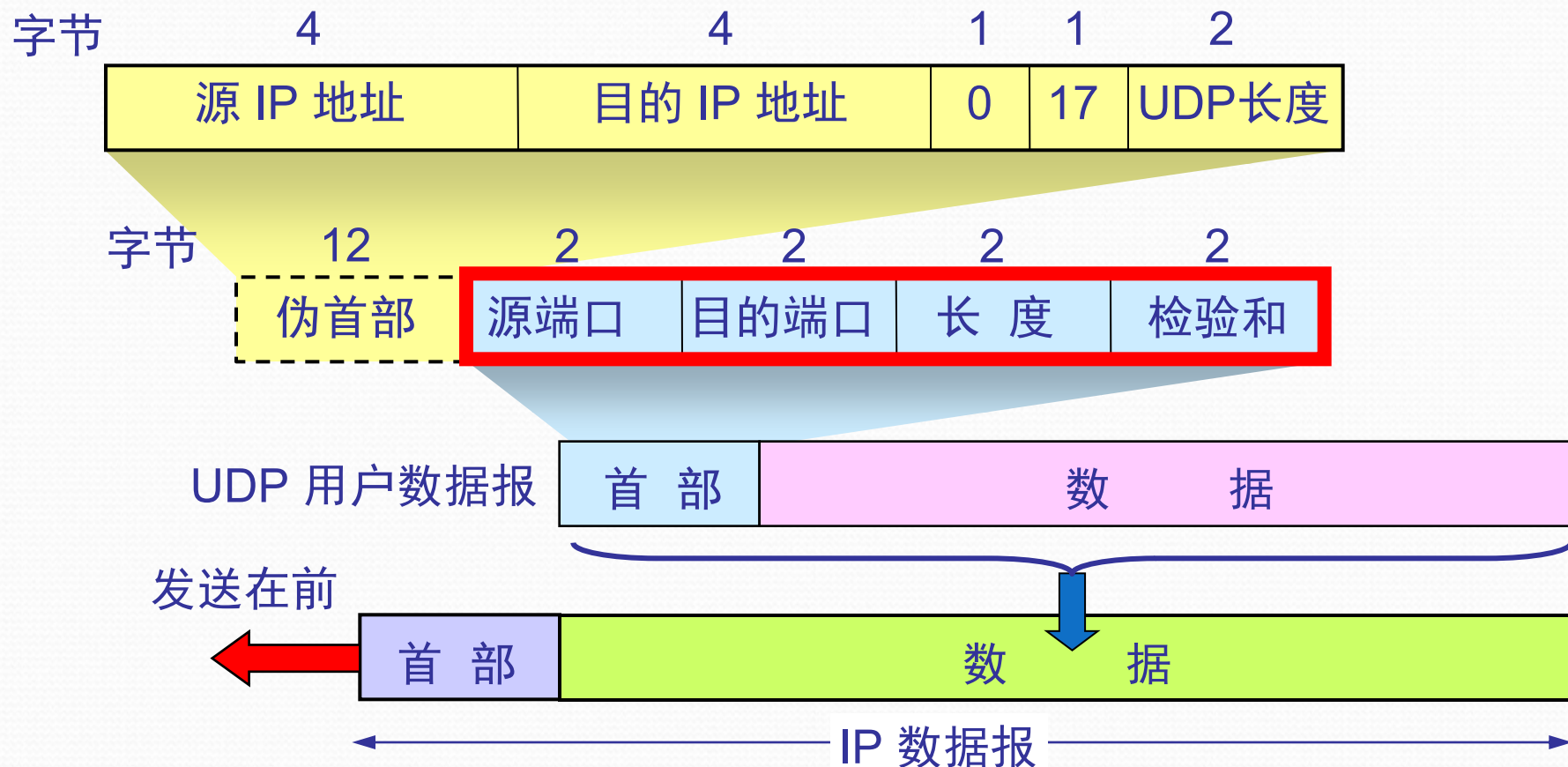




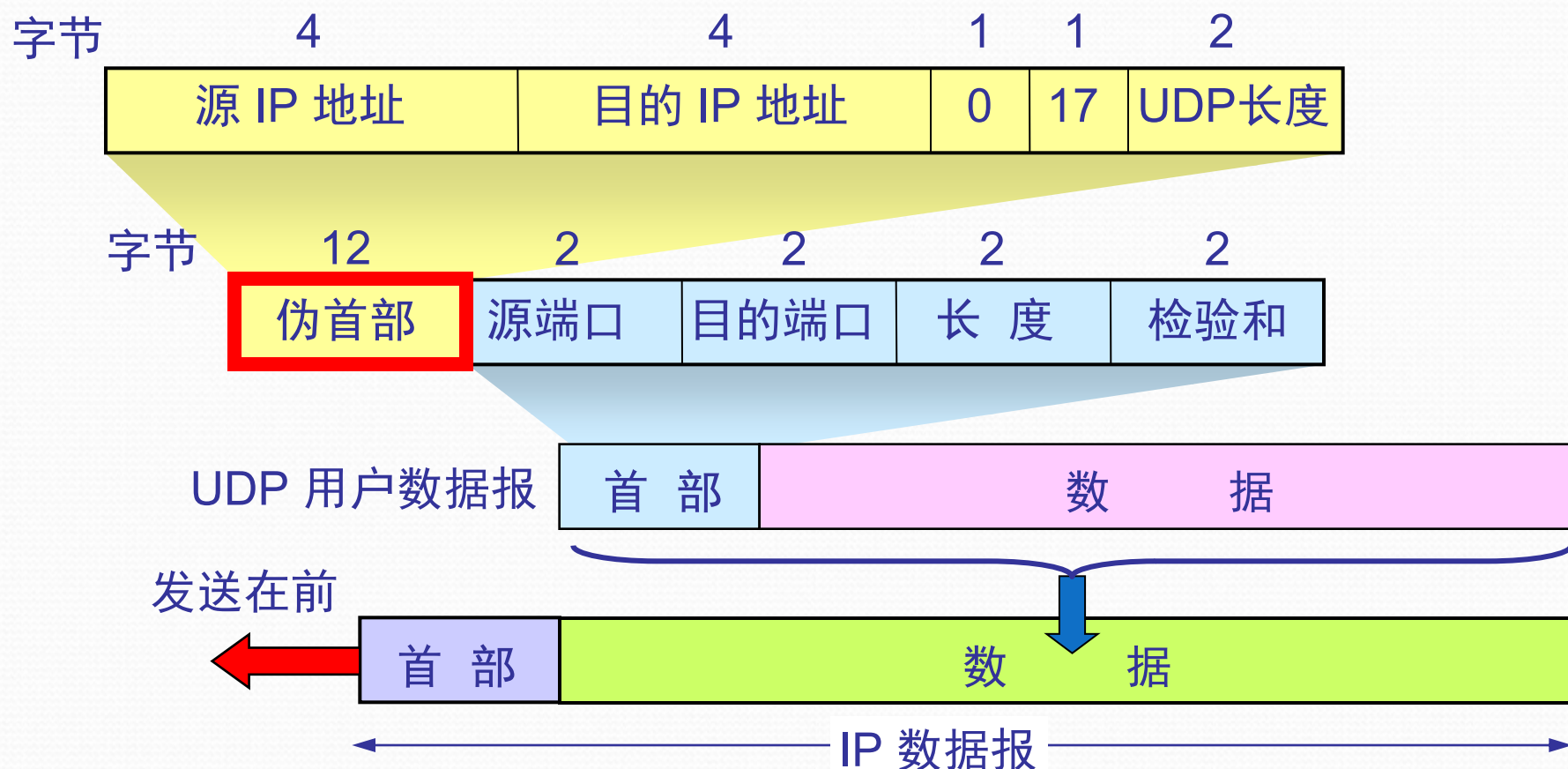
UDP 基于端口的分用



用户数据报 UDP 有两个字段：数据字段和首部
字段。首部字段有 8 个字节，由 4 个字段组成，
每个字段都是两个字节。



在计算检验和时，临时把“伪首部”和 UDP 用户数据报连接在一起。伪首部仅仅是为了计算检验和。



计算 UDP 检验和的例子

12 字节 伪首部	153.19.8.104			
	171.3.14.11			
	全 0	17	15	
8 字节 UDP 首部	1087		13	
	15		全 0	
7 字节 数据	数据	数据	数据	数据
	数据	数据	数据	全 0

填充

10011001 00010011 → 153.19
 00001000 01101000 → 8.104
 10101011 00000011 → 171.3
 00001110 00001011 → 14.11
 00000000 00010001 → 0 和 17
 00000000 00001111 → 15
 00000100 00111111 → 1087
 00000000 00001101 → 13
 00000000 00001111 → 15
 00000000 00000000 → 0 (检验和)
 01010100 01000101 → 数据
 01010011 01010100 → 数据
 01001001 01001110 → 数据
 01000111 00000000 → 数据和 0 (填充)

按二进制反码运算求和 10010110 11101101 → 求和得出的结果
 将得出的结果求反码 01101001 00010010 → 检验和



传输控制协议 TCP 概述

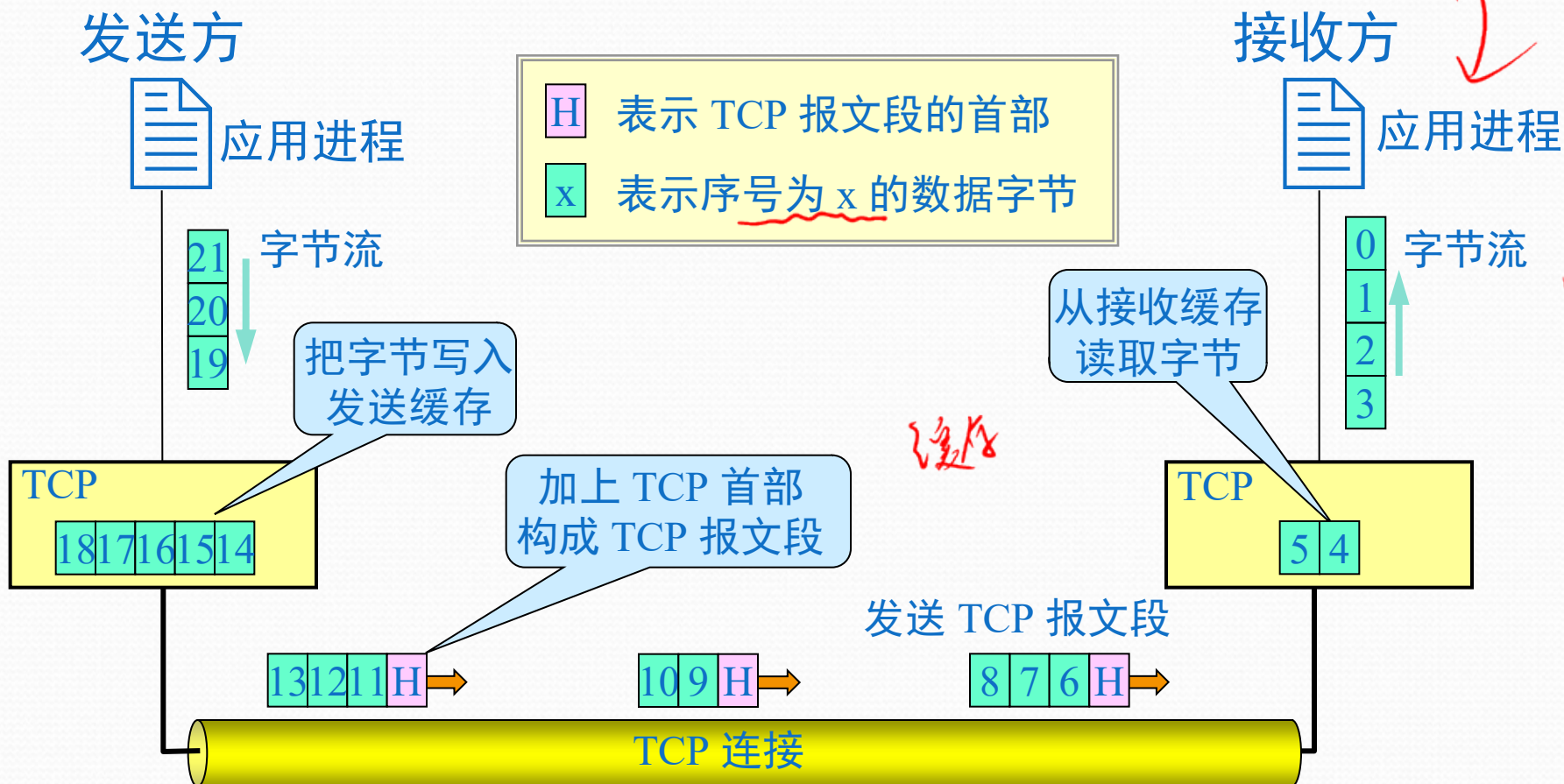
TCP 最主要的特点

- TCP 是面向连接的运输层协议。
- 每一条 TCP 连接只能有两个端点(endpoint), 每一条 TCP 连接只能是点对点的（一对一）。
- TCP 提供可靠交付的服务。
- TCP 提供全双工通信。
- 面向字节流。

TCP 面向流的概念

1 2 3

2





应当注意

- TCP 连接是一条虚连接而不是一条真正的物理连接。
- TCP 对应用进程一次把多长的报文发送到TCP 的缓存中是不关心的。
- TCP 根据对方给出的窗口值和当前网络拥塞的程度来决定一个报文段应包含多少个字节（UDP 发送的报文长度是应用进程给出的）。
- TCP 可把太长的数据块划分短一些再传送。TCP 也可等待积累有足够多的字节后再构成报文段发送出去。



TCP 的连接

- TCP 把连接作为最基本的抽象。
- 每一条 TCP 连接有两个端点。
- TCP 连接的端点不是主机，不是主机的IP 地址，不是应用进程，也不是运输层的协议端口。TCP 连接的端点叫做套接字(socket)或插口。
- 端口号拼接到(contatenated with) IP 地址即构成了套接字。



套接字 (socket)

套接字 socket = (IP地址: 端口号) ^{端口} (5-1)

- 每一条 **TCP** 连接唯一地被通信两端的两个端点（即两个套接字）所确定。即：

TCP 连接 ::= {socket1, socket2}
= {(IP1: port1), (IP2: port2)} (5-2)



同一个名词 socket 有多种不同的意思

- 应用编程接口API称为 socket API, 简称为 socket。
- socket API 中使用的一个函数名也叫作 socket。
- 调用 socket 函数的端点称为 socket。
- 调用 socket 函数时其返回值称为 socket 描述符, 可简称为 socket。
- 在操作系统内核中连网协议的 Berkeley 实现, 称为 socket 实现。



本讲总结

运输层协议概述
用户数据报协议UDP
传输控制协议TCP



作业

- 5-09, 5-11