



工业和信息化部“十二五”规划教材

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

# 计算机网络教程

## (第4版)

A Textbook on Computer Networks (4th Edition)

谢钧 谢希仁 编著

- 参考计算机专业考研大纲
- 体现了作者多年的教学经验
- 吸收了多种国外著名教材的优点
- 提供实验建议、教学PPT、部分习题答案

# 第五章 运输层

## 主要内容

### 运输层的基本原理

- ❑ 运输层的地位
- ❑ 运输层与网络层的区别

### 运输层机制

- ❑ 用户数据报协议（UDP）
- ❑ 传输控制协议（TCP）
  - | 端口和连接管理

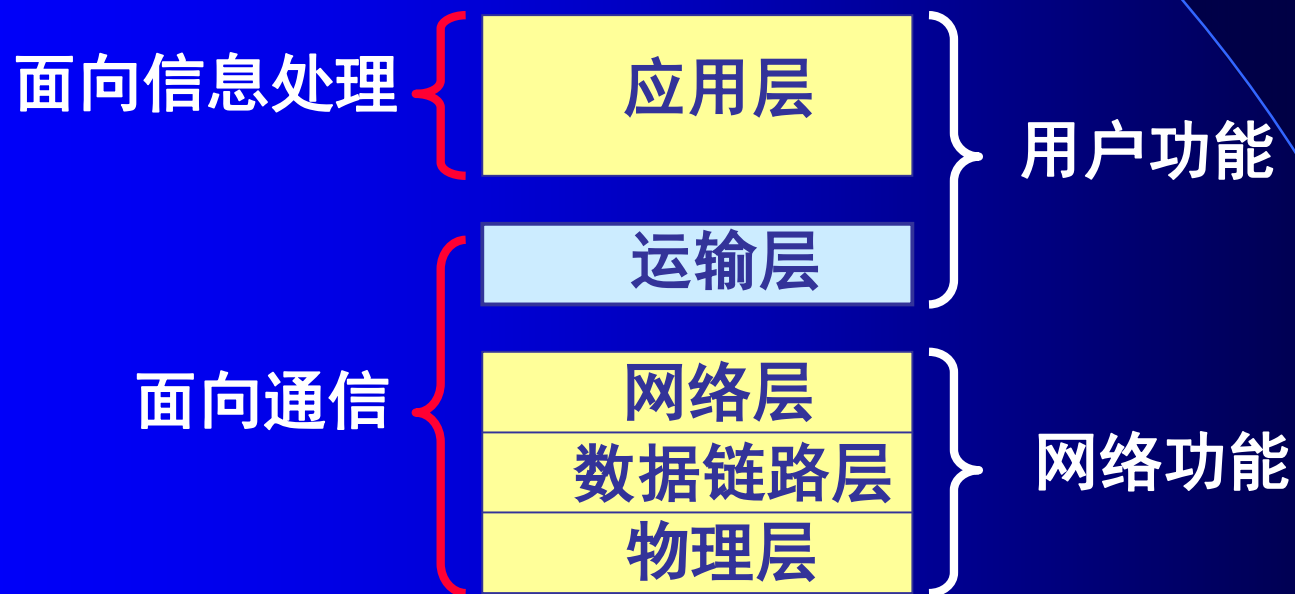
# 本章知识点分布

章	节	知识点	相关概念或原理
5	§ 5.1 运输层协议概述	进程间通信、运输层协议、运输层复用、运输层分用、端口	运输层地位、运输层主要功能、运输层协议与网络层协议区别、端口概念、端口分类、SOCKET
	§ 5.2 用户数据报协议UDP	UDP包格式、UDP校验和、UDP伪首部	UDP优点、伪首部作用
	§ 5.3 传输控制协议TCP	TCP报文结构、TCP可靠传输、TCP流量管理、TCP连接管理	TCP特点、TCP校验和、TCP伪首部、滑动窗口、三次握手
	§ 5.4 拥塞控制	拥塞控制	慢开始、拥塞避免、快速恢复

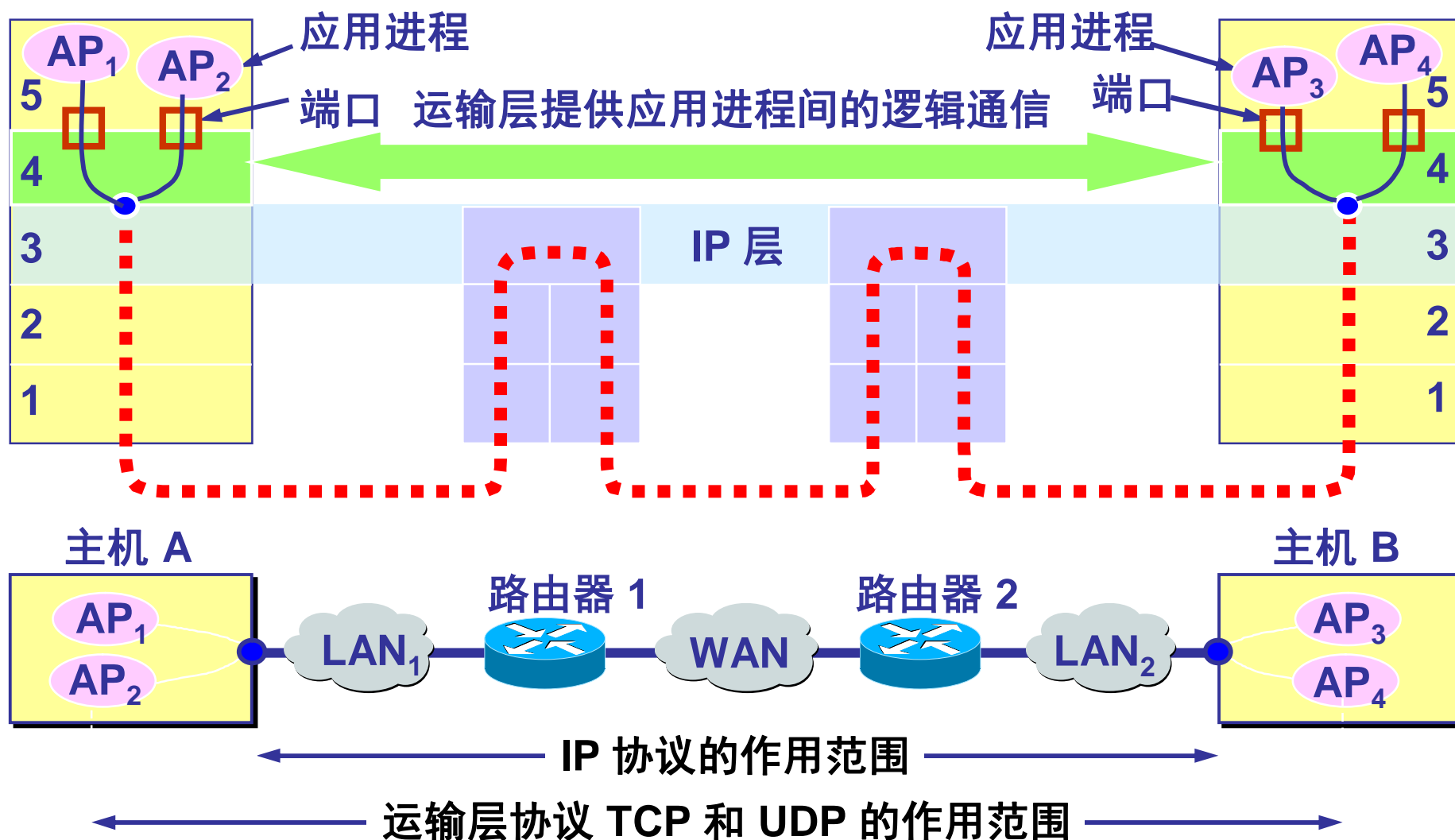
# 5.1 运输层协议概述

## 5.1.1 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。

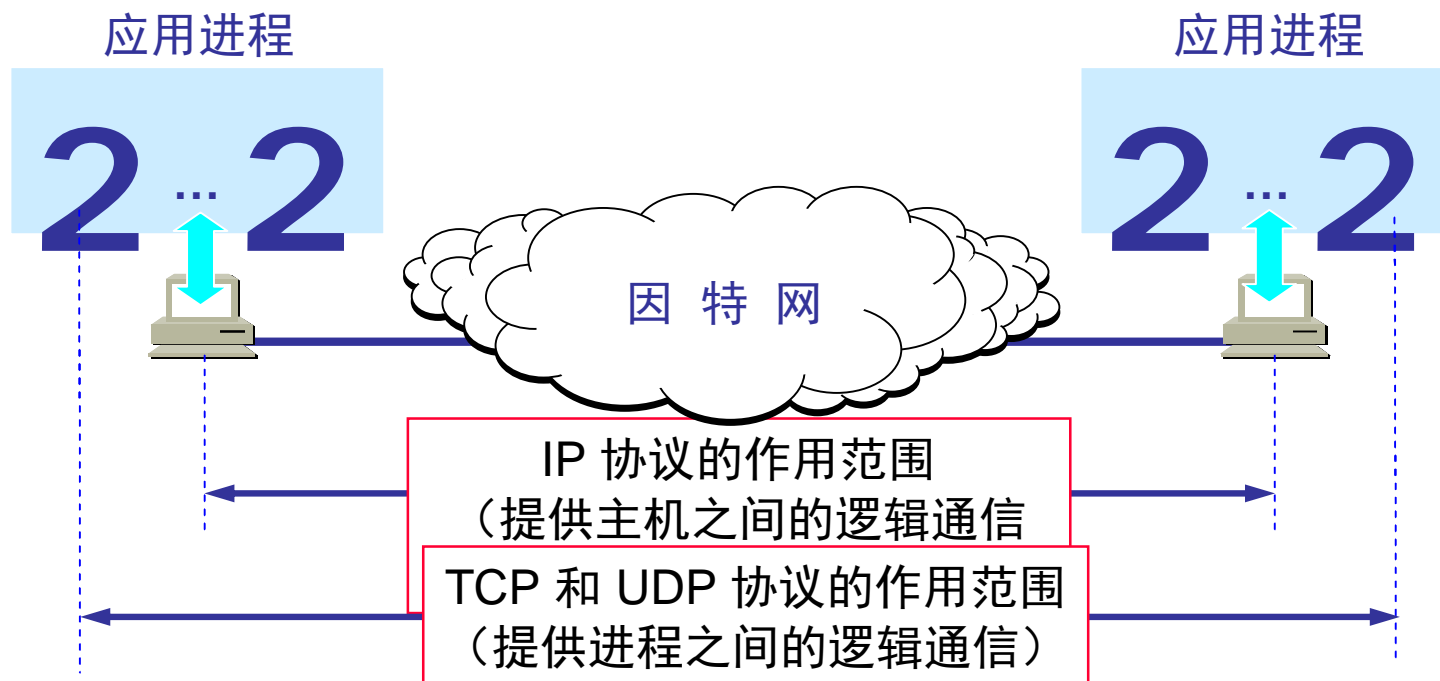


# 运输层为相互通信的应用进程提供了逻辑通信



# 运输层协议与网络层协议的区别

- (1) 运输层为应用进程之间提供端到端的逻辑通信，而网络层是为主机之间提供逻辑通信；(P200图5-2)
- (2) 运输层还为收到的报文进行差错检测，而网络层只检验IP数据报首部是否出错却不检查数据部分；
- (3) 根据应用不同，运输层需要有两种不同的协议：面向连接的TCP和无连接的UDP，而网络层无法同时实现这两种协议，仅是一种无连接的服务。



# 运输层的主要功能

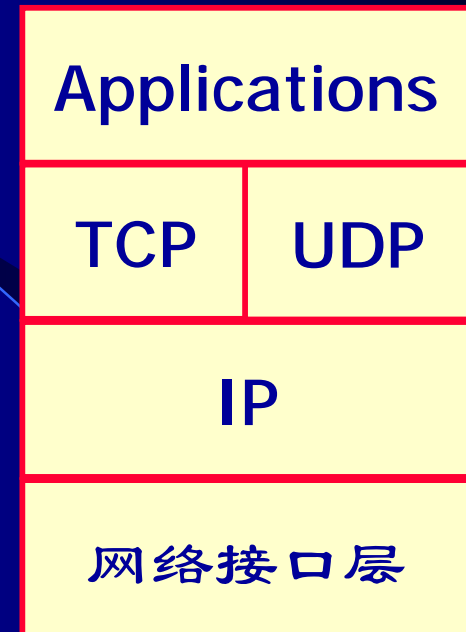
- Ø 运输层为应用进程之间提供端到端的逻辑通信（但网络层是为主机之间提供逻辑通信）。
- Ø 运输层还要对收到的报文进行差错检测。
- Ø 运输层需要有两种不同的运输协议，即面向连接的 TCP 和无连接的 UDP。

## 5.1.2 因特网的运输层协议

**UDP – User Datagram Protocol**

**TCP – Transport Control Protocol**

各种应用详见课本P201表5-1





## 5.1.3 运输层的复用与分用

### 1、端口概念的引出：

Ø当前大多数操作系统都是支持多进程的，即每一台主机上可以同时运行多个进程或应用程序。每一台主机中的每一个进程都可以与另一台主机中的另一个进程通信。

Ø但在IP协议层，每一个目标IP地址标识一台主机，没有更多的信息可以用来区分数据报应该交给那一个进程或应用程序。如何解决这个问题？

## 端口概念的引出：

Ø 解决的方案是：按功能（服务）分类，为每类功能提供一个协议“端口”作为最终的报文接收端。

Ø 协议端口是逻辑意义上的，用一些正整数标识

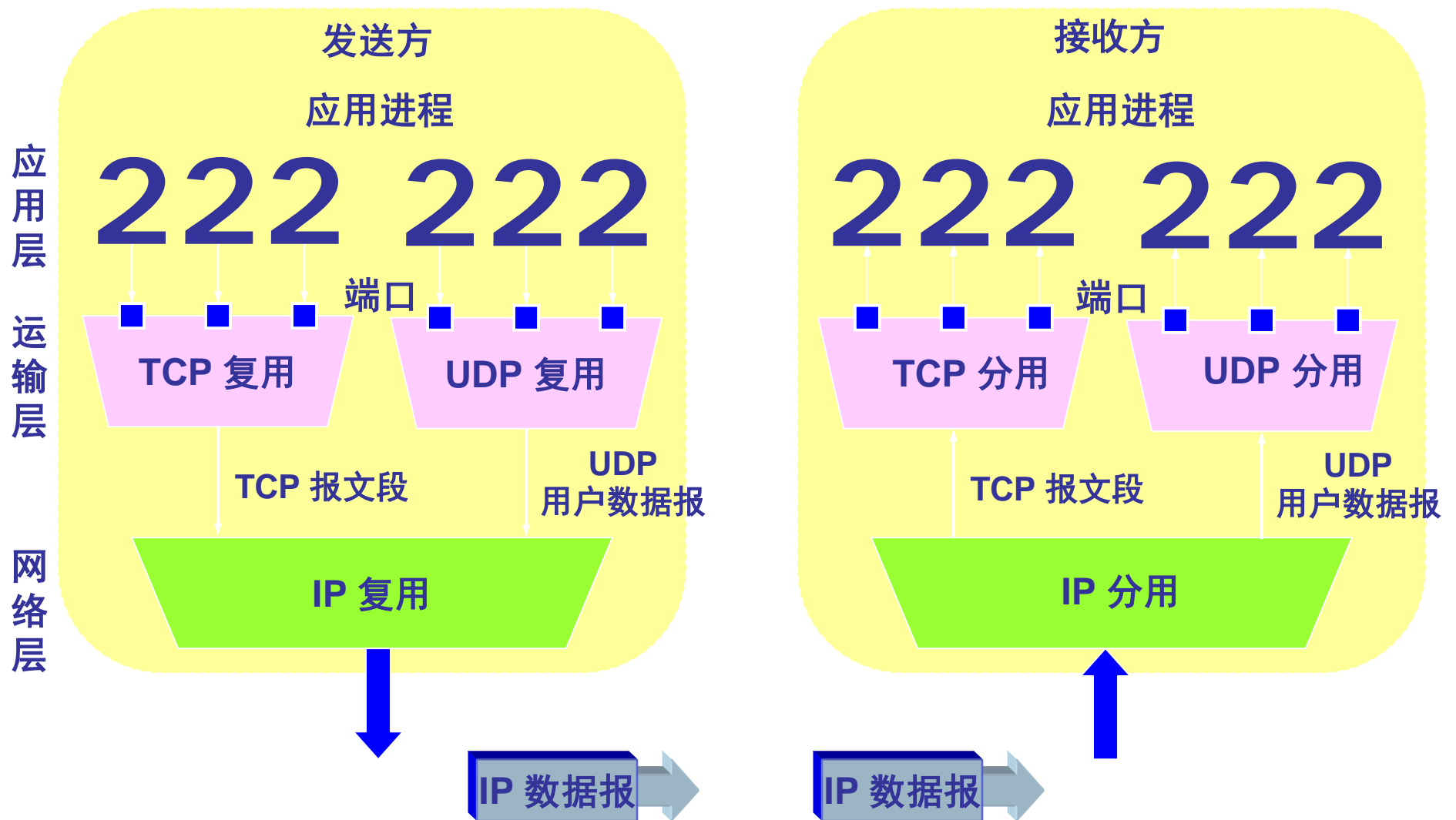
Ø 一台主机可以设置多个端口，也即意味着同一主机可运行多种服务。

Ø 基于操作系统的支持，多个端口可并行访问，即多种服务可同时运行。

## 2、端口的概念

- Ø 端口就是运输层服务访问点 TSAP。
- Ø 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。
- Ø 从这个意义上讲，端口是用来标志应用层的进程。

# 端口在进程之间的通信中所起的作用



### 3、三类端口

- (1) **熟知端口**，其数值一般为 0~1023。当一种新的应用程序出现时，必须为它指派一个熟知端口。
- (2) **登记端口**，其数值为 1024~49151。这类端口是 ICANN 控制的，使用这个范围的端口必须在 ICANN 登记，以防止重复。
- (3) **动态端口**，其数值为 49152~65535。这类端口是留给客户进程选择作为临时端口。

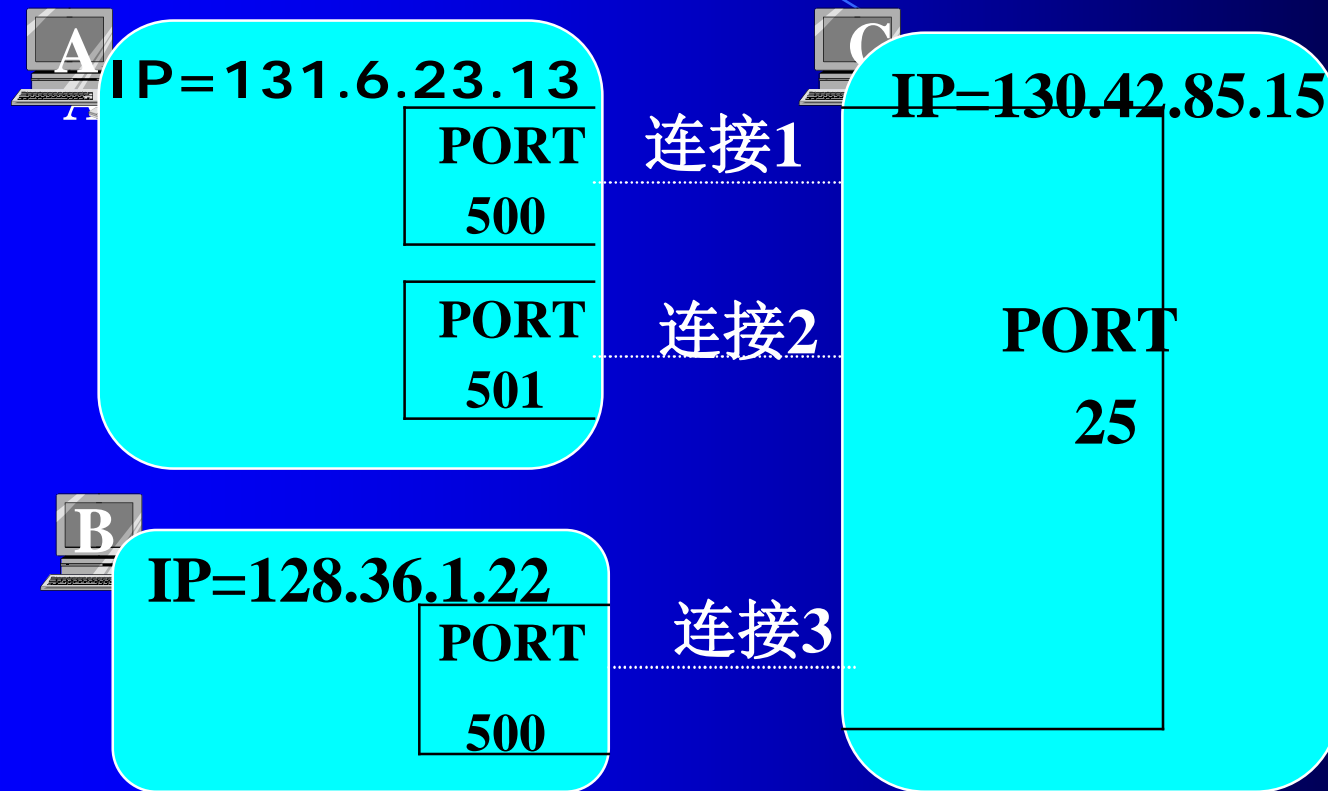
# 熟知端口：最常用的应用服务的端口

Application	RPC	SNMP	TFTP	SMTP	FTP	TELNET
	( )	( )	( )	( )	( )	( )
	111	161	69	25	21	23
Transport	UDP			TCP		
Network	IP					

Well-known port

## 4、SOCKET概念

- SOCKET（套接字）：标识连接的端点，IP地址 + 端口号



- | 连接1的一对socket是(131.6.23.13, 500)和(130.42.85.15, 25)
- | 连接2的一对socket是(131.6.23.13, 501)和(130.42.85.15, 25)
- | 连接3的一对socket是(128.36.1.22, 500)和(130.42.85.15, 25)

## 5.2 用户数据报协议UDP

### User Datagram Protocol

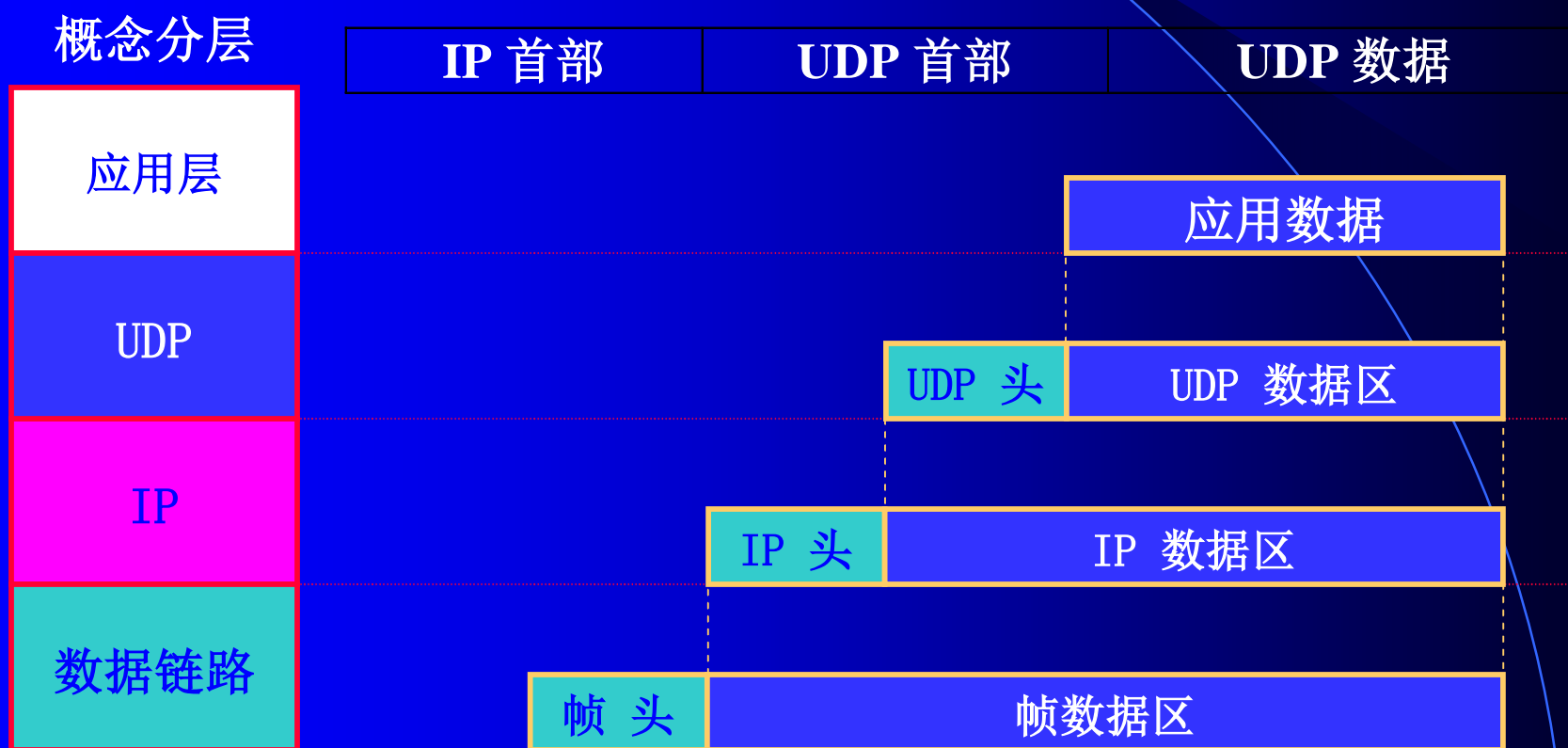
#### UDP的优点：详见P203

- 1、UDP 是无连接的，即发送数据之前不需要建立连接。
- 2、UDP 使用尽最大努力交付，不保证可靠交付，也不使用拥塞控制。
- 3、UDP 没有拥塞控制，因此网络出现的拥塞不会使源主机的发送速率降低。
- 4、UDP是面向报文的。
- 5、UDP 支持一对一、一对多、多对一和多对多的交互通信。
- 6、首部开销小。
- 7、对某些实时应用较好。



# UDP 协议封装

- UDP 数据报由两部分构成：UDP 报头和数据区
- UDP 报文是封装在 IP 分组中进行传送的



# UDP 数据报的格式



## UDP 源端口：

发送端的 UDP 端口号；当不需要对端返回数据时，该字段为 0。

## UDP 目的端口：

接收端的 UDP 端口号。

## 数据报长度：

以字节计算的整个数据报的长度，最小值 8 字节（只含 UDP 头）。

## 数据报校验和（可选）：

0：表示未选用校验功能；  
其它值表示数据报的校验和，若该字段为全 1 则表示校验和为 0。

# UDP 数据报的校验

- Ø UDP 数据报校验和是一项可选的功能
- Ø 用户禁止该功能可以进一步提高通信的效率
- Ø UDP 校验和的计算方法：与 IP 分组头的校验方法相似
- Ø 校验和计算：除校验数据报首部外，还校验 UDP 伪首部和数据部分。

Ø 其中伪首部的格式为：

q 伪首部并非 UDP 数据报中实际的有效成分

q 伪首部是一个虚拟的数据结构：

其中的信息是从数据报所在 IP 分组头的分组头中提取的

源 IP 地址	目的 IP 地址	UDP 数据报的长度 (不含伪首部)
填充域	协议	UDP 长度

使用伪首部是为了验证 UDP 数据报是否正确地传到了目的系统中

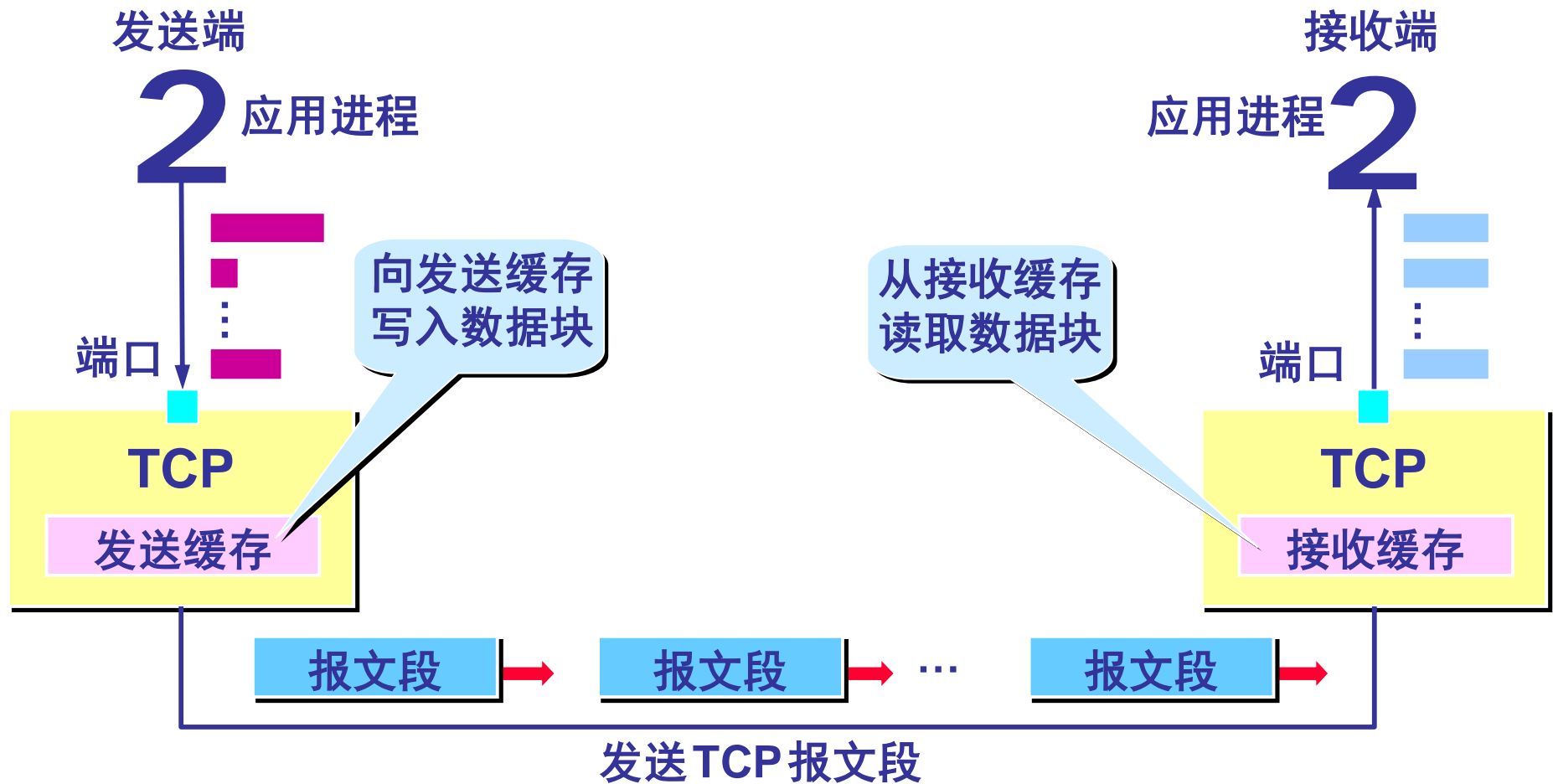
q 伪首部的采用在一定程度上违反了网络结构分层的原则

全 0

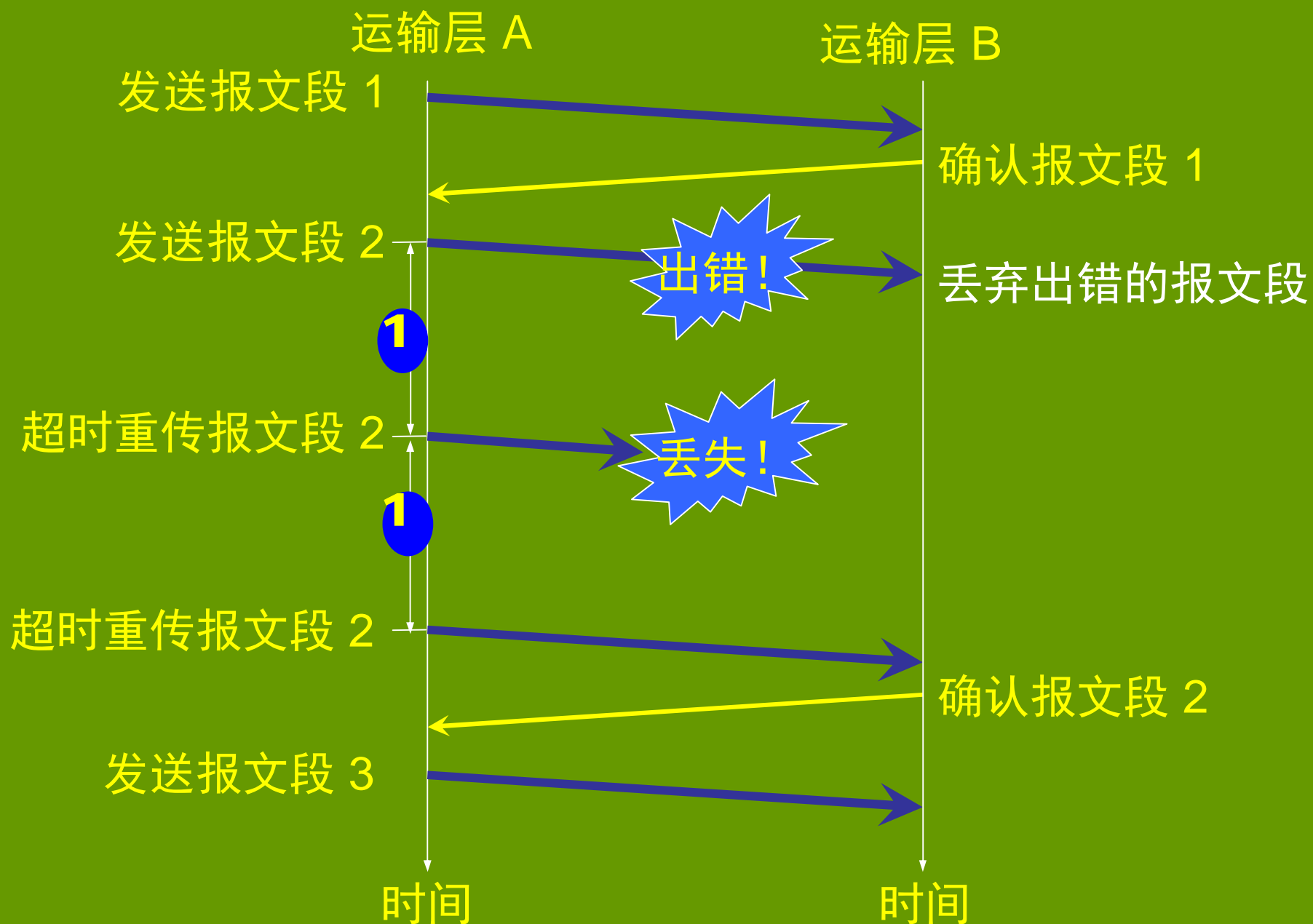
IP 分组头中指定的协议类型码 (UDP=17)

## 5.3 传输控制协议TCP

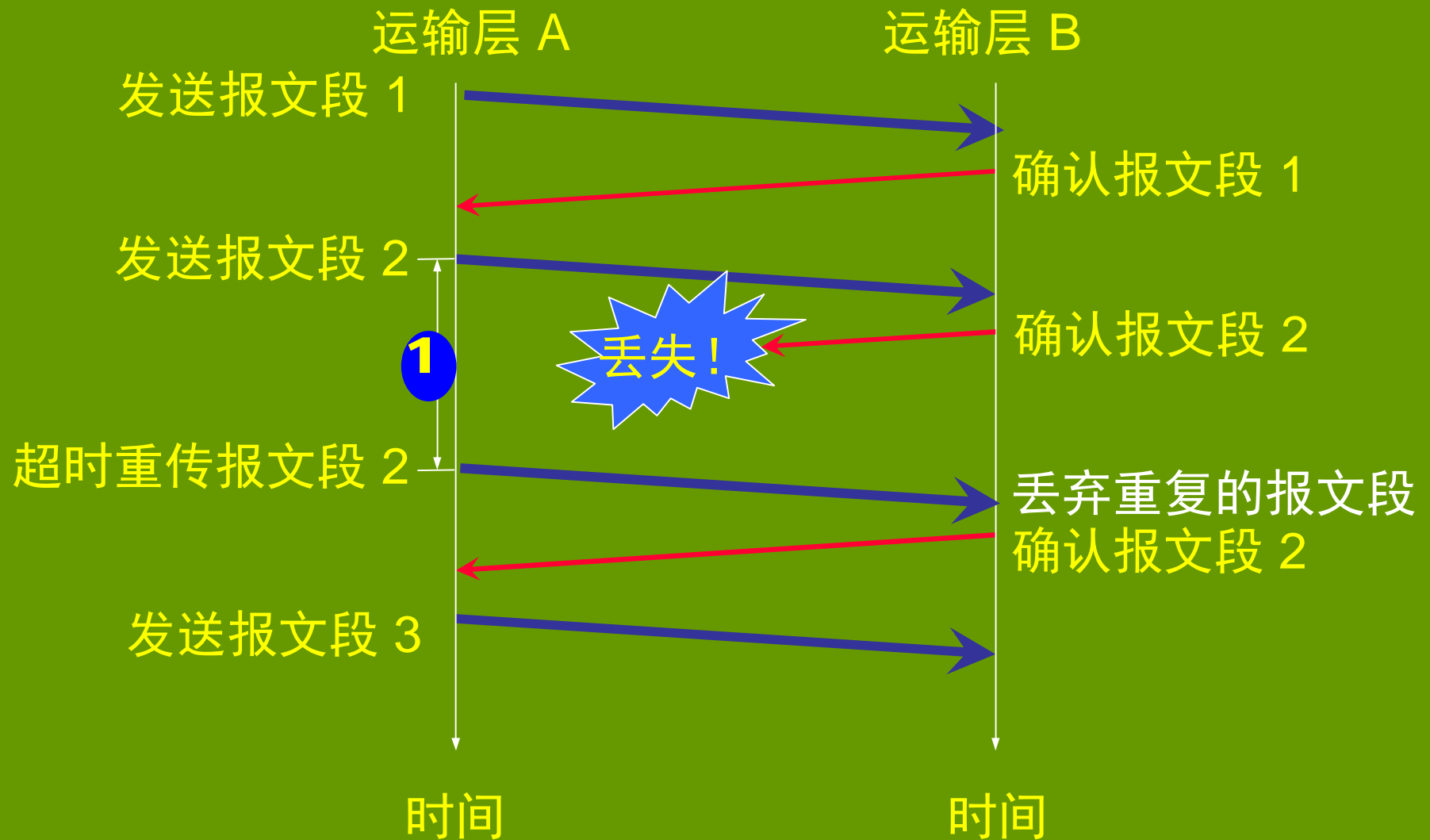
### 5.3.1 TCP 主要特点



# 用确认和超时重传保证可靠传输



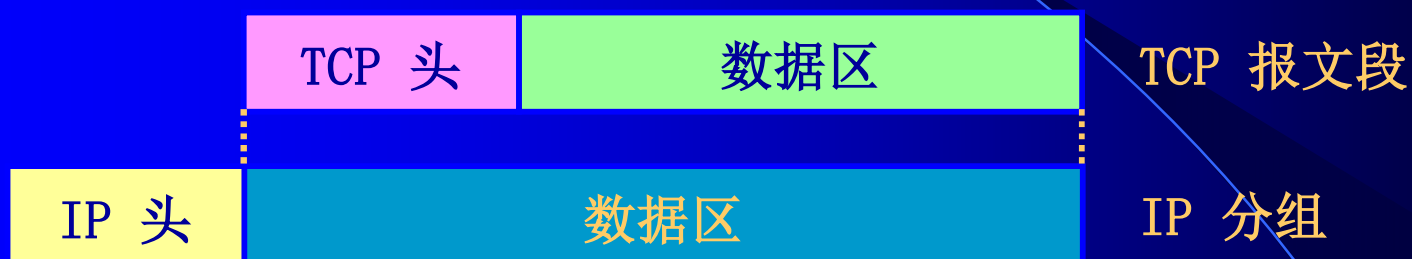
# 确认报文丢失情况



## 5.3.2 TCP 报文段结构

### Ø TCP 报文段的结构

Ø 报文段分为头部和数据区，并封装在一个 IP 分组中传输



Ø TCP 头：携带必须的标识和控制信息，包括：

Ø 连接标识：

源端口和目的端口：标识连接的两个端点

Ø 差错和流量控制：

序号：指出本报文段在发送方的数据字节流中的位置

确认序号：指出本机希望接收的下一个字节的序号

## TCP 报文段格式

源端口/目的端口：TSAP地址，标识接收端和发送端的应用进程。这两个值加上IP首部中的源IP地址和目的IP地址唯一

确认序号：期望收到的下一个消息第一字节的序号。为确认的一端所期望接收的下一个序号。只有在标识位中的ACK 比特设置为1时，此序号才有效。

窗口大小：滑动窗口协议中的窗口大小

数据偏移：以32比特为计算单位，通知发送方接收窗口的大小，即最多可以发送的字节数

数据偏移	保留	控制字段	窗口
校验和		紧急指针	
任选项（若有）			填充

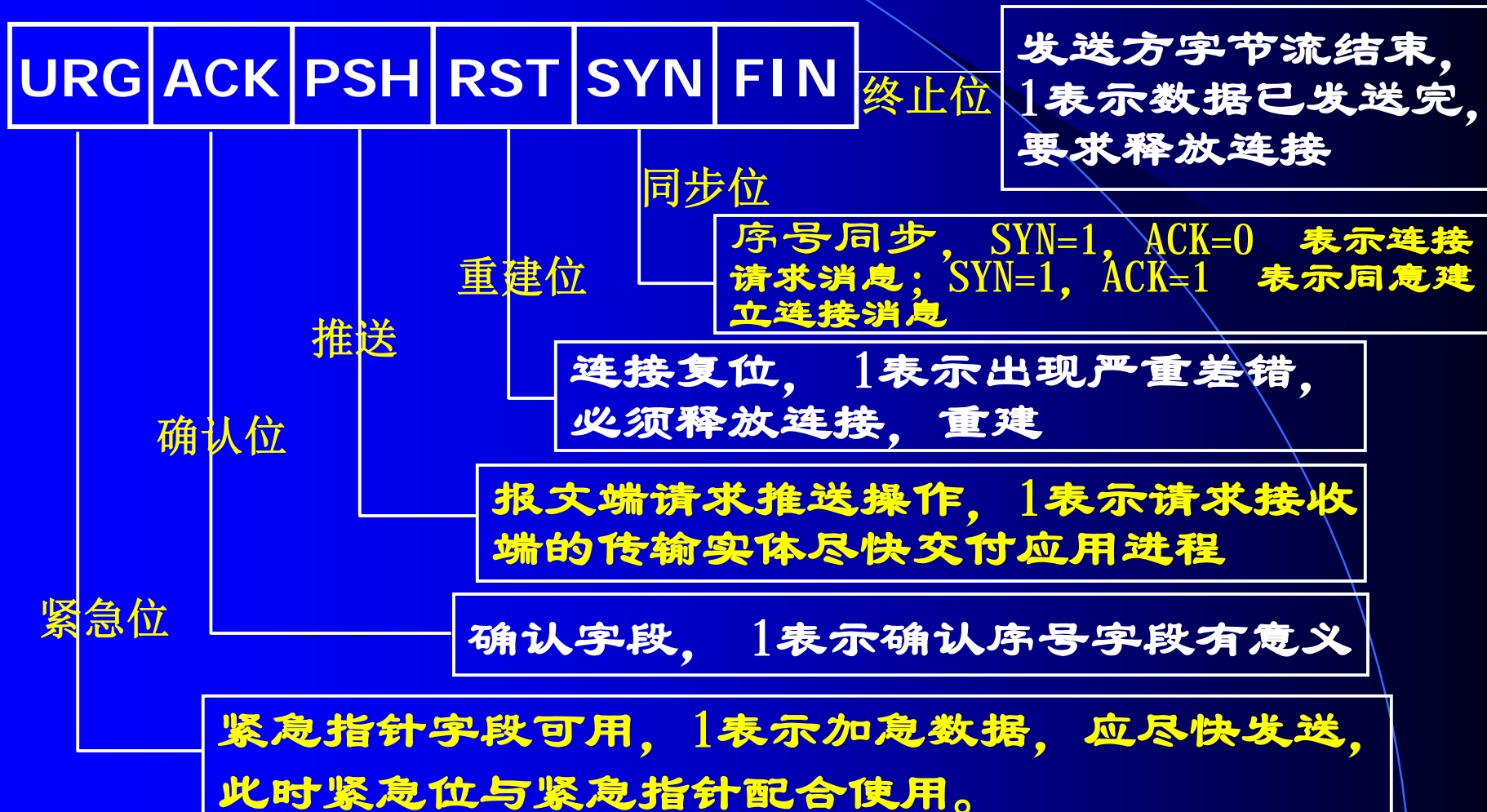
检校和：对整个TCP首部和TCP数据校验和。这个字段在TCP中是强制性的，一定存在，并在收端进行验证。12B的伪首部+数据。紧急指针指出本报文段中的紧急数据的最后一个字节的序号，它使接收方知道紧急数据共有多长。



# TCP 报文段中的控制字段(选学)

## Ø 控制字段 (Control BITS) 6bit

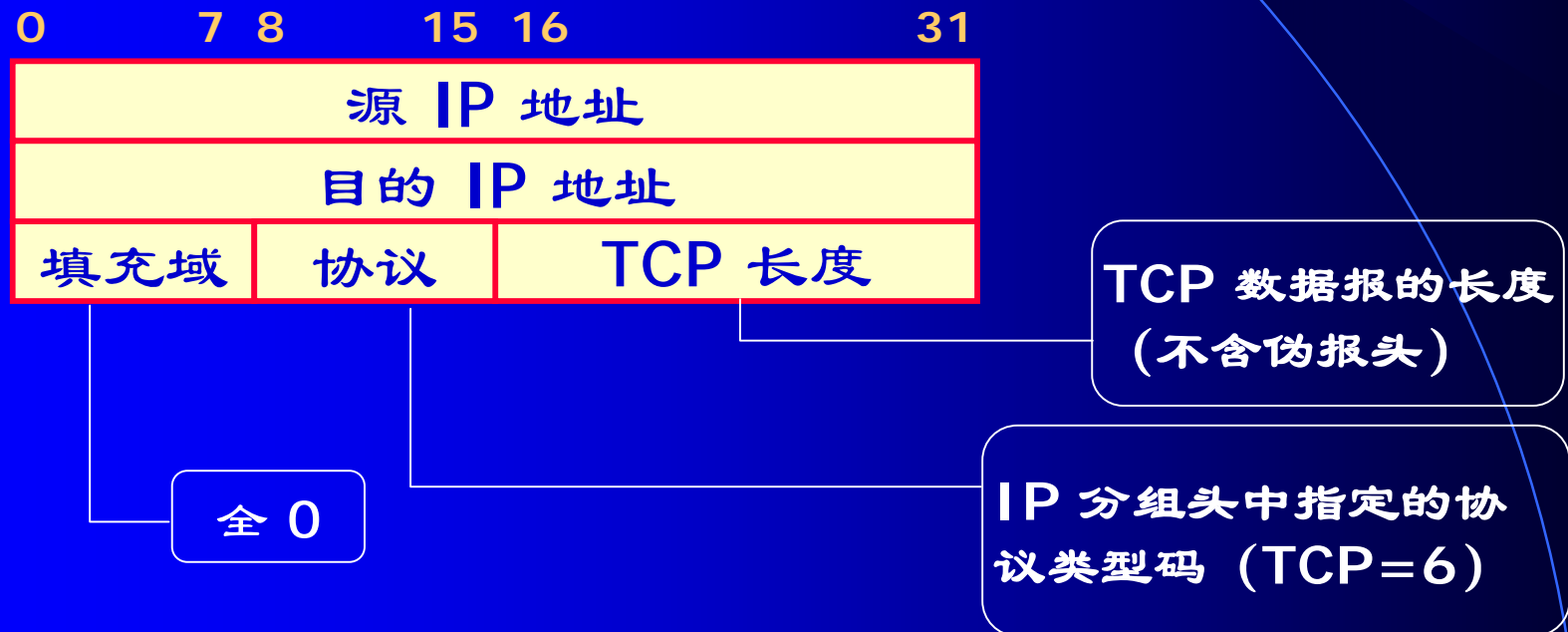
- 指出报文段的目的和内容, 给出报文头中其他字段的解释



# TCP 报文段的校验

Ø TCP 校验和的计算方法：同 UDP 的校验

- 校验和计算：除覆盖数据报外，还覆盖一个 12字节的 TCP 伪首部
- TCP 伪首部的目的与 UDP 基本相同



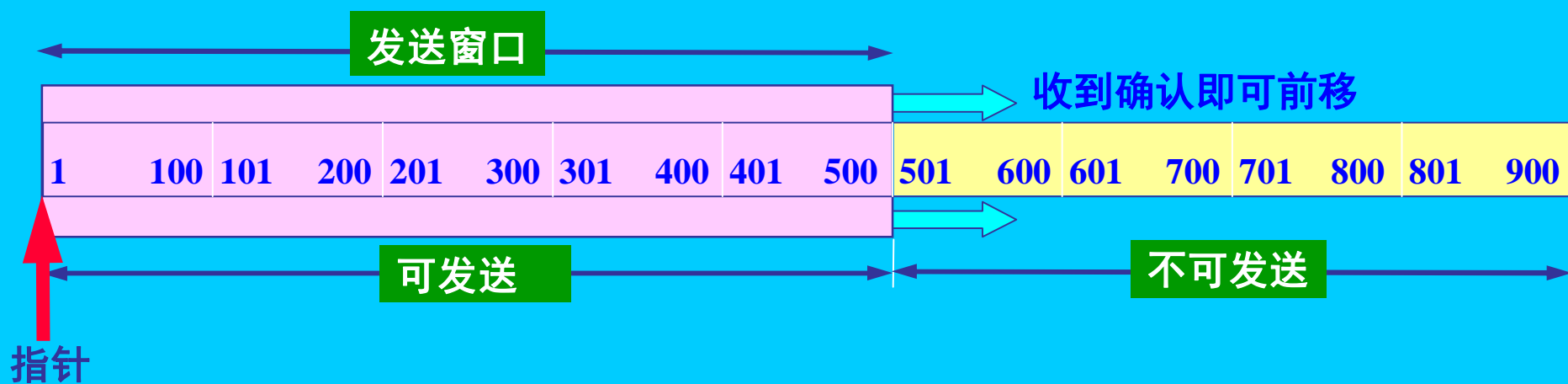
## 5.3.3 TCP 的可靠传输(选学)

### 一、 数据编号与确认

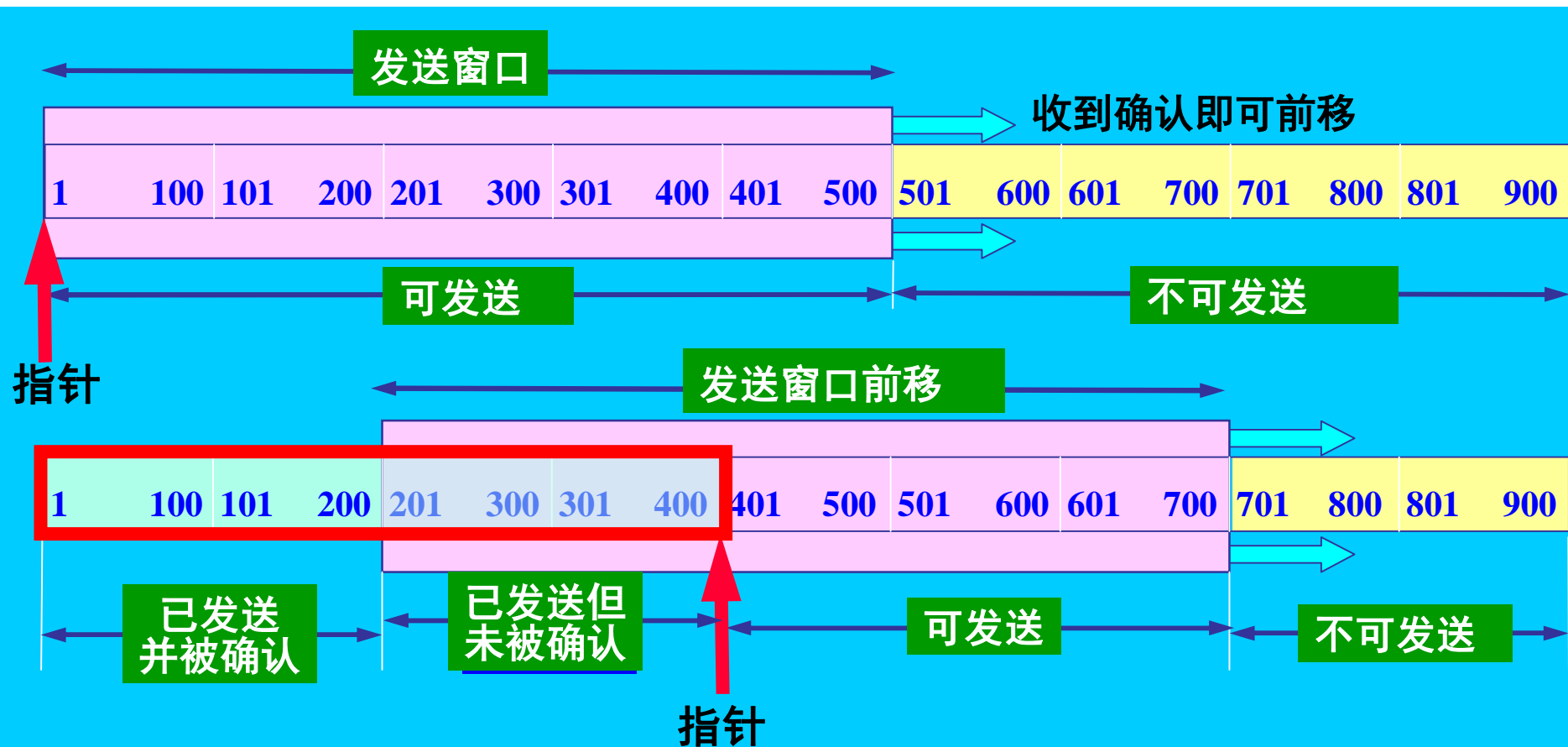
- Ø TCP 协议是面向字节的。TCP 将所要传送的报文看成是字节组成的数据流，并使每一个字节对应于一个序号。
- Ø 在连接建立时，双方要商定初始序号。TCP 每次发送的报文段的首部中的序号字段数值表示该报文段中的**数据部分的第一个字节的序号**。
- Ø TCP 的确认是对接收到的数据的最高序号表示确认。接收方返回的确认号是已收到的数据的最高序号加 1。因此确认号表示**接收方期望下次收到的数据中的第一个数据字节的序号**。

## 2. 滑动窗口的概念

- Ø 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。
- Ø 在 TCP 报文段首部的窗口字段写入的数值就是当前给对方设置的发送窗口数值的上限。
- Ø 发送窗口在连接建立时由双方商定。但在通信的过程中，接收方可根据自己的资源情况，随时动态地调整对方的发送窗口上限值(可增大或减小)。

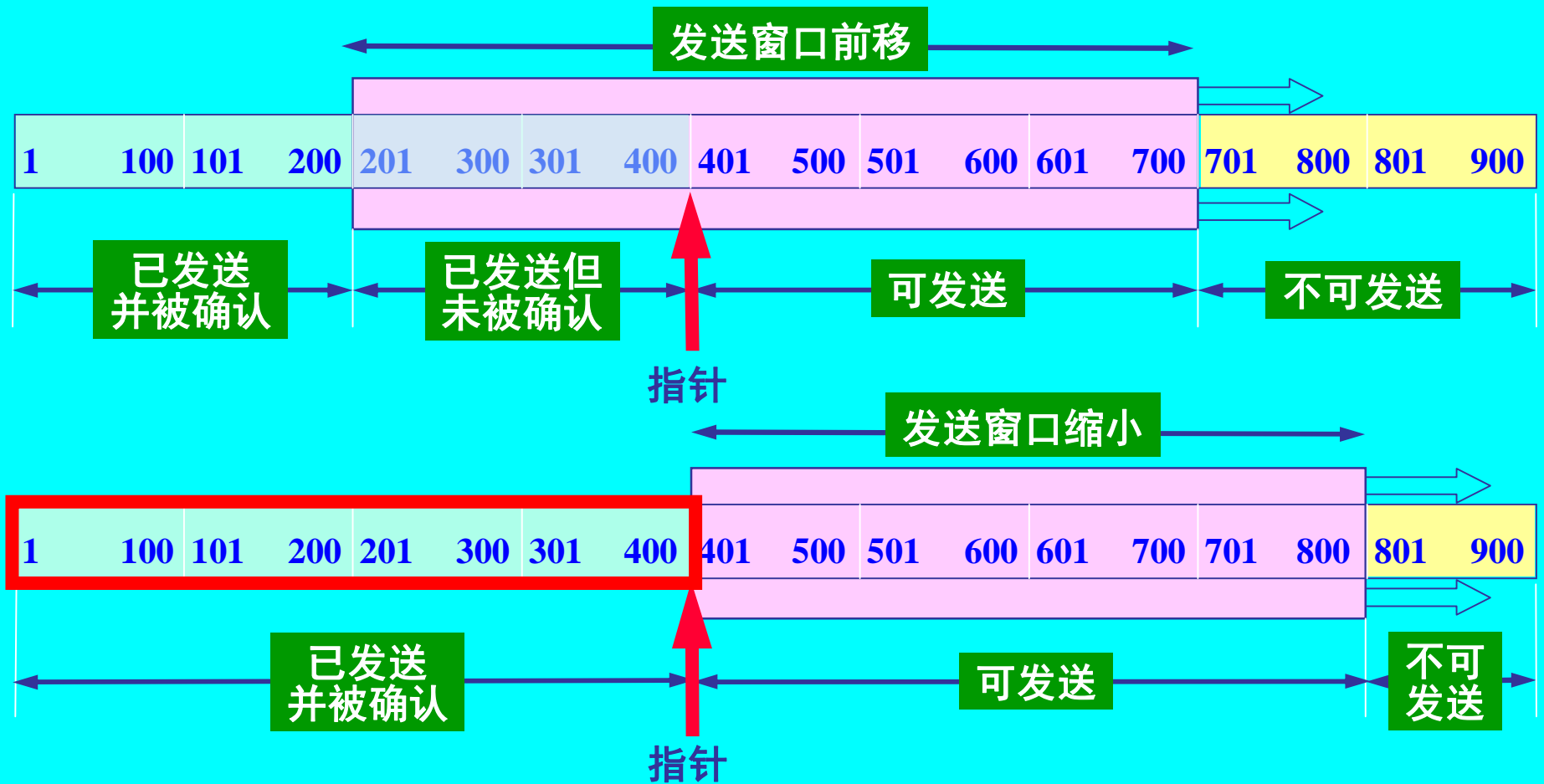


- Ø 发送方要发送 900 字节长的数据，划分为 9 个 100 字节长的报文段，而发送窗口确定为 500 字节。
- Ø 发送方只要收到了对方的确认，发送窗口就可前移。
- Ø 发送 TCP 要维护一个指针。每发送一个报文段，指针就向前移动一个报文段的距离。



Ø 发送方已发送了 400 字节的数据，但只收到对前 200 字节数据的确认，同时窗口大小不变。

Ø 现在发送方还可发送 300 字节。



Ø 发送方收到了对方对前 400 字节数据的确认，但对方通知发送方必须把窗口减小到 400 字节。

Ø 现在发送方最多还可发送 400 字节的数据。

### 3. 慢开始和拥塞避免(选学)

- Ø 发送方的主机在确定发送报文段的速率时，既要根据接收方的接收能力，又要从全局考虑不要使网络发生拥塞。
- Ø 因此，每一个 TCP 连接需要有以下两个状态变量：
  - Ø 接收方窗口 `rwnd` (receiver window) 又称为通知窗口 (advertised window)。
  - Ø 拥塞窗口 `cwnd` (congestion window)。
- Ø (1) 接收方窗口：是接收方根据其目前的接收缓存大小所许诺的最新的窗口值，是来自接收方的流量控制。接收方将此窗口值放在 TCP 报文的首部中的窗口字段，传送给发送方。
- Ø (2) 拥塞窗口：是发送方根据自己估计的网络拥塞程度而设置的窗口值，是来自发送方的流量控制。



## 三个窗口的关系

发送端的**发送窗口上限**取“通知窗口”和“拥塞窗口”中的较小的一个：

$$\text{发送窗口} = \min(\text{通知窗口}, \text{拥塞窗口})$$

Ø当  $\text{rwnd} < \text{cwnd}$  时，是接收方的接收能力限制发送窗口的最大值。

Ø当  $\text{cwnd} < \text{rwnd}$  时，则是网络的拥塞限制发送窗口的最大值。

## 三个技术

为了更好地进行拥塞控制，常使用以下三种技术，即：

**慢开始(slow-start):** 在主机刚刚开始发送报文段时可先将拥塞窗口 `cwnd` 设置为一个最大报文段 `MSS` 的数值。在每收到一个对新的报文段的确认后，将拥塞窗口增加至多一个 `MSS` 的数值。

用这样的方法逐步增大发送方的拥塞窗口 `cwnd`，可以使分组注入到网络的速率更加合理。

**乘法减小(multiplicative decrease):** 是指每出现一次超时，就将拥塞窗口值减半。若超时频繁出现，则慢开始门限值减小的速率是很快的。

**加法增大(additive increase):** 是指执行拥塞避免算法后，当收到对所有发送出的报文的确认就将拥塞窗口值增加，使拥塞窗口缓慢增大，以防止网络过早出现拥塞。

# 拥塞避免

**拥塞避免(congestion avoidance):** 是指当拥塞窗口增大到门限窗口值时, 就将拥塞窗口指数增长速率降低为线性增长速率, 避免网络再次出现拥塞。

使用这些技术的一个前提就是: 由于通信线路带来的误码而使得分组丢失的概率很小(远小于1%): 因此, 只要出现分组丢失或延迟过长而引起超时重发, 就意味着在网络中的某个地方出现了拥塞。

## 5.3.4 TCP 的流量控制(选学)

Ø 如果发送方把数据发送得过快，接收方就可能来不及接收，这就会造成数据的丢失。

Ø 流量控制(flow control)就是让发送方的发送速率不要太快，既要让接收方来得及接收。

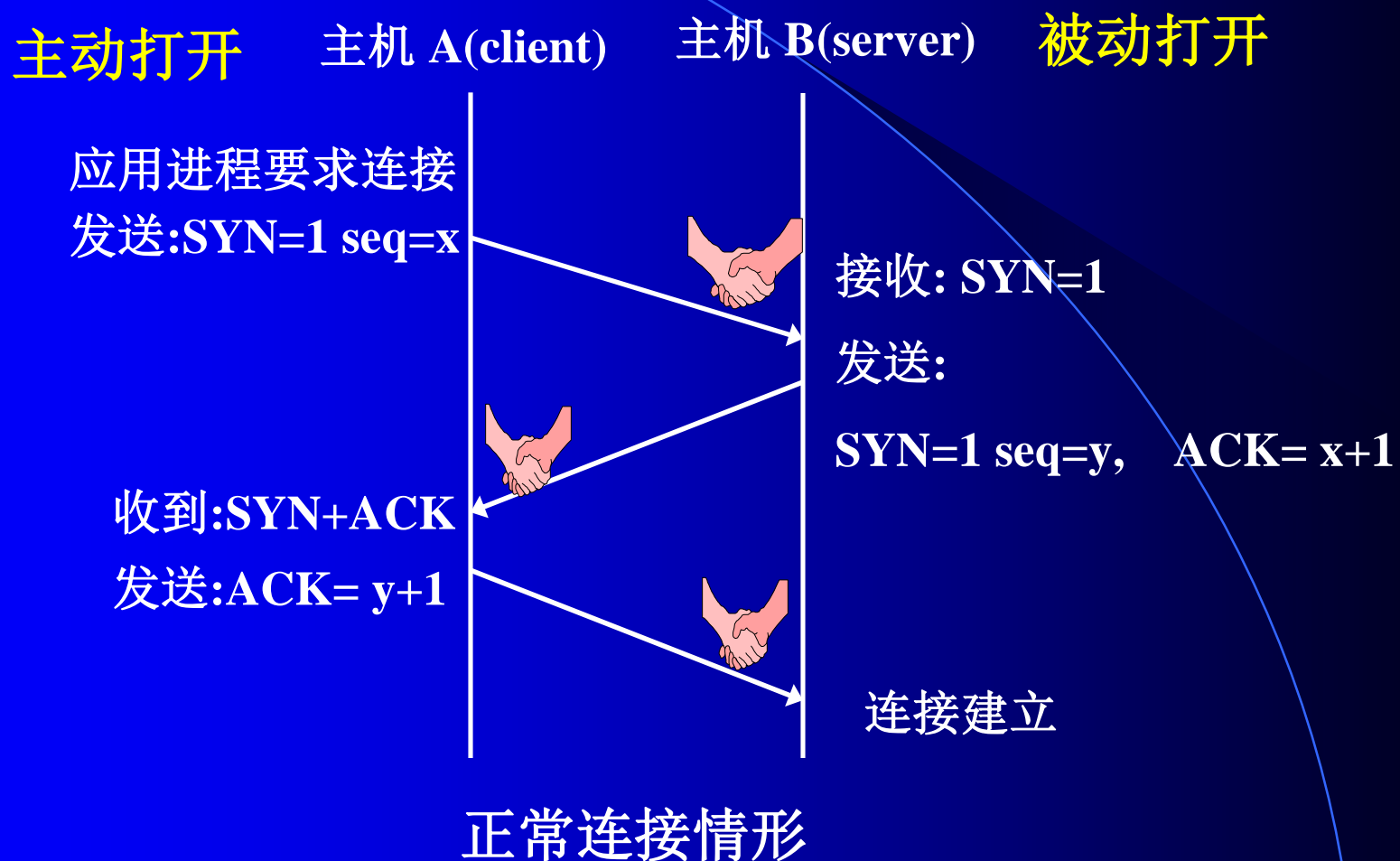
Ø 利用滑动窗口机制可以很方便地在 TCP 连接上实现流量控制。

## 5.3.5 TCP 的连接管理

在连接建立过程中要解决三个问题:

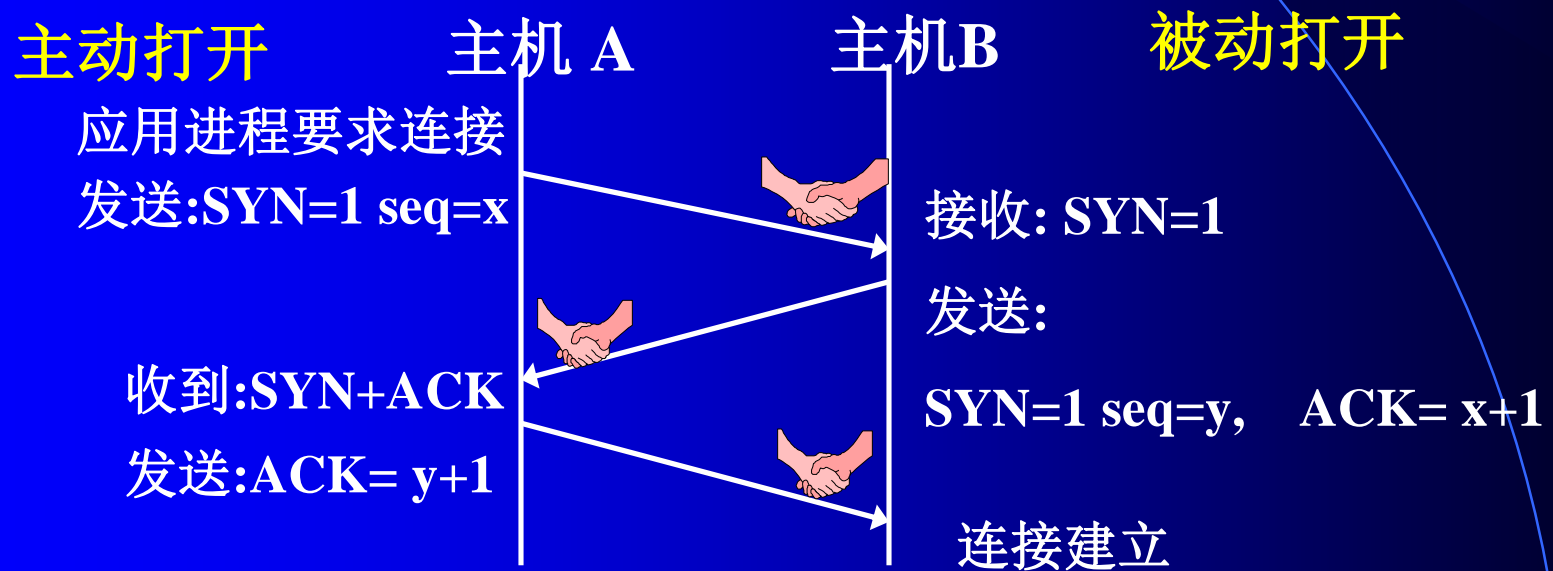
- 1、使每一方知道对方的存在;
- 2、允许双方协商一些参数(如最大窗口大小、服务质量);
- 3、能够对资源(如缓冲区大小、连接表中的项目)进行分配。

# 建立TCP连接：三次握手

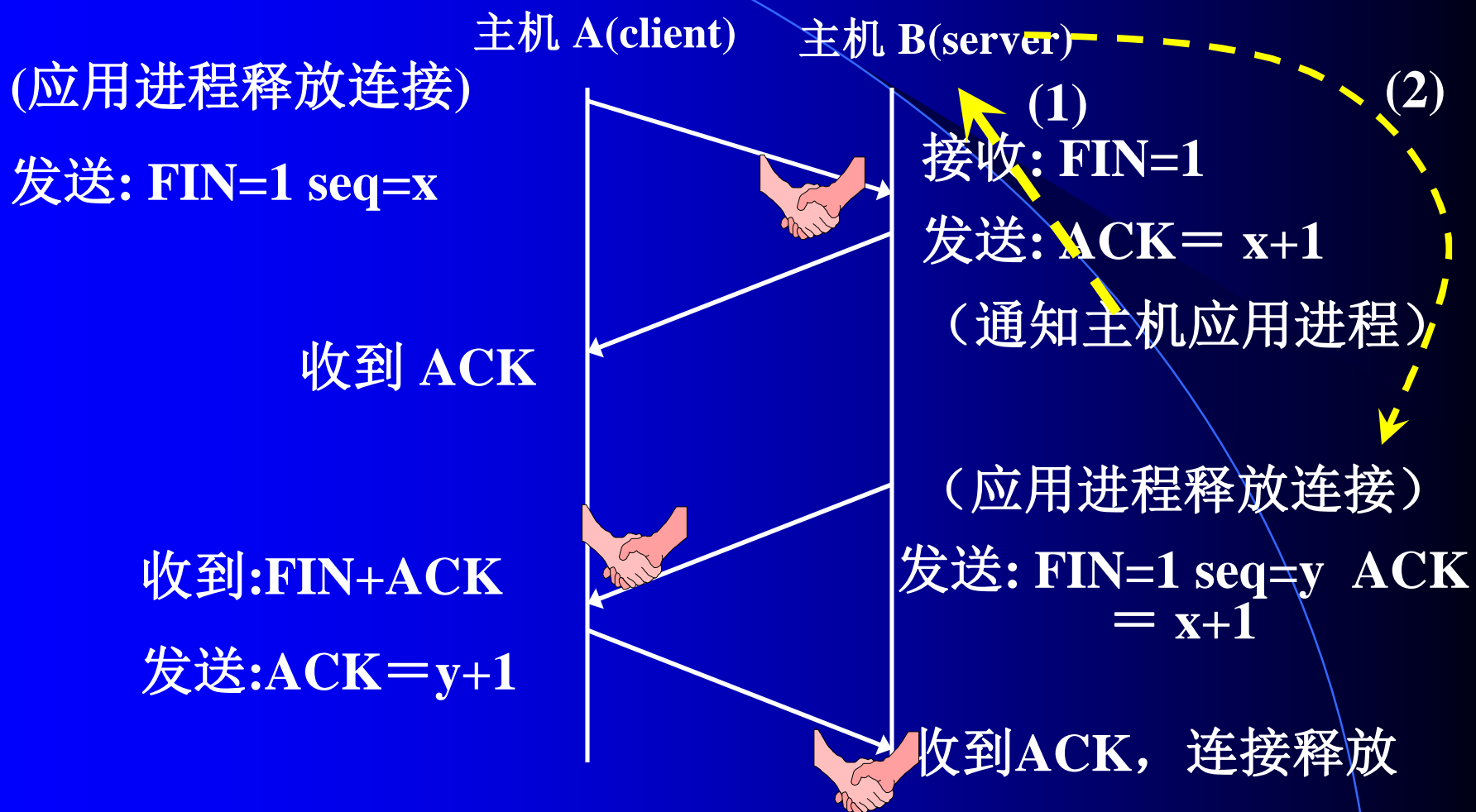


# 三次握手的功能:

- Ø 保证双方都相互知道对方已准备好进行数据传输
- Ø 双方确认一个数据传输的初始序列号, 如上图中, 发送方的初始序列号为 $x$ , 接收方的初始序列号为 $y$ , 均被对方确认



# 释放TCP连接：修改的三次握手



- 采用定时器方法，处理确认丢失的情况



## 动画演示

### TCP连接建立和释放过程

**PLAY**

**STOP**

# 本章作业

P229: 5-1    5-5    5-6