



西北師範大學
NORTHWEST NORMAL UNIVERSITY

计算机网络

第五章 运输层 (1)

陈旺虎

chenwh@nwnu.edu.cn



第5章 运输层

5.1 运输层协议概述

5.2 运输层协议概述

5.3 传输控制协议 **TCP** 概述

5.4 可靠传输的工作原理

4.5 **TCP** 报文段的首部格式

5.6 **TCP** 可靠传输的实现

5.7 **TCP**的流量控制

5.8 **TCP**的拥塞控制

5.9 **TCP**的连接管理

5.1 运输层协议概述

- 从通信和信息处理的角度看
 - ▣ 运输层向应用层提供通信服务
 - ▣ 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
 - ▣ 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

5.1.1 进程之间的通信

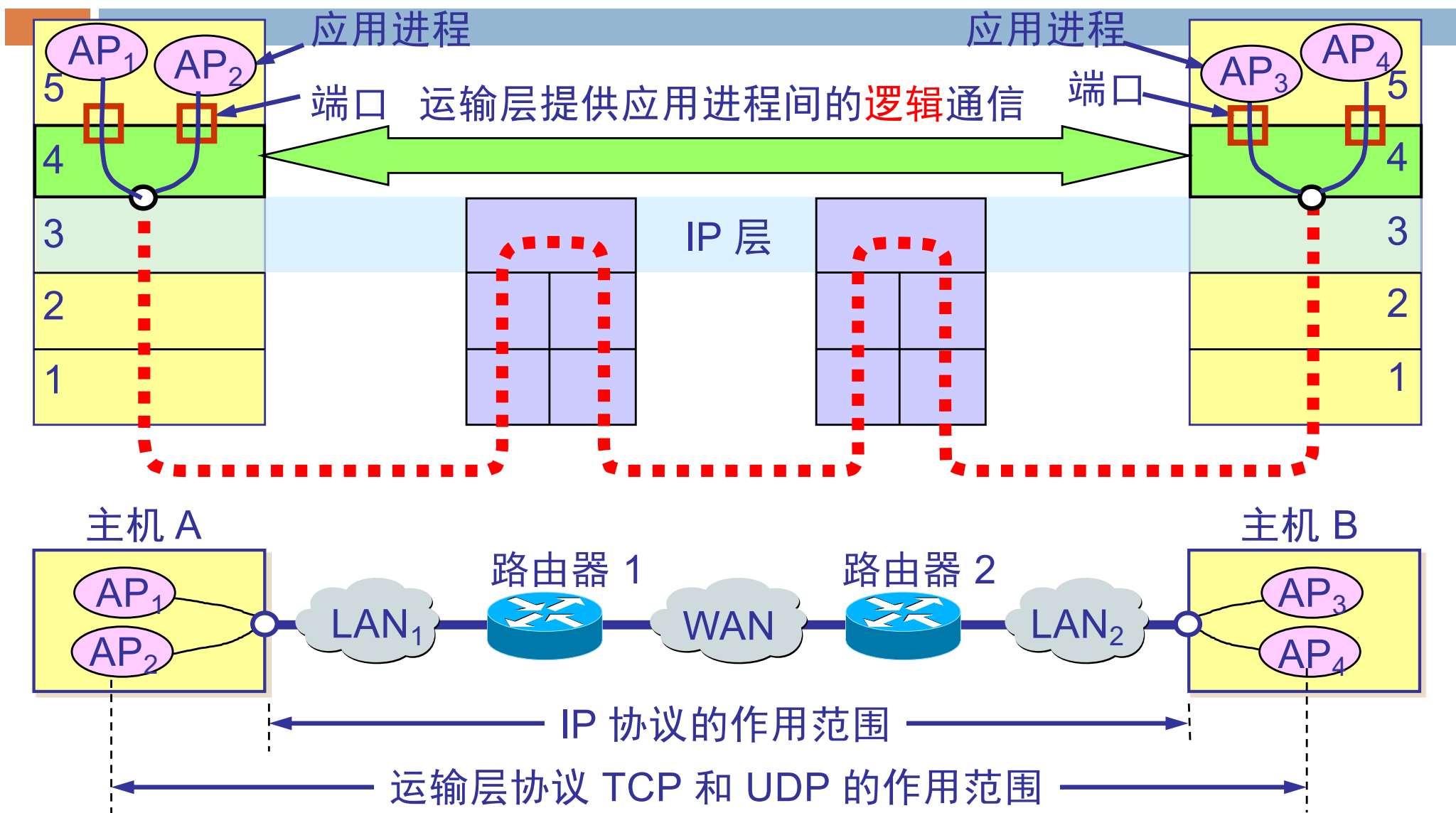
□ 端到端的通信

- ▣ 两个主机进行通信，实际上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

- ▣ 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。

运输层为相互通信的应用进程提供了逻辑通信



运输层协议和网络层协议的主要区别



运输层的主要功能

- 为应用进程之间提供端到端的逻辑通信
- 对收到的报文进行差错检测
- 运输层向高层用户屏蔽了网络层的细节
 - ▣ 如网络拓扑、所采用的路由选择协议等
 - ▣ 使得从应用进程的角度来看，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

两种不同的运输协议

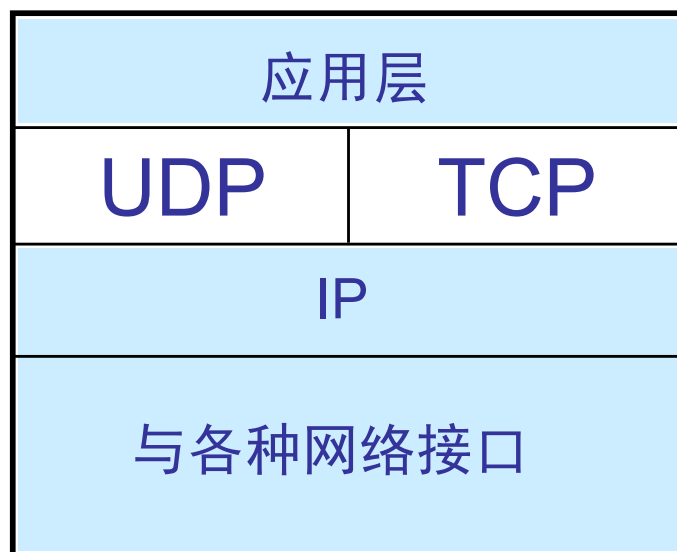
- 运输层提供两种不同的运输协议
 - ▣ 面向连接的 TCP
 - ▣ 无连接的 UDP
- 当运输层采用面向连接的 TCP 协议时
 - ▣ 尽管下面的网络是不可靠的，但这种逻辑通信信道就相当于一条全双工的可靠信道
- 当运输层采用无连接的 UDP 协议时
 - ▣ 这种逻辑通信信道是一条不可靠信道

5.1.2 运输层的两个主要协议

- 用户数据报协议 UDP
User Datagram Protocol
- 传输控制协议 TCP
Transmission Control Protocol
- 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 传送的数据单位是 TCP 报文段(segment)
 - UDP 传送的数据单位是UDP报文或用户数据报

TCP/IP 体系中的运输层协议

运输层



TCP&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

- ▣ 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：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。

5.1.3 TCP/IP体系中应用进程的标识

- 应用层的进程标识？
- 与计算机系统进程标识的区别
 - ▣ 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
 - ▣ 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

应用进程标识面临的困难

□ 动态性

- ▣ 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
- ▣ 有时我们会改换接收报文的进程，但并不需要通知所有发送方

- 通常，利用目的主机提供的功能来识别通信的终点，而不需要知道实现这个功能的进程

端口号 (protocol port number)

- 解决上述问题的方法
 - ▣ 在运输层使用协议端口号(protocol port number), 简称端口(port)
- 虽然通信的终点是应用进程，但我们可以把端口想象为通信的终点
 - ▣ 因为，我们只要把要传送的报文交到目的主机的某一个合适的目的端口，剩下的工作（即最后交付目的进程）就由运输层的协议来完成

软件端口与硬件端口

- 在协议栈层间的抽象的协议端口是软件端口。
 - ▣ 应用层的各种协议进程与运输实体进行层间交互的一种地址
- 路由器或交换机上的端口是硬件端口
 - ▣ 不同硬件设备进行交互的接口

TCP/IP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
 - ▣ 只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没有联系。

三类端口

□ 服务器端使用的端口号

- 熟知端口：0~1023

- 登记端口号：1024~49151

- 为没有熟知端口号的应用程序使用，必须在 IANA 登记，以防止重复

□ 客户端使用的端口号

- 49152~65535

- 客户进程选择暂时使用，通信结束后，这个端口号可供其他客户进程以后使用。(短暂端口号)

5.2 用户数据报协议UDP

5.2.1 UDP概述

- UDP只在IP的数据报服务之上增加了少量功能
 - ▣ (1) 端口的功能；
 - ▣ (2) 差错检测的功能
- 特点
 - ▣ 无连接，发送数据之前不需要建立连接
 - ▣ 使用尽最大努力交付
 - ▣ 面向报文的
 - ▣ 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

面向报文

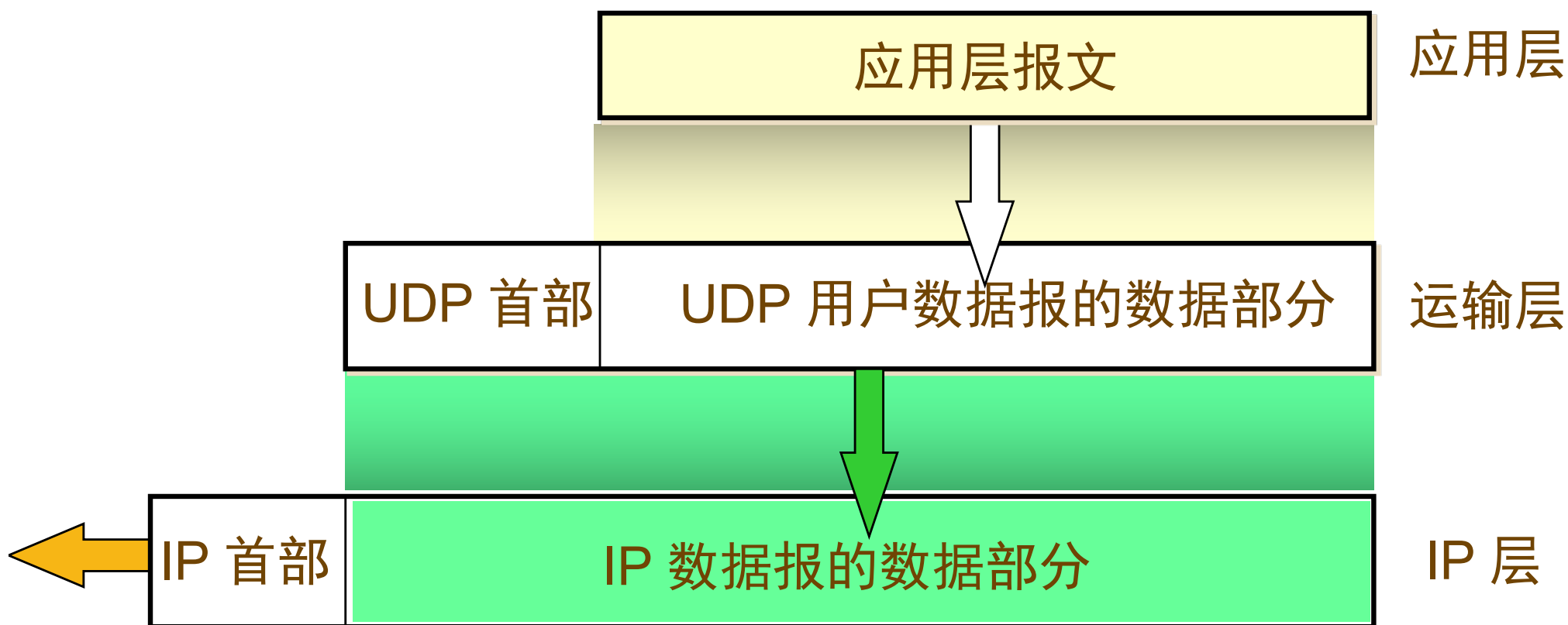
□ 发送方UDP

- 对应用程序交下来的报文，在添加首部后便交付 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

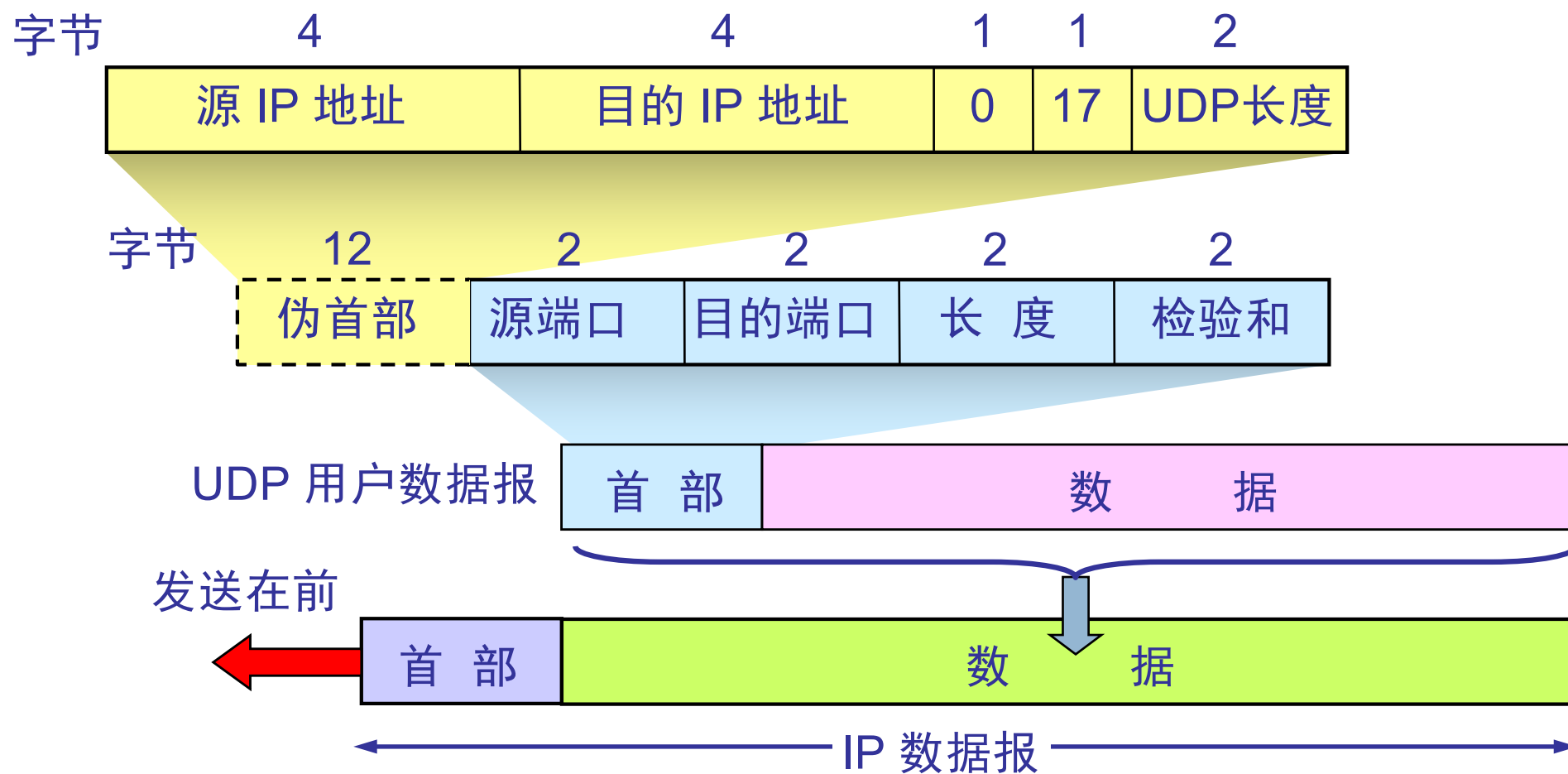
□ 接收方UDP

- 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

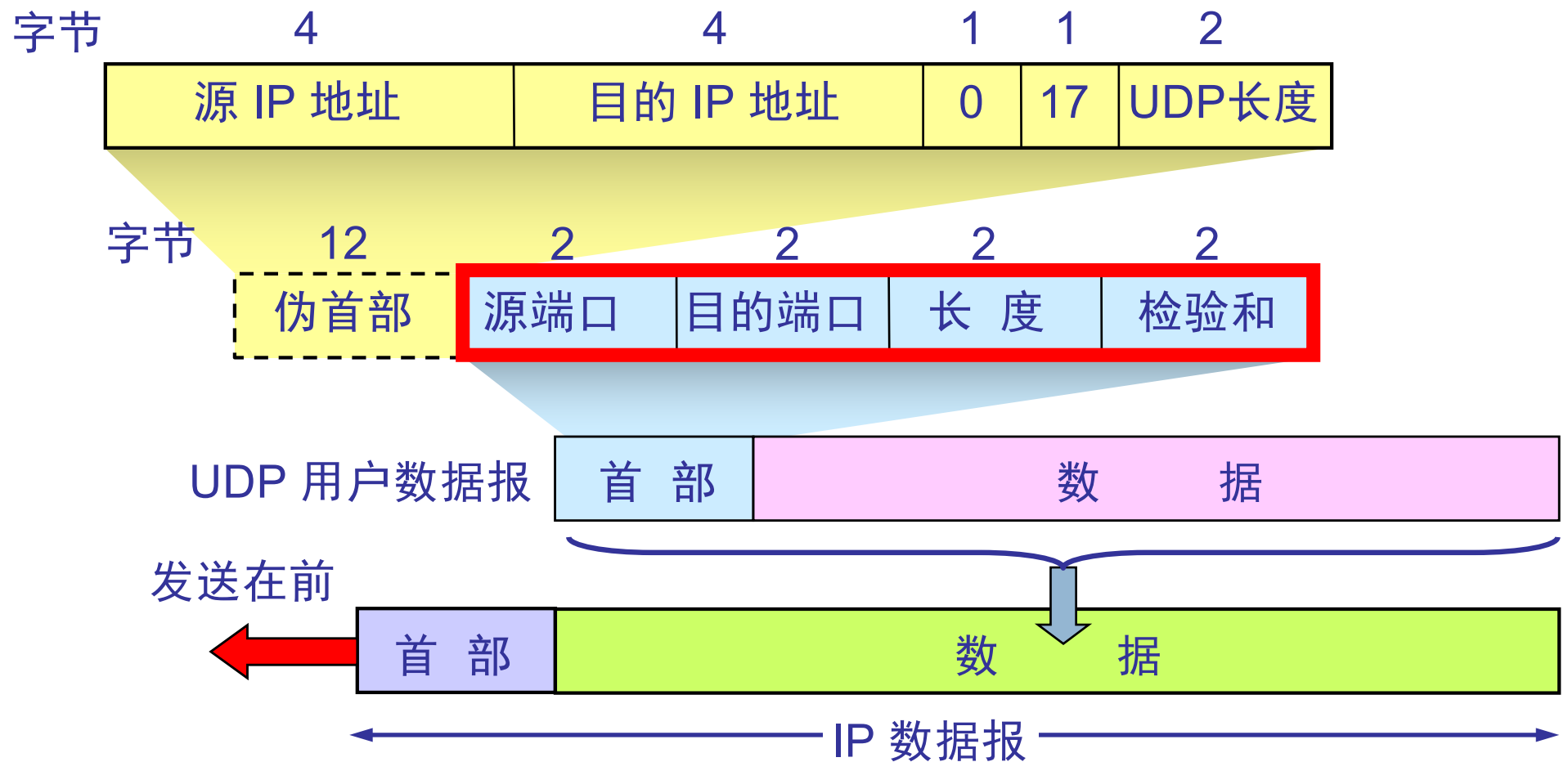
□ 应用程序必须选择合适大小的报文



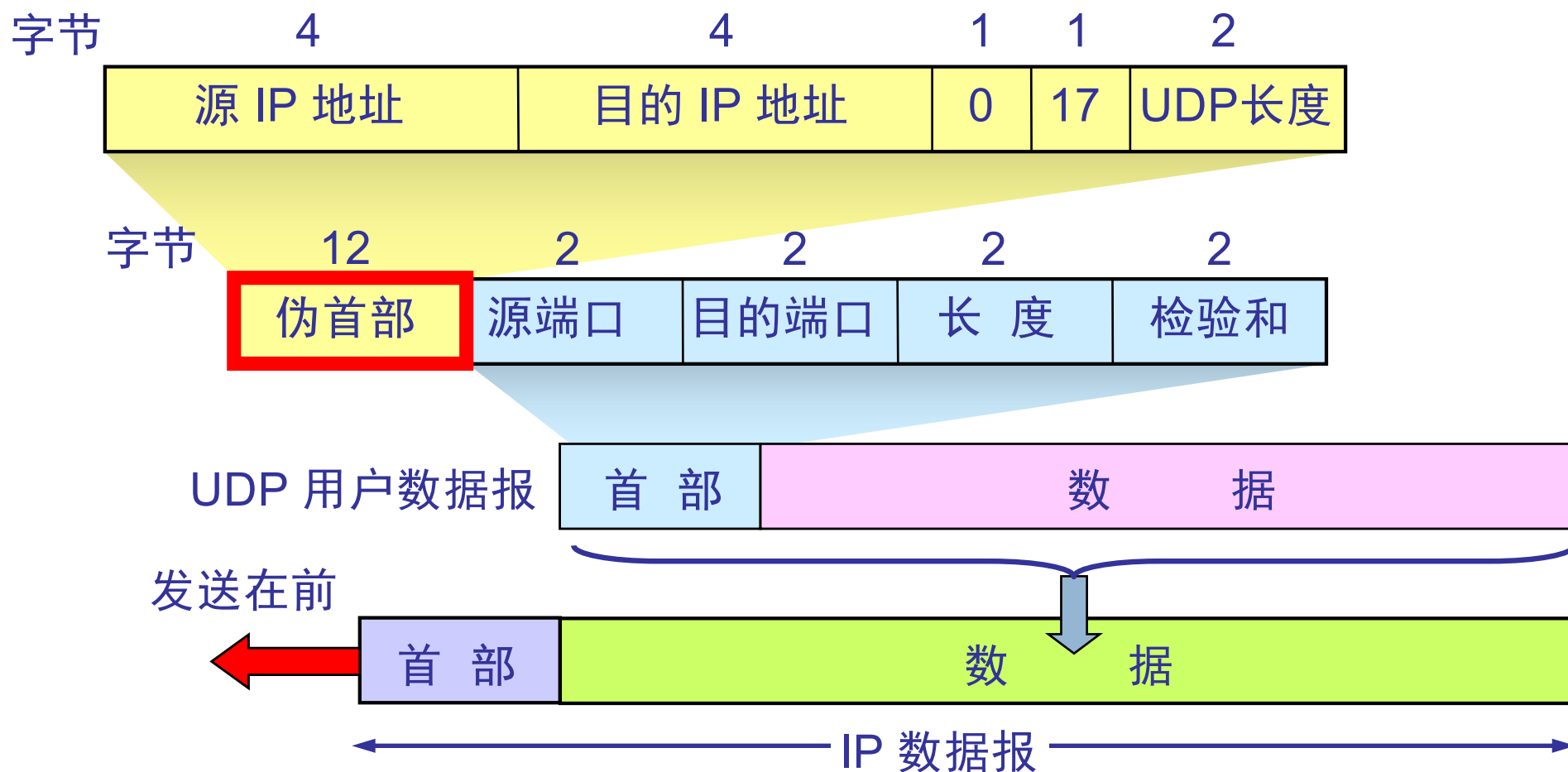
5.2.2 UDP的首部格式



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。



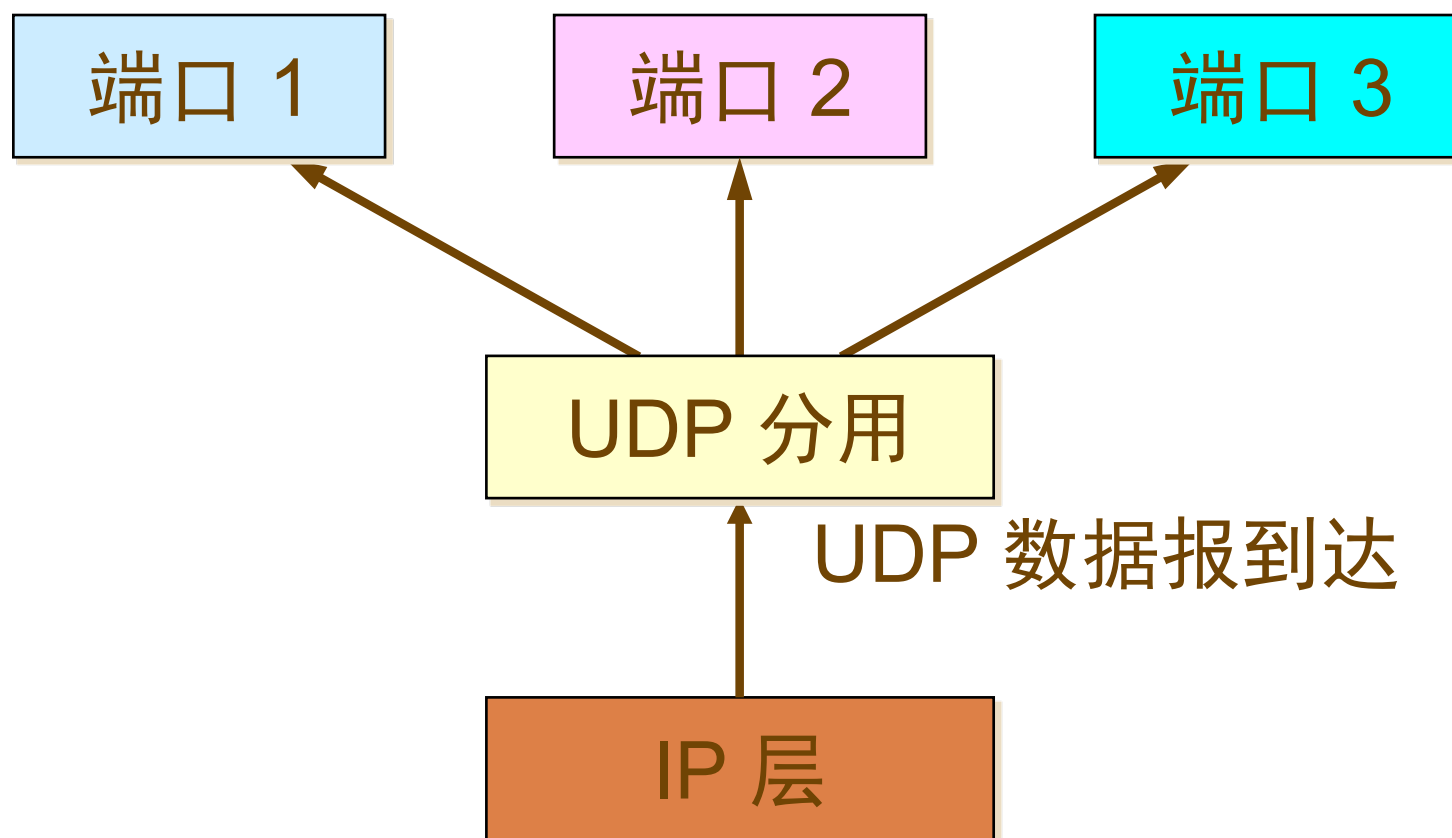
在计算检验和时，临时把“伪首部”和 UDP 用户数据报连接在一起。伪首部仅仅是为了计算检验和。



思考

- 为什么要增加伪首部用以校验？
 - ▣ 伪首部包含IP首部的一些字段
 - ▣ 可让UDP两次检查数据是否已经到达目的地，以及IP层是否正确地传输了数据

UDP基于端口的分用



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)

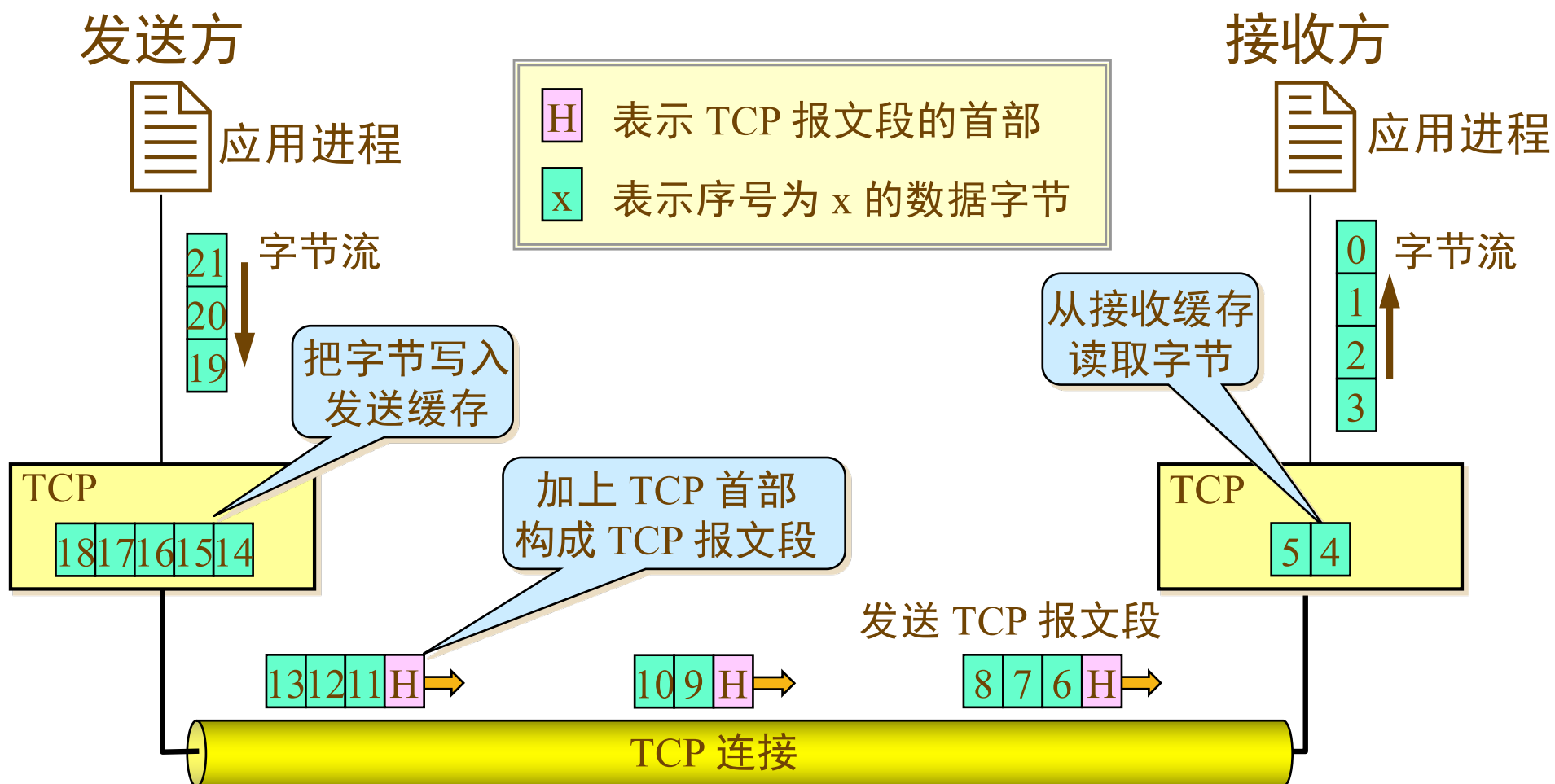
按二进制反码运算求和 10010110 11101101 → 求和得出的结果
 将得出的结果求反码 01101001 00010010 → 检验和
 接收方检验时, 结果全1则为正确报文

5.3 传输控制协议 TCP

5.3.1 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面向字节流的概念



应当注意

- TCP 连接是一条虚连接，而非真正的物理连接
- 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。

5.3.2 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IP 地址即构成了套接字。

套接字 (socket)

套接字 `socket` = (IP地址: 端口号)

- 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

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

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

Thanks, Any Question?

The End.

chenwh@nwnu.edu.cn

