



西北師範大學
NORTHWEST NORMAL UNIVERSITY

计算机网络

第五章 运输层 (2)

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

Review



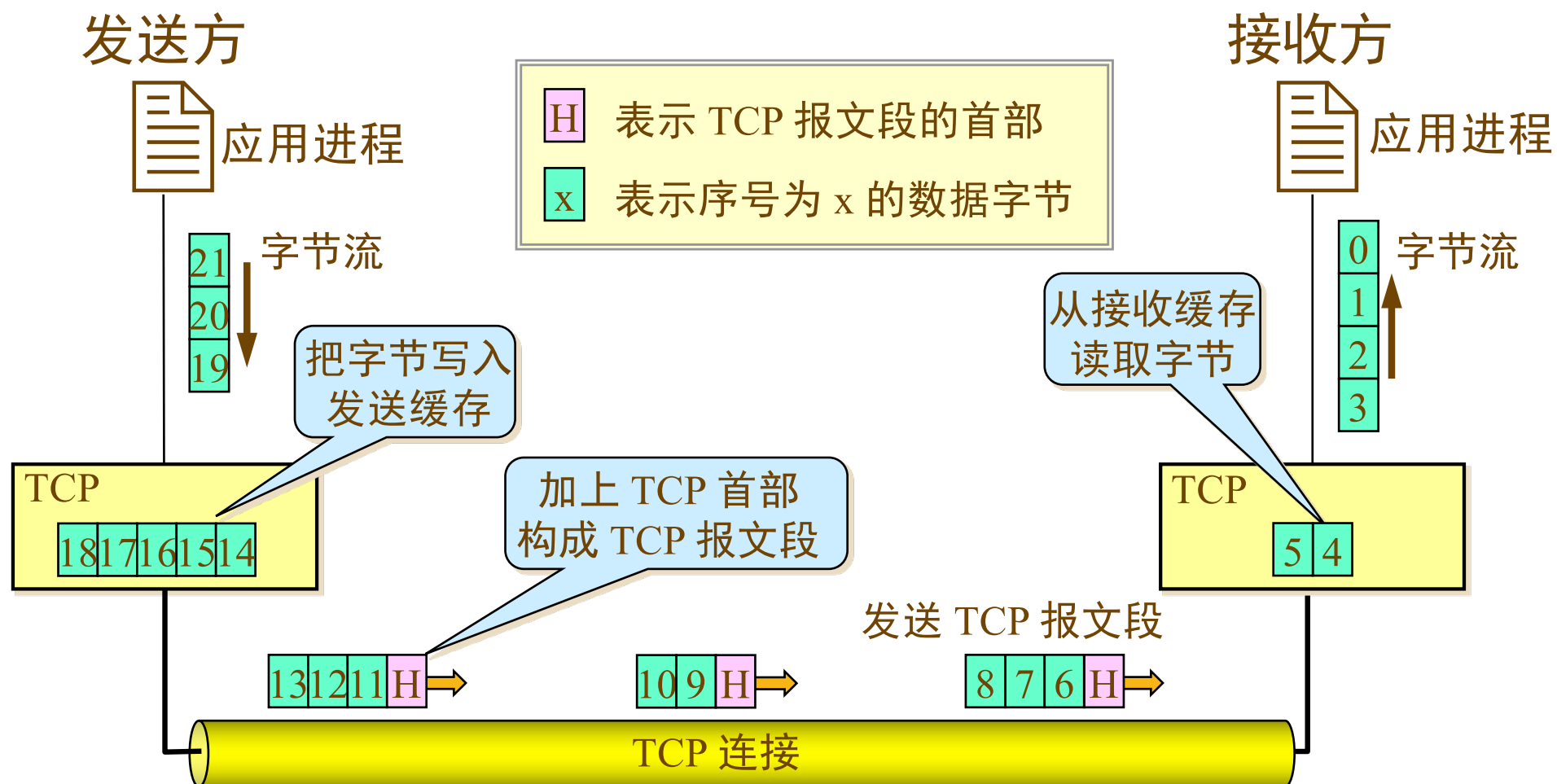
- 运输层的功能
- UDP协议
- 网络中进程的标识方法

5.3 传输控制协议TCP

5.3.1 TCP最主要的特点

- 面向连接
 - ▣ 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
 - ▣ 端口号拼接上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。即：

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

同一个名词socket，多种含义

- 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**

5.4 可靠传输的工作原理

- 理想化的数据传输中，协议的对应层无需特殊的措施

假定1

- 链路是理想的传输信道，数据既不会出错也不会丢失

假定2

- 不管发送方的速率多快，接收方总能接收到并上交

- 实际情况并非如此

- 传输中可能出错

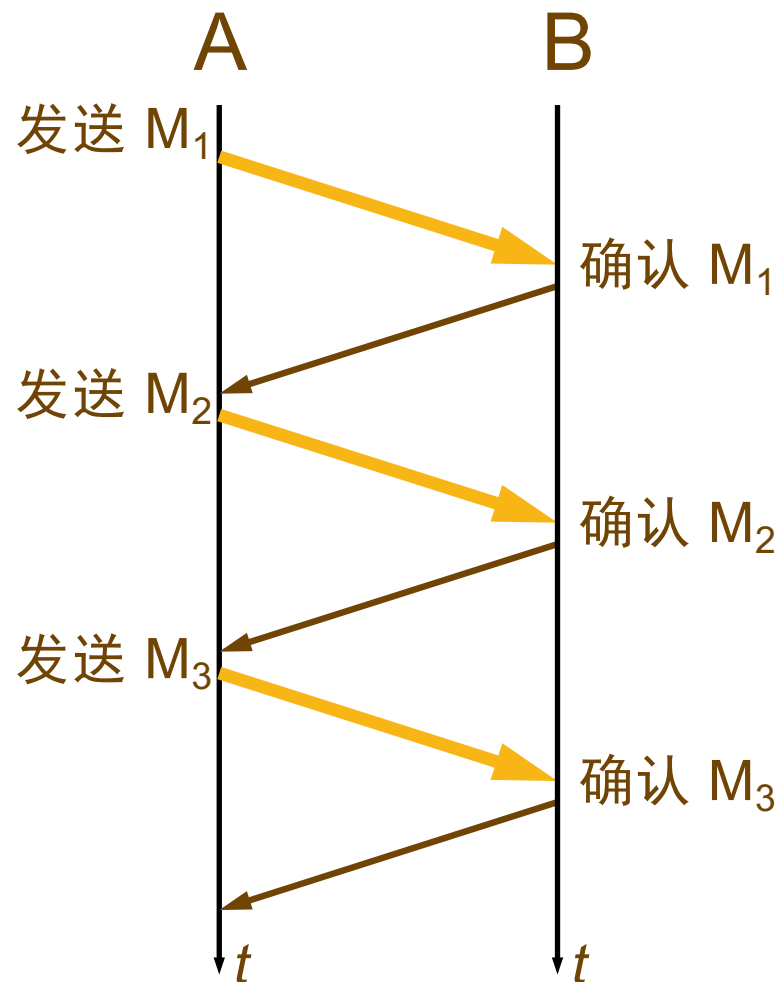
- 可能发生接收缓存的溢出和数据帧丢失

- 接收方对应层的缓存每存满一帧就向接收方主机交付一帧。
 - 如果没有专门的流量控制协议，则接收方并没有办法控制发送方的发送速率；而接收方又无法做到其向主机交付数据的速率始终高于发送方的发送数据的速率。

5.4.1 停止等待协议

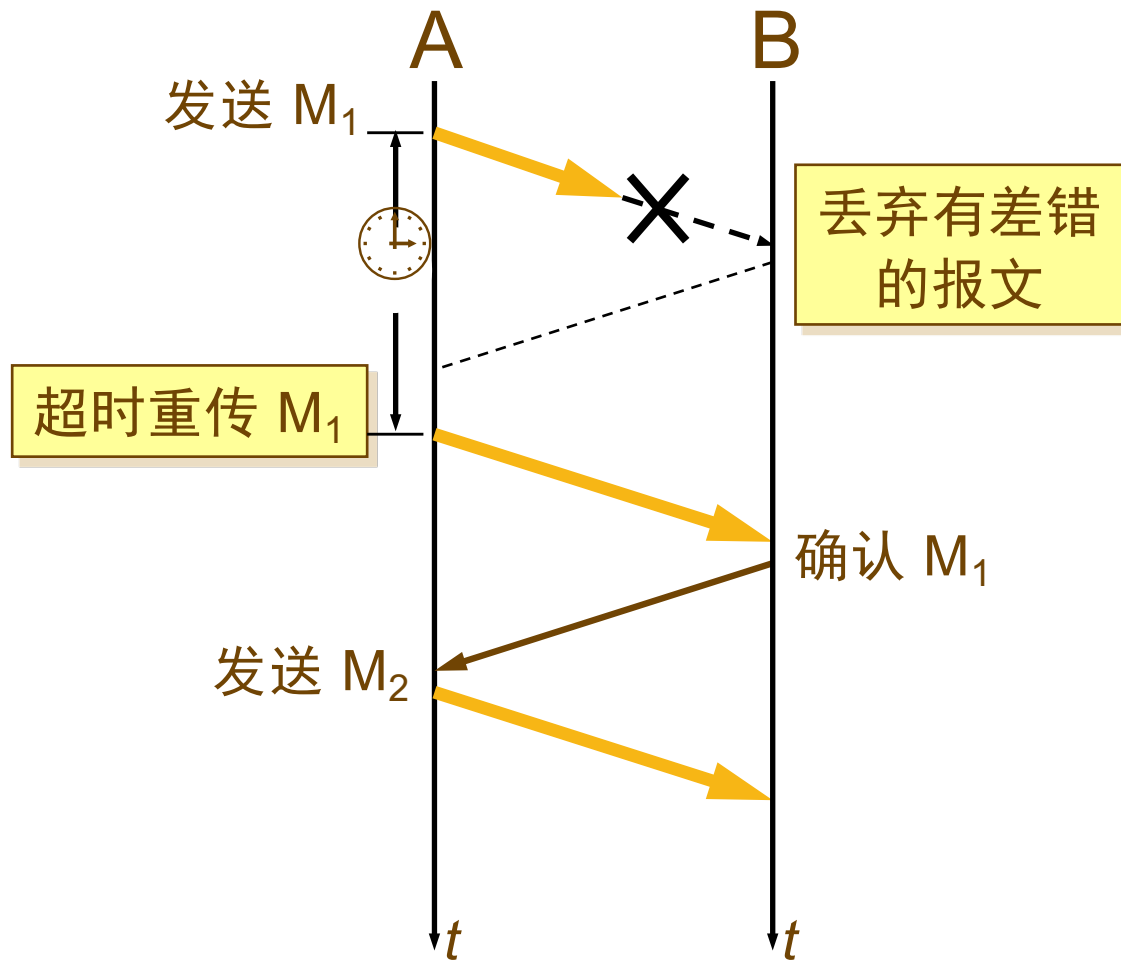
- 如果去掉假定2，保留假定1，即：
 - 无差错的理想信道
 - 不能保证接收端向主机交付数据的速率永远不低于发送端发送数据的速率
- 如何使接收方的接收缓存任何情况下都不溢出？

具有简单流量控制的停等协议



(a) 无差错情况

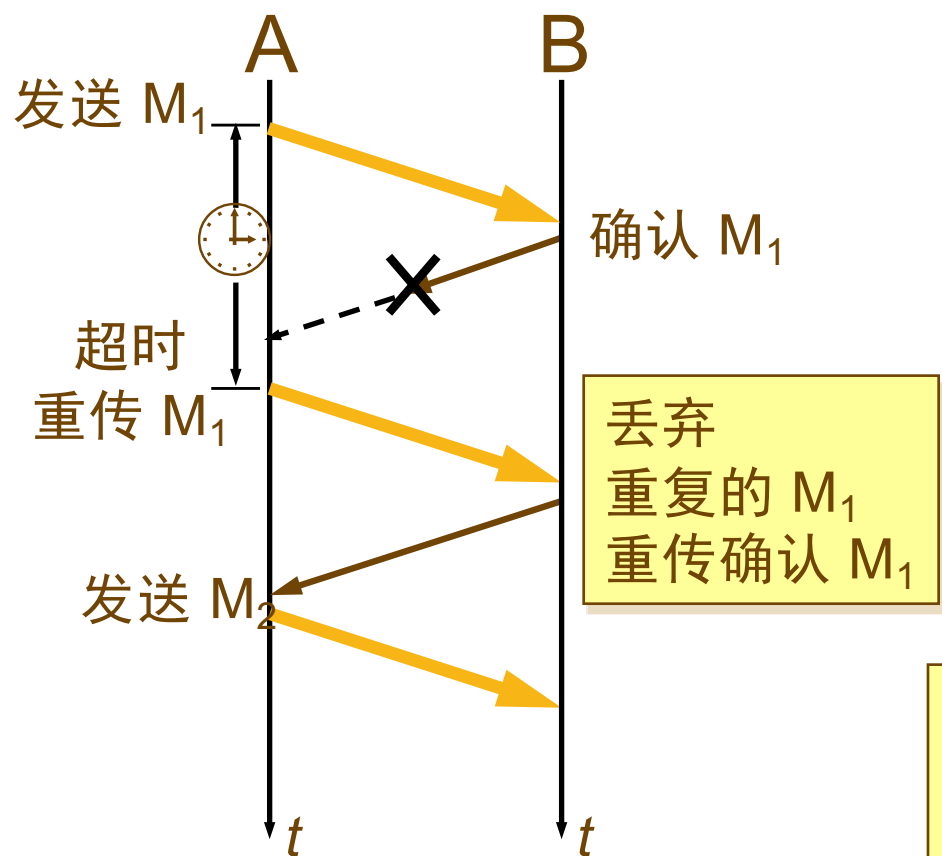
实用的停止等待协议



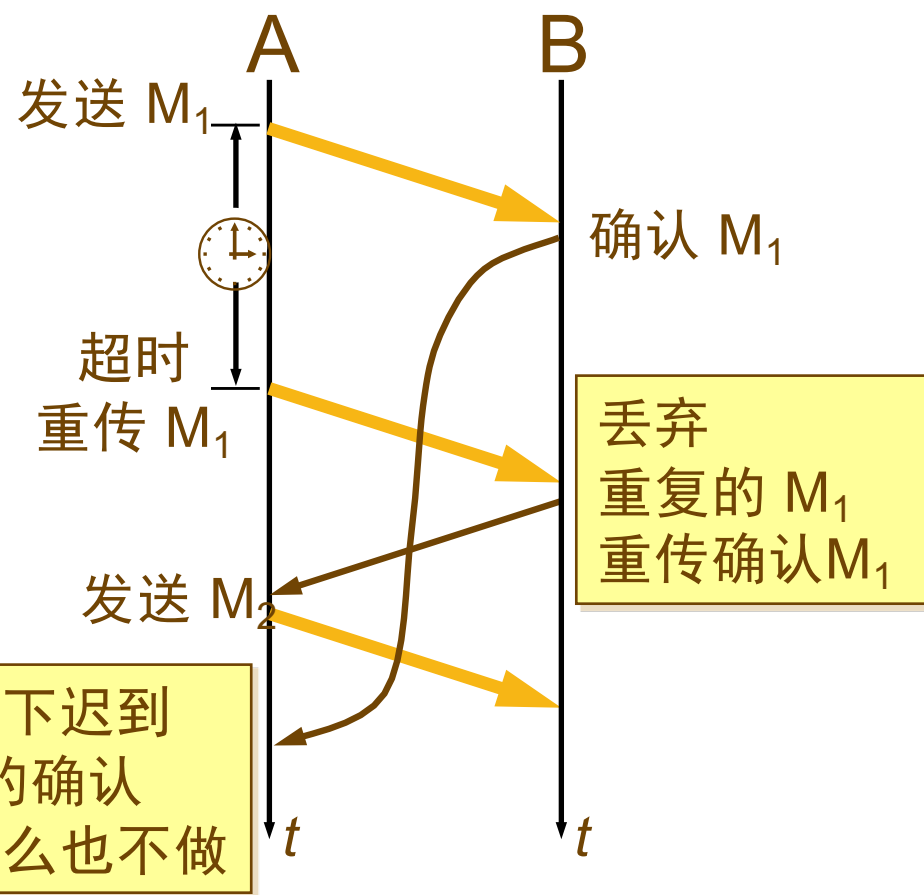
(b) 超时重传

- 去掉假定一和假定二
 - 可能出错、丢失
 - 缓存可能溢出
- 实用的停止等待协议
 - 设置超时自动重传机制
- 重传时间一般设定为略大于RTT

确认丢失和确认迟到



(a) 确认丢失（出现重复数据帧）



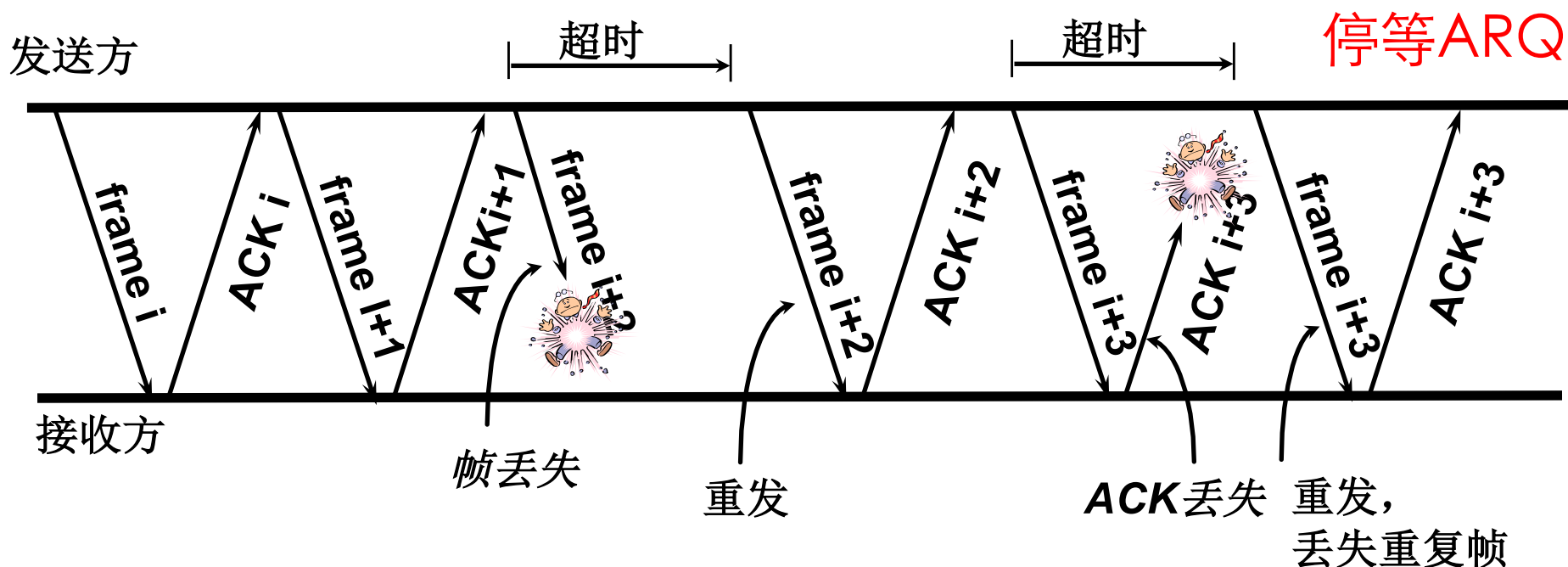
(b) 确认迟到（出现重复确认帧）

注意

- 接收方如何确定重复帧？
- (1) 分组和确认分组需要编号
 - ▣ 由于每发送一个数据帧就停止等待应答，接收方只需区分是前一分组还是后一分组，故只需1位比特作为序号即可。
 - ▣ 0和1交替标号
 - ▣ 确认分组可以不编号，但若超时时间设置较短则会出现发送方收到两个确认帧，最好也编号
- (2) 发送完一个分组后，必须暂时保留已发送的分组的副本

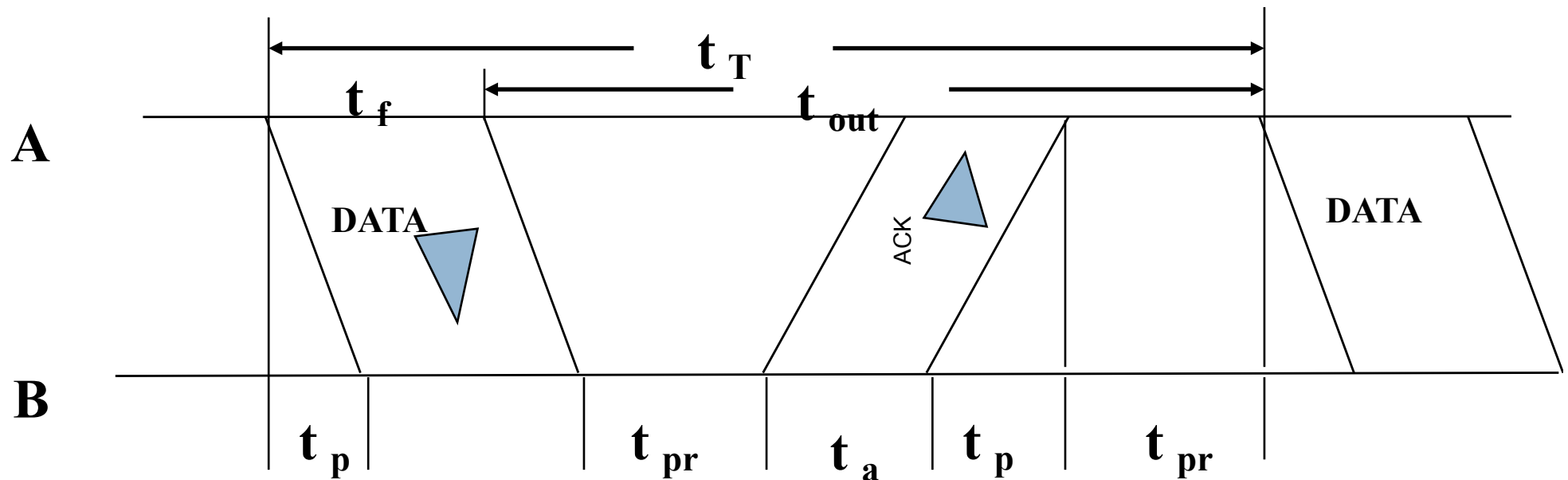
可靠通信的实现

- 使用上述确认和重传机制，可以在不可靠的传输网络上实现可靠的通信
- 这种可靠传输协议常称为**自动重传请求**ARQ
- 重传的请求是自动进行的，接收方不需要请求发送方重传某个出错的分组



停止等待协议中的几个时间关系

- 假设A向B发送分组，而B只发送确认信息不发送分组



- t_T 两个发送成功的分组之间的最小时间间隔
- t_f 是一个分组的发送时间； t_p 表示分组的传播时延
- t_{pr} 处理时间； t_a 结点B发送确认分组的发送时间
- t_{out} 结点A的重传时间

- 设数据帧的长度为 L_f ，数据发送速率为 C ，则：

$$t_f = L_f / C$$

$$t_{out} = t_p + t_{pr} + t_a + t_p + t_{pr}$$

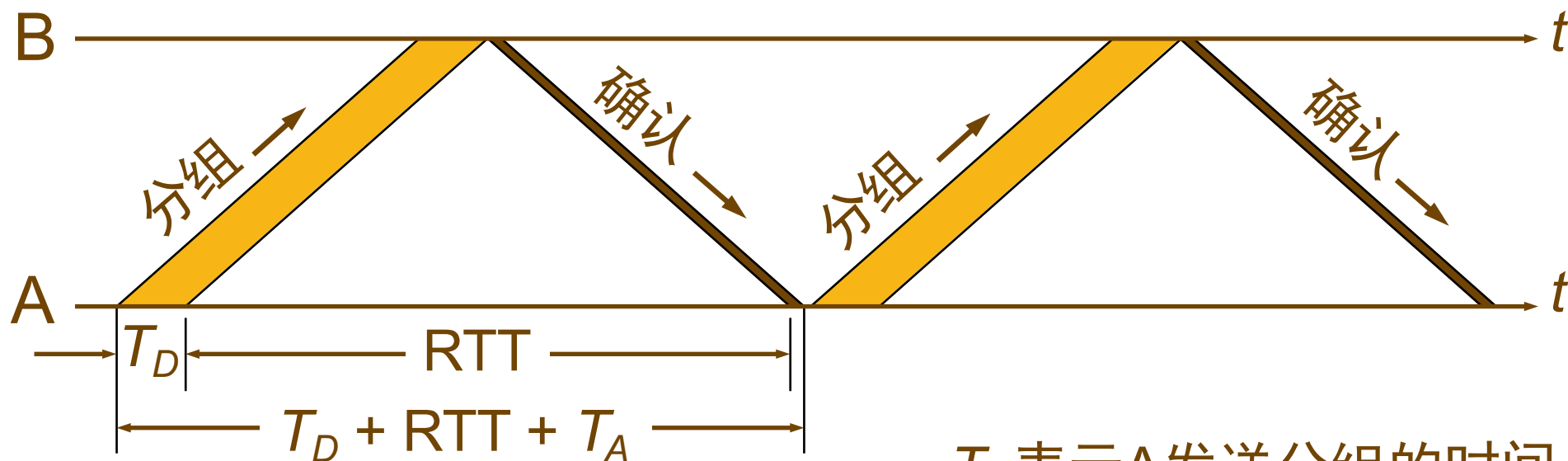
- 忽略 t_{pr} ，则

$$t_{out} = 2t_p + t_a$$

$$t_T = t_f + t_{out} = t_f + 2t_p + t_a$$

信道利用率

- 停止等待协议的优点是简单，但缺点是信道利用率太低



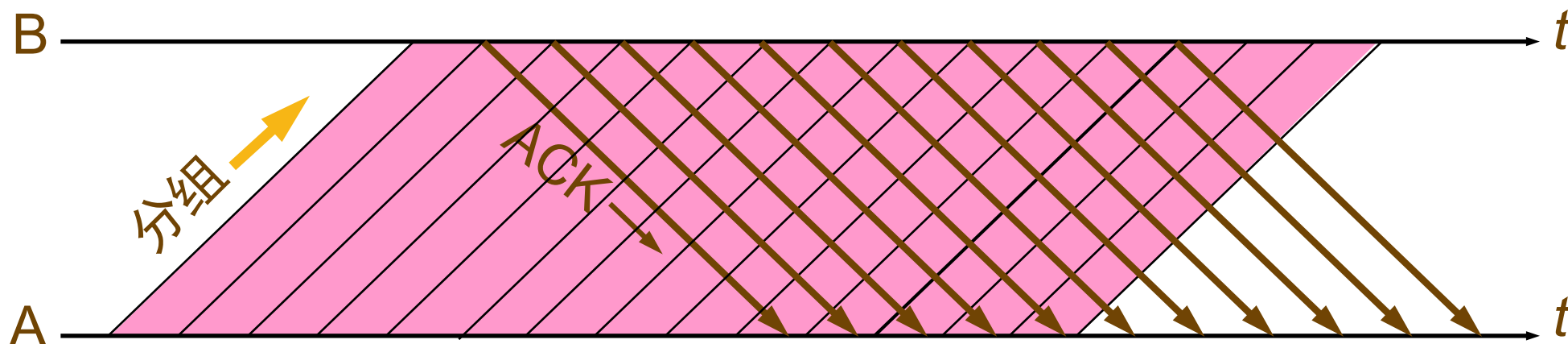
- T_D 表示A发送分组的时间
- T_A 表示B发送确认分组的时间
- A经过时间 $T_D + RTT + T_A$ 后，发送下一分组

信道的利用率 U

$$U = \frac{T_D}{T_D + \text{RTT} + T_A}$$

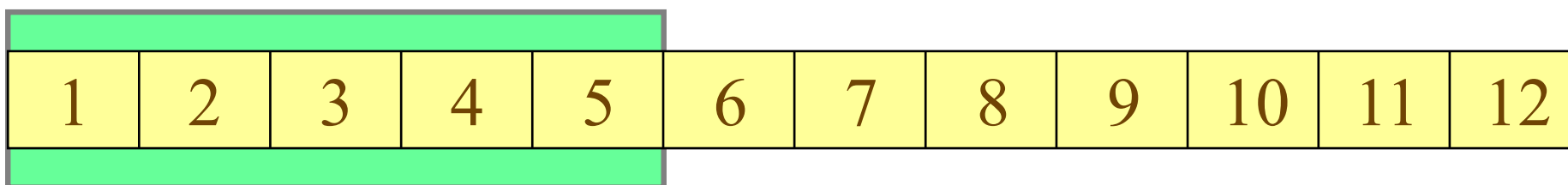
流水线传输

- 发送方可连续发送多个分组，不必每发完一个分组就停顿下来等待对方的确认
- 由于信道上一一直有数据不间断地传送，这种传输方式可获得很高的信道利用率。



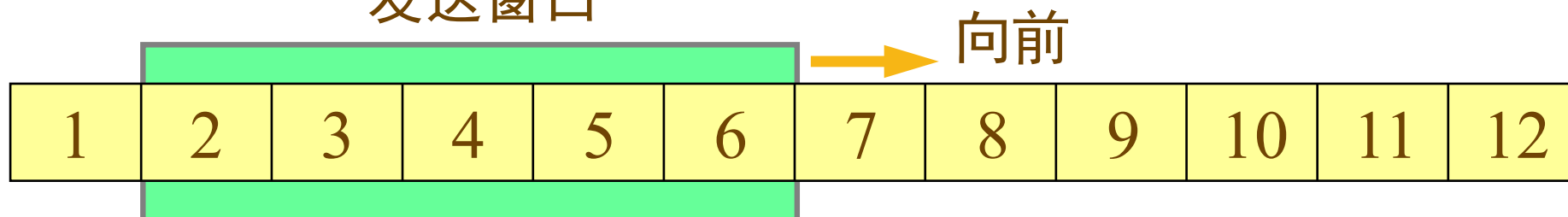
5.4.2 连续ARQ协议

发送窗口



(a) 发送方维持发送窗口（发送窗口是 5）

发送窗口



(b) 收到一个确认后发送窗口向前滑动

累积确认

- 接收方对按序到达的最后一个分组发送确认
 - ▣ 表示到这个分组为止的所有分组都已正确收到
- 优点
 - ▣ 容易实现，即使确认丢失也不必重传
- 缺点
 - ▣ 不能向发送方反映出接收方已经正确收到的所有分组的信息

Go-back-N（回退 N）

- 例如：
 - ▣ 如果发送方发送了前 5 个分组，而中间的第 3 个分组丢失了
 - ▣ 这时接收方只能对前两个分组发出确认
 - ▣ 发送方无法知道后面三个分组的下落，只好把后面的三个分组都再重传一次
- Go-back-N（回退 N）：回退，重传已发送过的 N 个分组
- 当通信线路质量不好时，连续 ARQ 协议会带来负面的影响

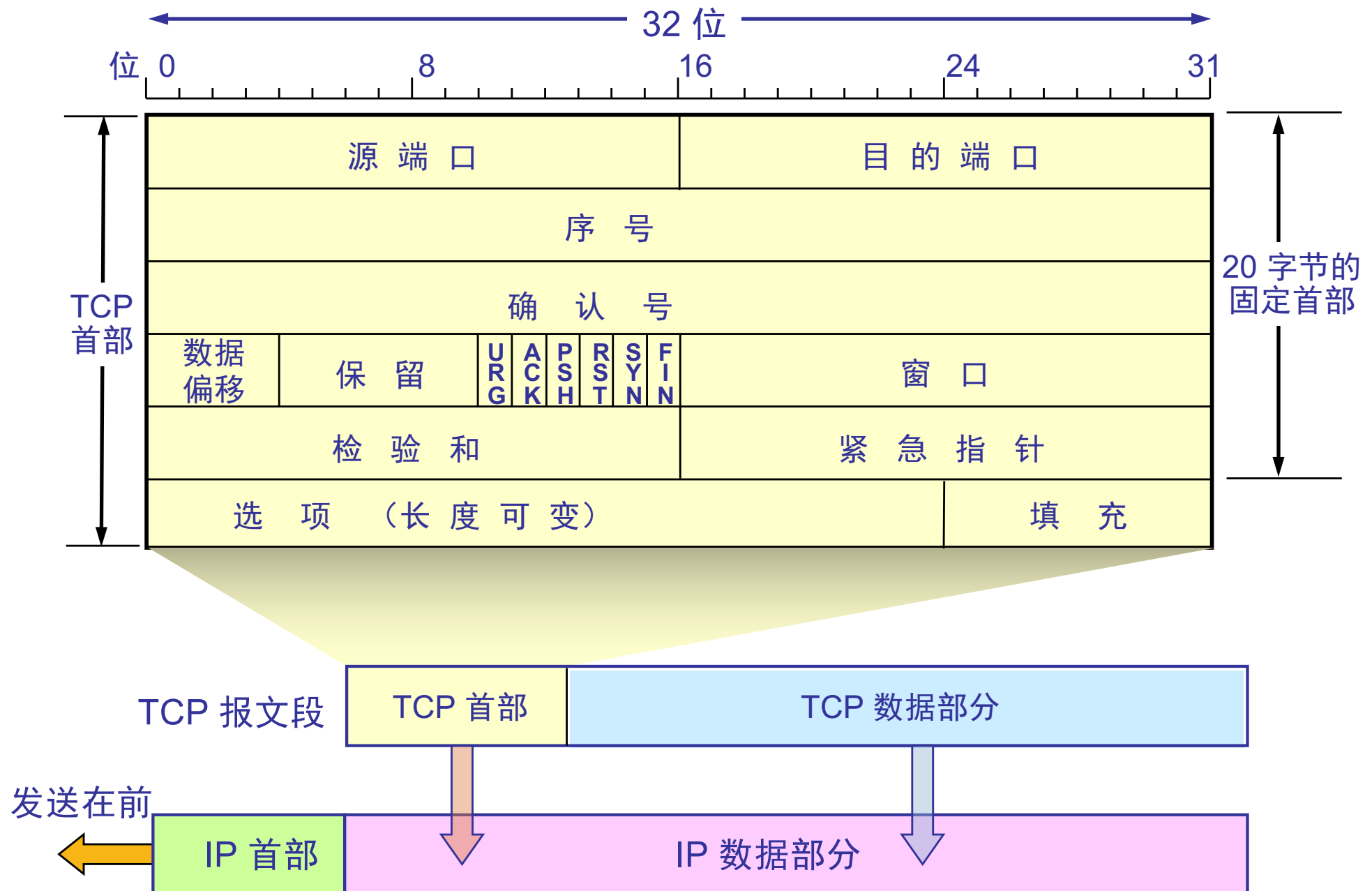
连续ARQ协议原理

- 发送方发完一个分组后，不必停下来等待对方的应答，可以连续发送分组；
- 如果在发送过程中收到接收方的肯定应答，可以继续发送；
- 若收到对其中某一分组的否认分组，则从该分组开始的后续分组全部重发

说明

- 结点**B**接收到出错分组丢弃后有两种处理方法:
 - (1) 向结点**A**发送否认分组, 结点**A**重发出错分组以后所有分组
 - (2) 结点**B**不做任何响应, 只等待**A**结点超时重传即可
 - 方式 (2) 使协议较简单, 使用较多

5.5 TCP 报文段的首部格式





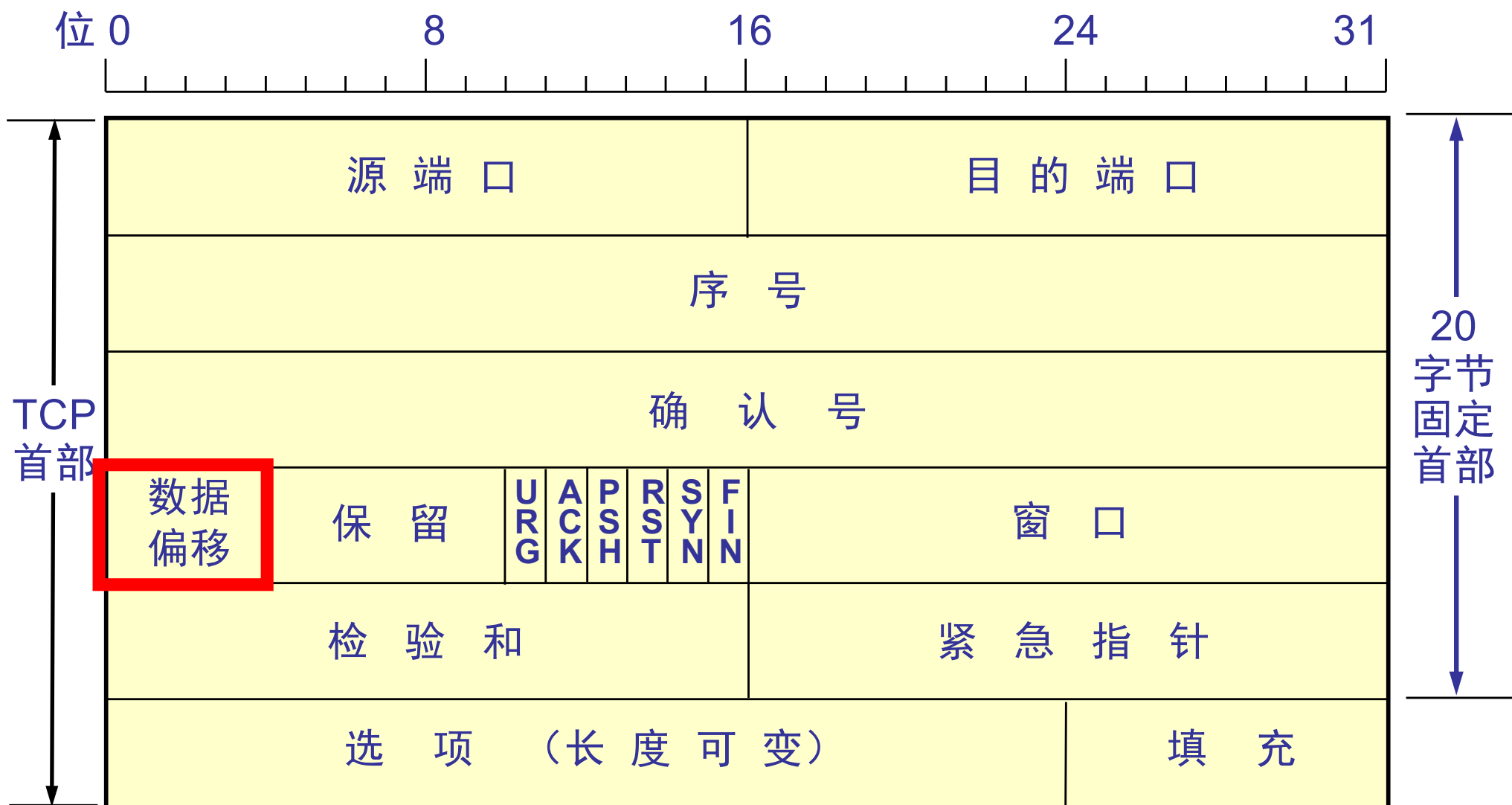
源端口和目的端口字段——各占 2 字节。端口是运输层与应用层的服务接口。运输层的复用和分用功能都要通过端口才能实现。



序号字段——占 4 字节。TCP 连接中传送的数据流中的每一个字节都编上一个序号。序号字段的值则指的是本报文段所发送的数据的第一个字节的序号。



确认号字段——占 4 字节，是期望收到对方的下一个报文段的数据的第一个字节的序号。



数据偏移（即首部长度的）——占 4 位，它指出 TCP 报文段的数据起始处距离 TCP 报文段的起始处有多远。“数据偏移”的单位是 32 位字（以 4 字节为计算单位）。说明 TCP 首部最大值 60 字节。



保留字段——占 6 位，保留为今后使用，但目前应置为 0。



紧急 URG —— 当 $URG = 1$ 时，表明紧急指针字段有效。它告诉系统此报文段中有紧急数据，应尽快传送(相当于高优先级的数据)。



确认 ACK —— 只有当 $ACK = 1$ 时确认号字段才有效。当 $ACK = 0$ 时，确认号无效。



推送 PSH (PuSH) —— 接收 TCP 收到 PSH = 1 的报文段，就尽快地交付接收应用进程，而不再等到整个缓存都填满了后再向上交付。



复位 RST (ReSeT) —— 当 $RST = 1$ 时，表明 TCP 连接中出现严重差错（如由于主机崩溃或其他原因），必须释放连接，然后再重新建立运输连接。



同步 SYN —— 同步 SYN = 1 表示这是一个连接请求或连接接受报文。



终止 FIN (FINis) —— 用来释放一个连接。FIN = 1 表明此报文段的发送端的数据已发送完毕，并要求释放运输连接。



窗口字段——占 2 字节，用来让对方设置发送窗口的依据(发送本报文段一方的接收窗口)，单位为字节。



检验和 —— 占 2 字节。检验和检验的范围包括首部和数据这两部分。在计算检验和时，要在 TCP 报文段的前面加上 12 字节的伪首部。

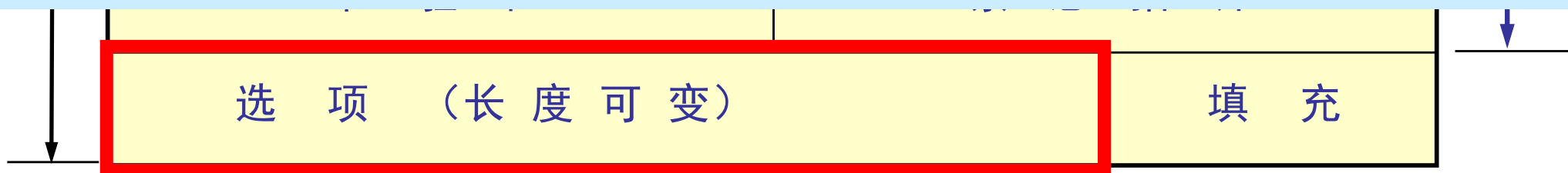


紧急指针字段 —— 占 16 位，指出在本报文段中紧急数据共有多少个字节（紧急数据放在本报文段数据的最前面）。

MSS (Maximum Segment Size)

是 TCP 报文段中的数据字段的最大长度。

数据字段加上 TCP 首部才等于整个的 TCP 报文段。



选项字段 —— 长度可变，最大40字节。TCP 最初只规定了一种选项，即最大报文段长度 MSS。MSS 告诉对方TCP：“我的缓存所能接收的报文段的数据字段的最大长度是 MSS 个字节。”

其他选项

□ 窗口扩大选项

- 占 3 字节，其中有一个字节表示移位值 S 。新的窗口值等于TCP首部中的窗口位数增大到 $(16 + S)$ ，相当于把窗口值向左移动 S 位后获得实际的窗口大小。

□ 时间戳选项

- 占10 字节，其中最主要的字段时间戳值字段（4 字节）和时间戳回送回答字段（4 字节）。

□ 选择确认选项

- 处理不连续的字节块（5.6.3 节）



填充字段 —— 这是为了使整个首部长度是 4 字节的整数倍。

作业

□ 12, 13, 18

Thanks, Any Question?

The End.

chenwh@nwnu.edu.cn

