# 第六章 传输层(1)


课程名称：计算机网络
主讲教师：姚烨
课程代码：U10M11016.02
E-MAIL ：yaoye@nwpu.edu.cn


第39-40讲

2021 – 2022 学年第一学期

# 第 6 章 传输层

# 第 6 章 传输层（续）

# 第 6 章 传输层（续）

# 本节内容提要

6.1 传输层概述
- 6.1.1 进程之间的通信
- 6.1.2 传输层的两个主要协议
- 6.1.3 传输层的端口

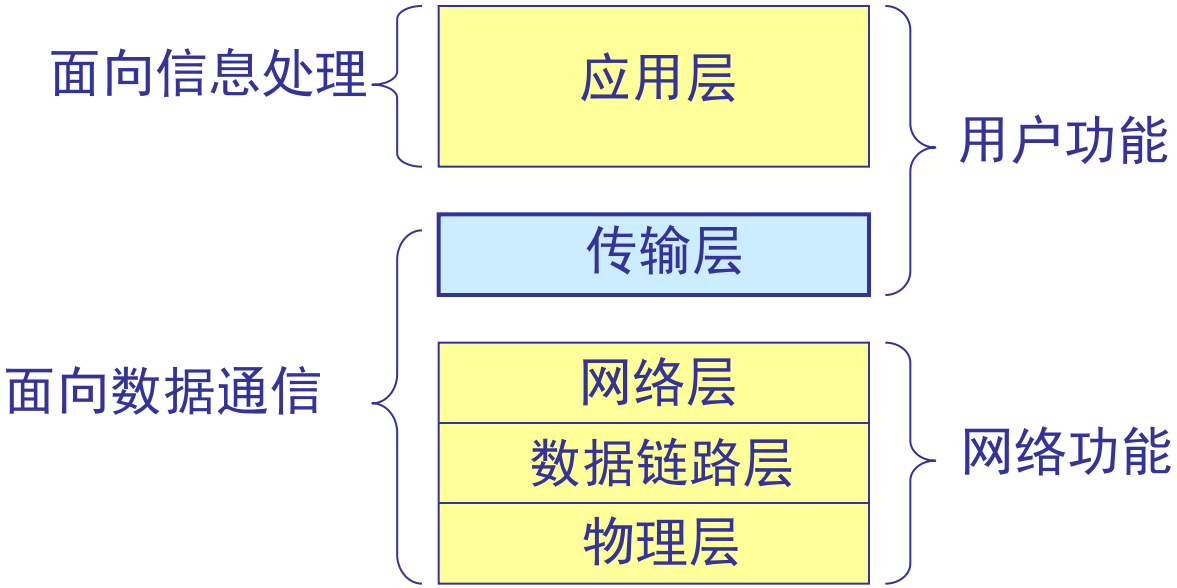
6.2 用户数据报协议 UDP
- 6.2.1 UDP 概述
- 6.2.2 UDP 的首部格式

6.3 传输控制协议 TCP 概述

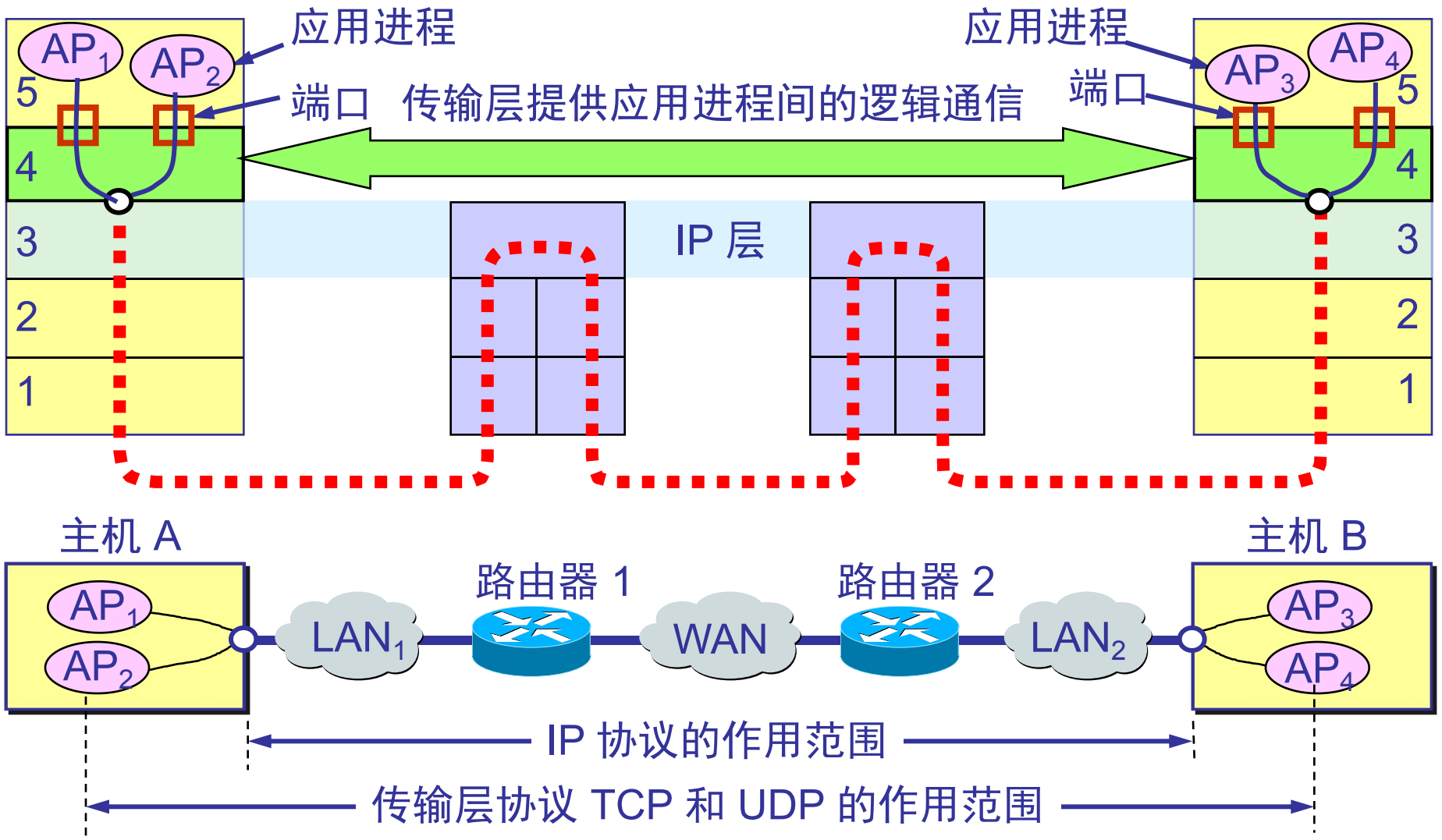
6.4 TCP 报文段的首部格式

# 6.1 传输层概述
# 6.1.1 进程之间的通信

- 从通信和信息处理的角度看，传输层向它上面的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。

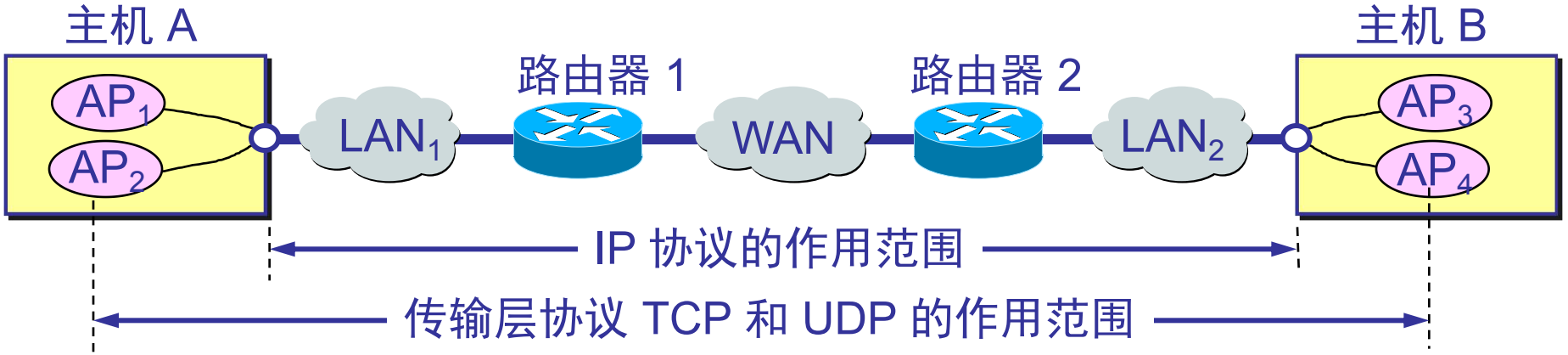


# 传输层主要功能：为端节点上不同应用进程间提供端到端的数据传输服务。

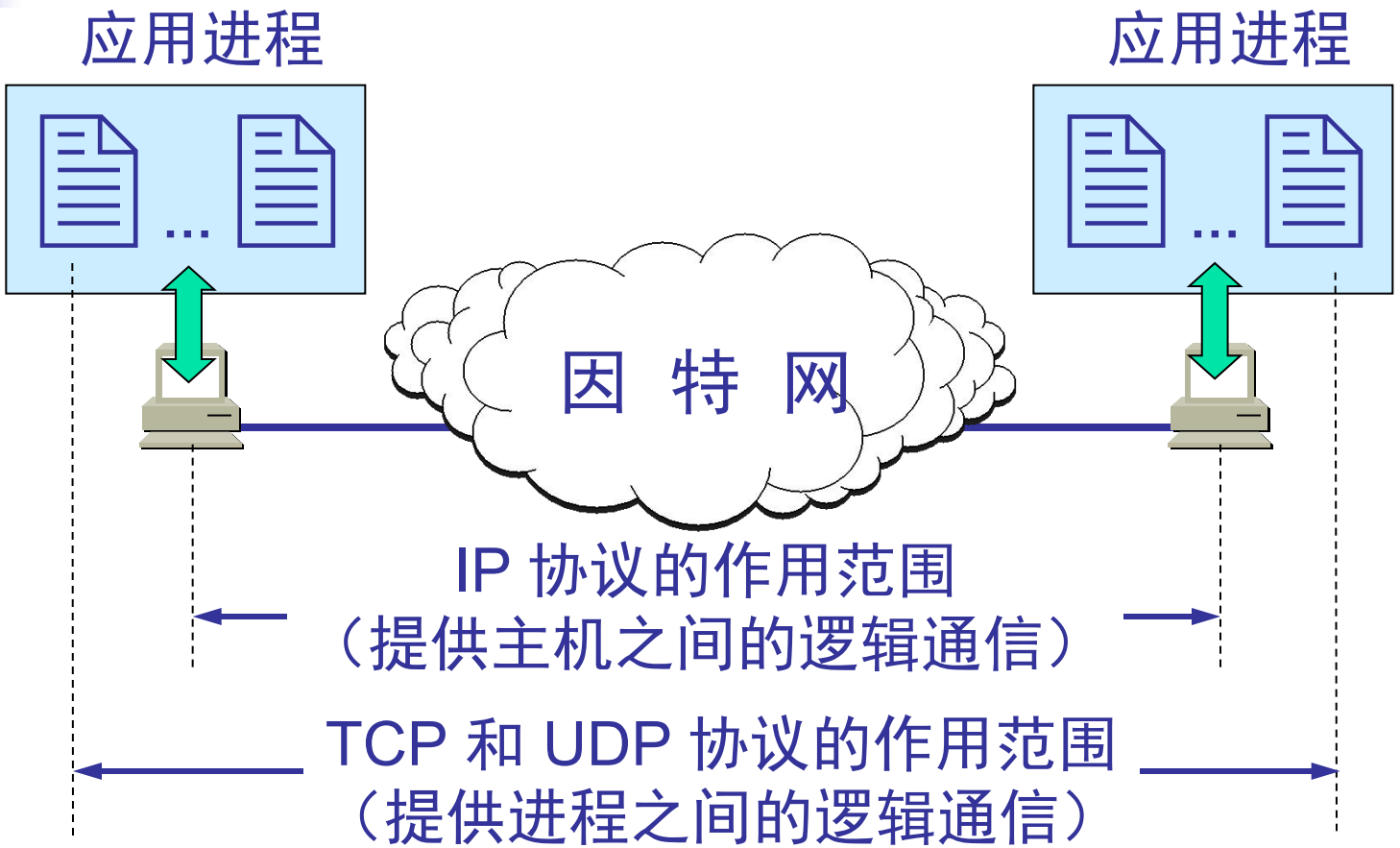


# 应用进程之间的通信

- 应用进程实现：应用层或用户自己实现。
- 当一个应用进程需要与另外一个远程进程通信时，它必须指明要与哪个应用进程进行通信（主机上存在多个进程）；
  - 需要对不同进程进行编号：端口号
- 网络环境下标识一个应用进程采用二元组
  - IP地址 + 端口号

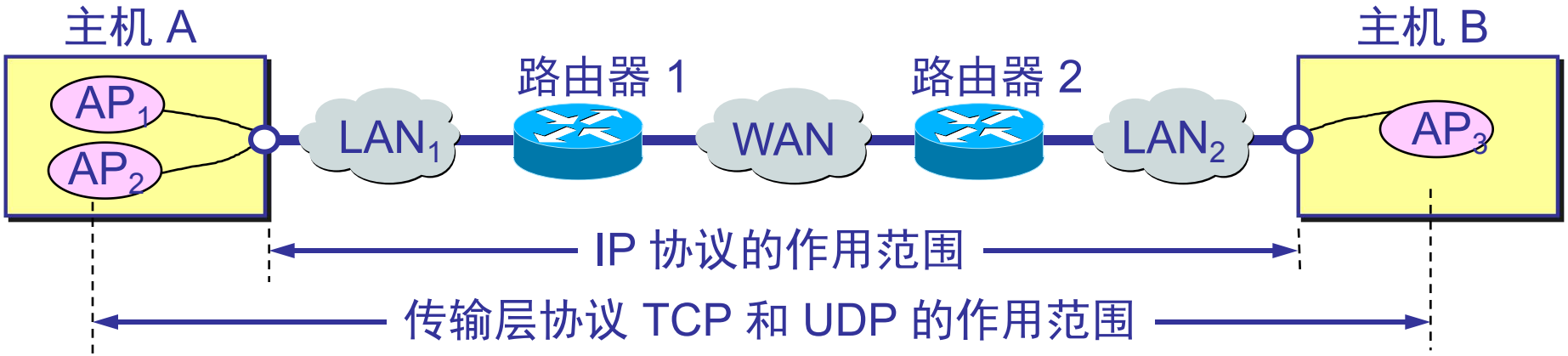


# 传输层协议和网络层协议的主要区别

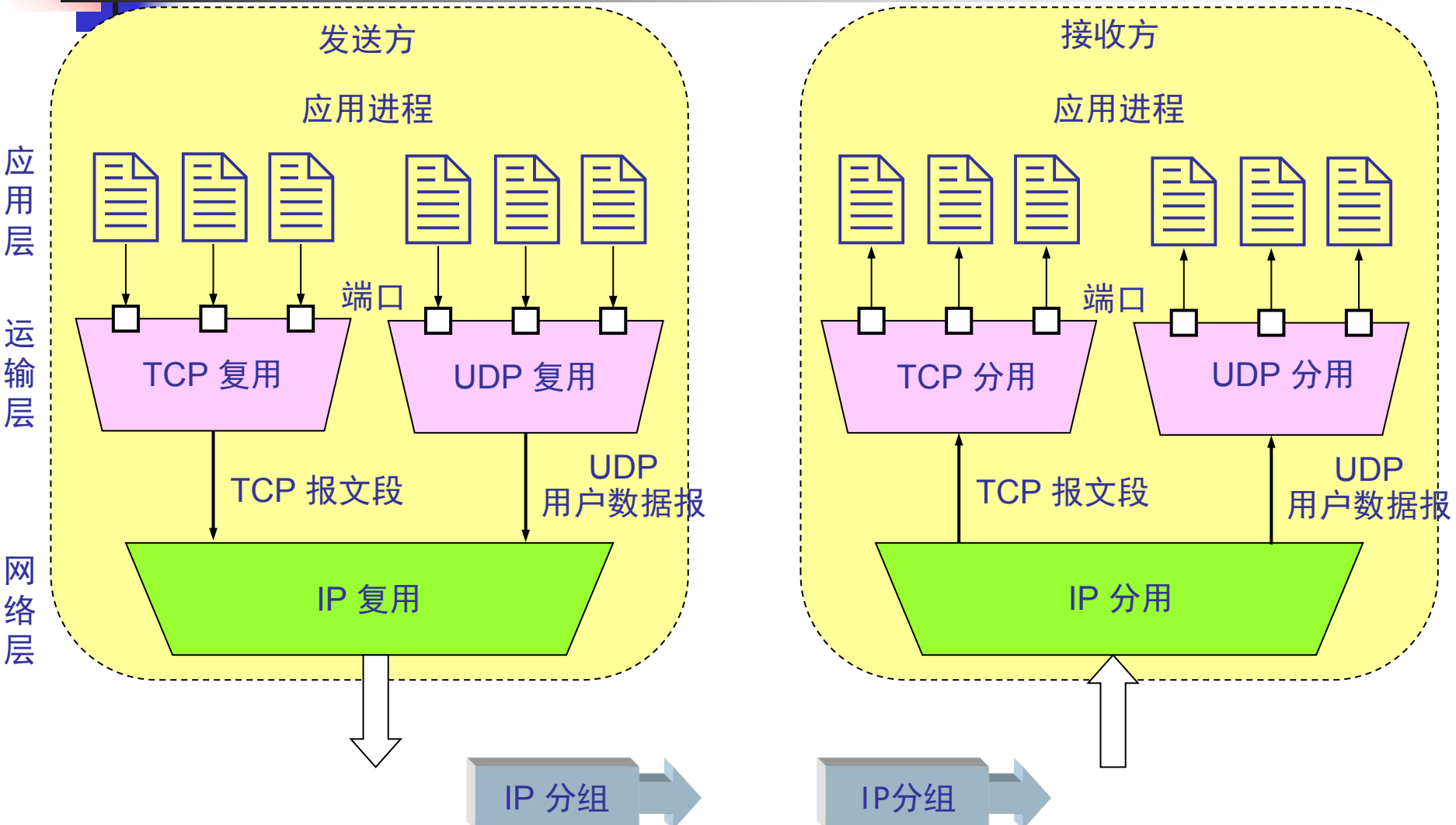


# 应用进程之间的通信

- 传输层对不同应用进程的编号（编址）：端口号（port），术语称为TSAP（传输层服务访问点，Transport Service Access Point。
  - 多个（本地或异地）进程也可以同时与一个远地进程进行通信。
  - 一个本地进程可以同时与多个远地进程进行通信。

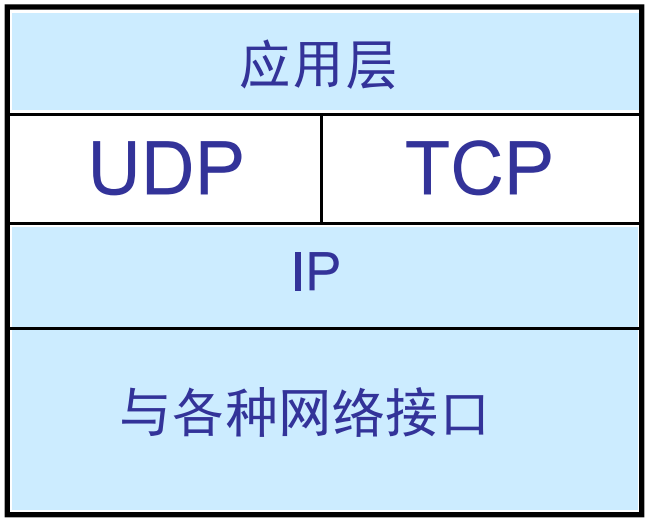


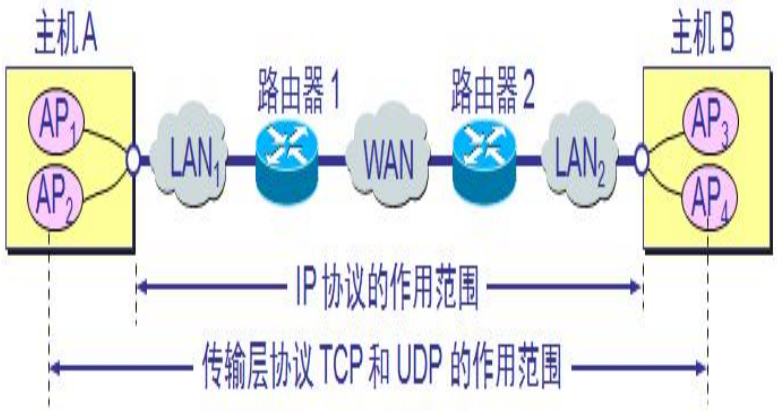
# 端口复用和分用


发送方
应用进程
接收方
应用进程
应用层
运输层
网络层
端口
端口
TCP 复用
UDP 复用
TCP 分用
UDP 分用
TCP 报文段
UDP
用户数据报
TCP 报文段
UDP
用户数据报
IP 复用
IP 分用
IP 分组
IP分组

# 6.1.2 传输层的两个主要协议

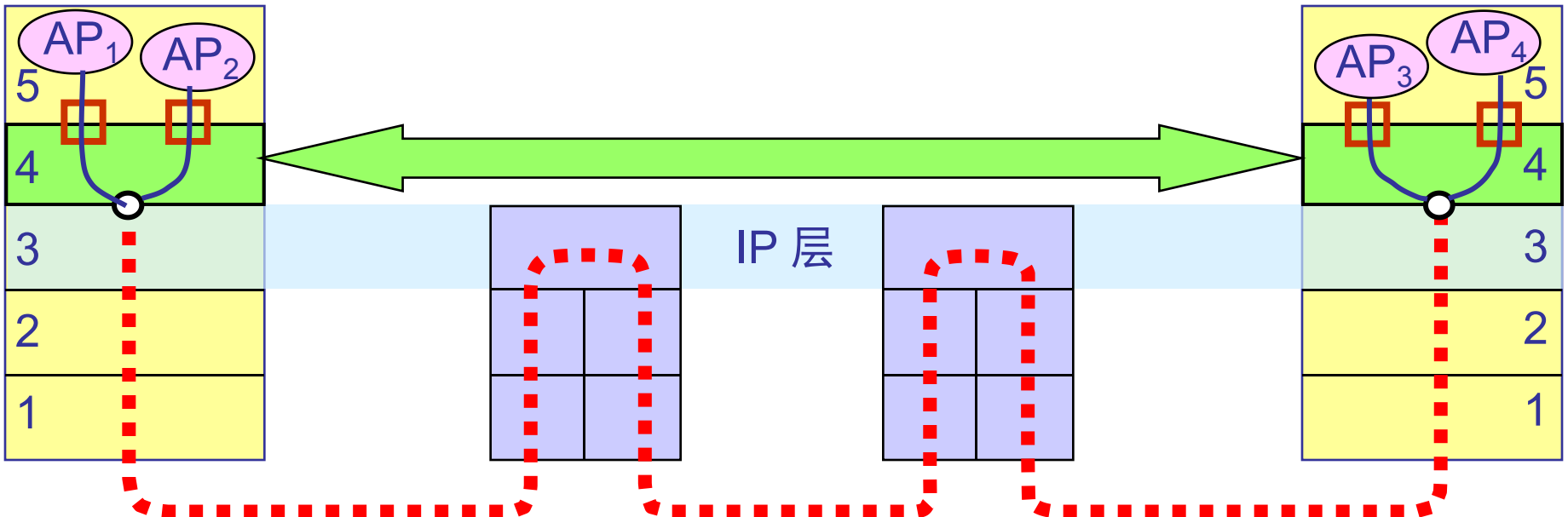
- 传输层提供两个协议。
  - UDP：用户数据报协议；
  - TCP：传输控制协议；
- UDP协议：提供无连接服务，在传输层建立一条不可靠通信信道。
- TCP:提供面向连接服务，尽管下面网络层是提供尽最大努力交付服务，但TCP协议实体通过软件在传输层提供一条全双工的可靠端到端通信信道。



# 6.1.2 传输层的两个主要协议

- 传输层两个对等实体在通信时传送的数据单位叫作传输层协议数据单元TPDU（Transport Protocol Data Unit）。
- 两端节点必须调用同一个传输层协议实体；
- TCP 产生的协议数据单元是 TCP报文段(segment)
- UDP 产生协议数据单元是 UDP报文或用户数据报。

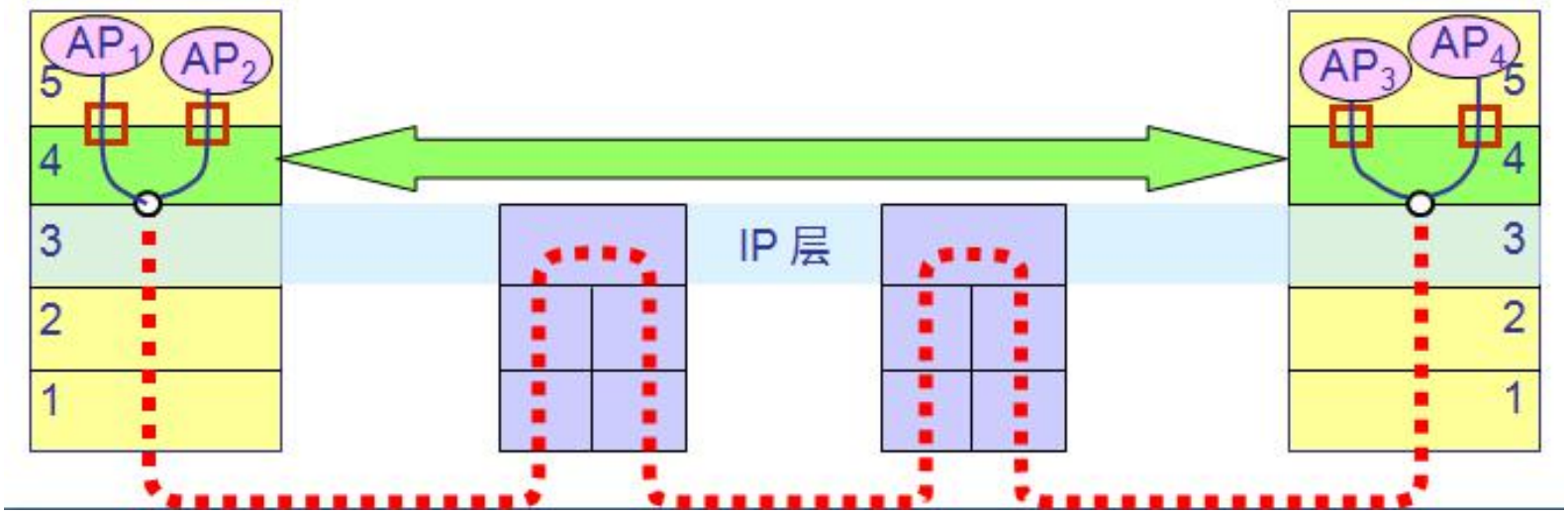


# TCP与UDP

- TCP 提供面向连接的服务
  - TCP 要提供可靠的传输服务，不可避免地增加了许多开销TCP报文段首部比较大，还要占用大量的主机资源。
  - TCP 不提供广播或组播服务，仅支持单播通信。
- UDP 提供无连接服务
  - 通信双方发送UDP用户数据报之前不需要先建立连接；
  - 接收方接收到UDP报文后，只检错，不给出任何确认。
  - 虽然 UDP 不提供可靠通信，但满足了一定实时通信服务需求，主要用于多媒体通信应用。
  - 支持单播、广播和组播通信；

# 强调三点

- TCP协议在通信时首先要通过三次握手建立一个TCP连接？建立连接是在传输层完成，路由器无法感知；
- TCP报文段和UDP用户数据报均以IP分组形式在网络传输，每一个IP分组在互联网中是独立路由，所以IP分组存在丢失、重复（？）、延迟、差错和乱序等传输问题，可靠性由TCP协议负责。

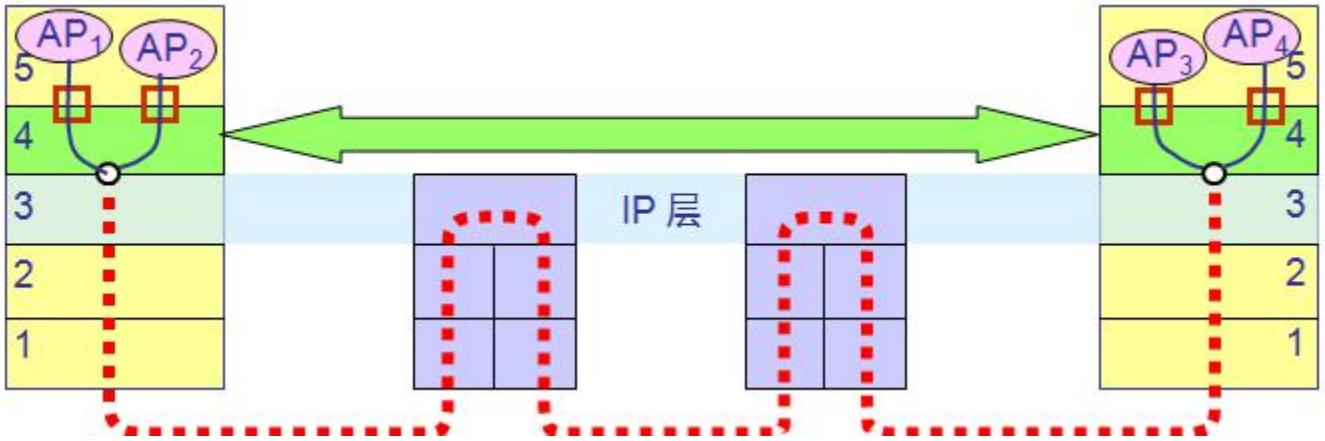


# 6.1.3 传输层的端口

- 在因特网上计算机使用的操作系统种类很多，不同操作系统又使用不同规范对进程进行标识编码。
- 在网络环境下，为了使运行不同操作系统计算机的应用进程能够互相通信，就必须用统一的规范对应用进程进行标识。
  - 传输层对应用进程进行统一编号（编址）：端口号
- 端口号是一个 16 比特位二进制数
- 端口号具有本地含义
  - 端口号在本地是唯一的:同一个主机，针对同一传输层协议，不能有两个或两个以上相同的端口号；
  - 不同主机可有相同端口号：代表不同主机不同进程，相互没有影响。

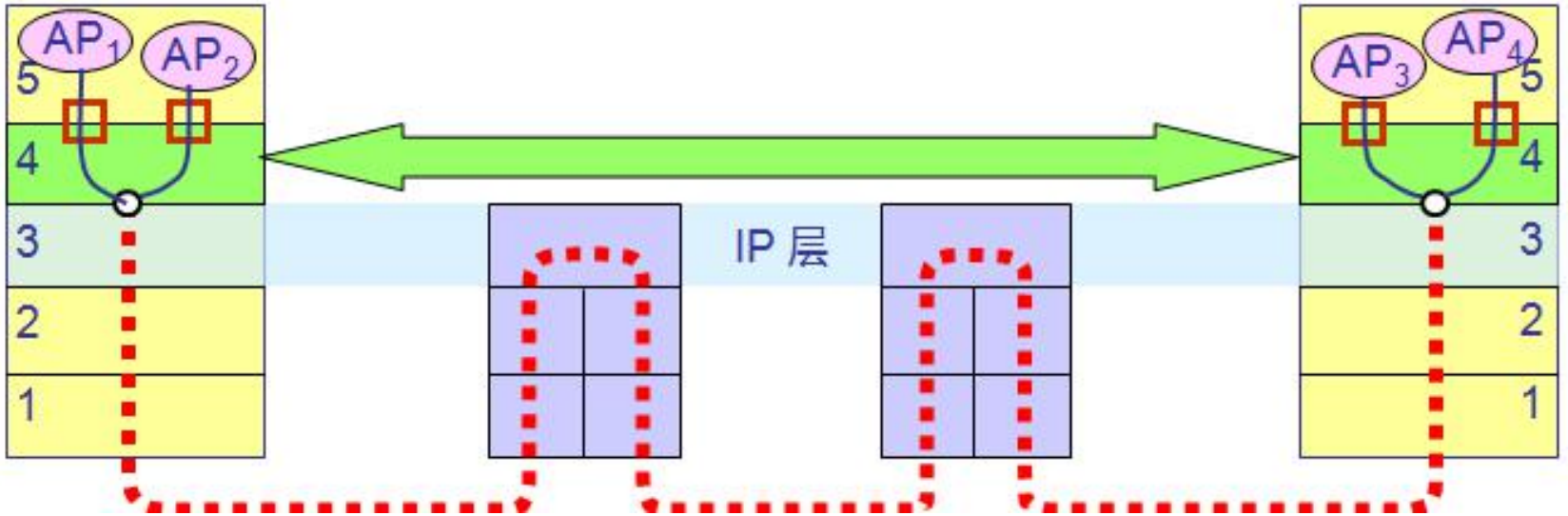
# 端口号(protocol port number)
# 简称为端口(port)

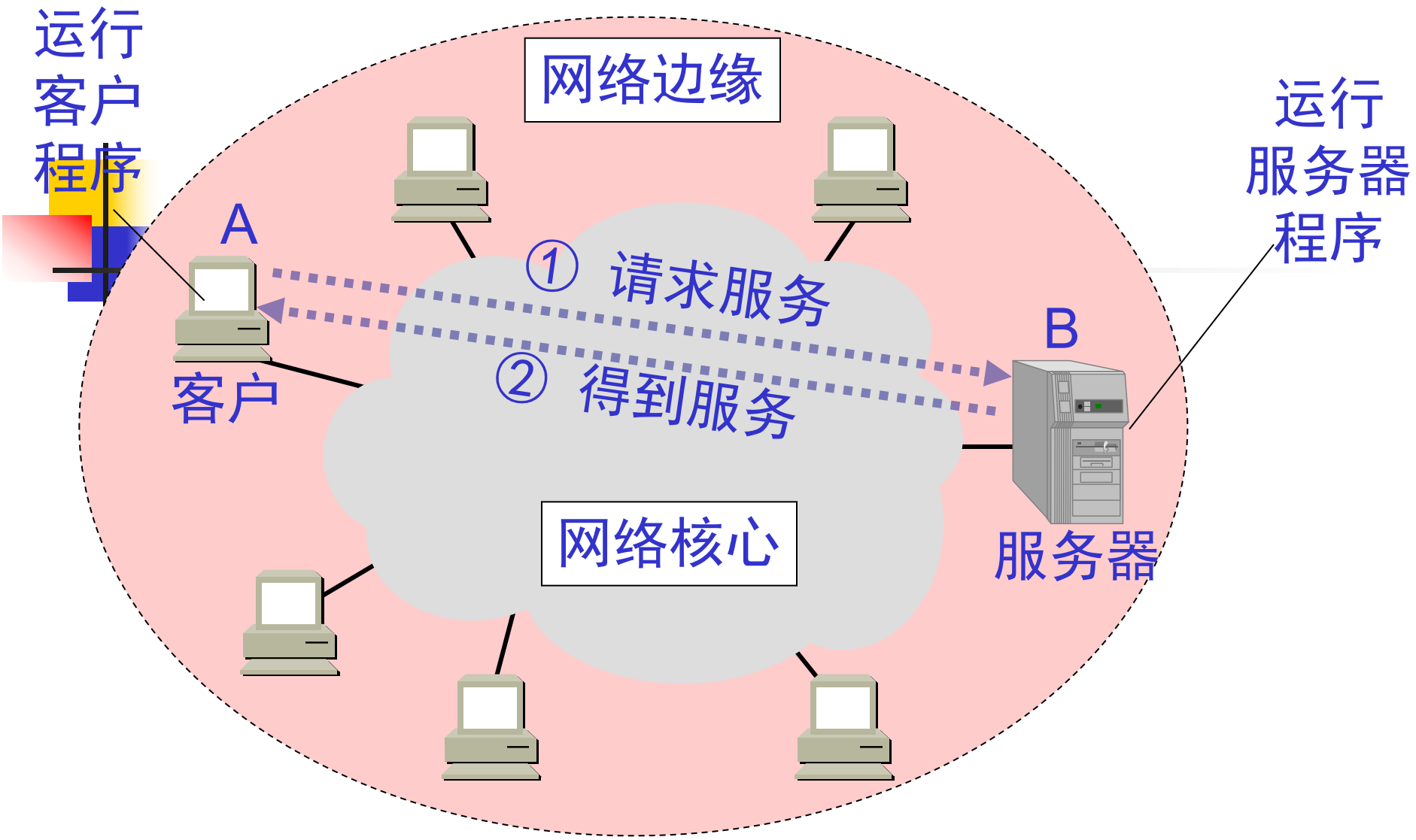
- 端口号作用
  - 发送方应用层的各种应用进程都能将其数据通过源端口号向下交付给传输层； – 端口复用
  - 接收方传输层根据接收到的数据单元首部中的目的端口号向上交付给应用层相应的应用进程。–端口分用



# 端口号(protocol port number)简称为端口(port)

- 端口号是用来标识应用层不同进程；
- 为什么网络通信时需要同时提供：目的IP+目的port和源IP+源port
  - 一般采用全双工通信模式，除了发送方发送数据给接收方外，接收方也需要发送数据（或应答）给发送方。
  - 五元组标识通信双方：源IP、源port + 目的IP、目的port +协议






客户 A 向服务器 B 发出请求服务，而服务器 B 向客户 A 提供服务。

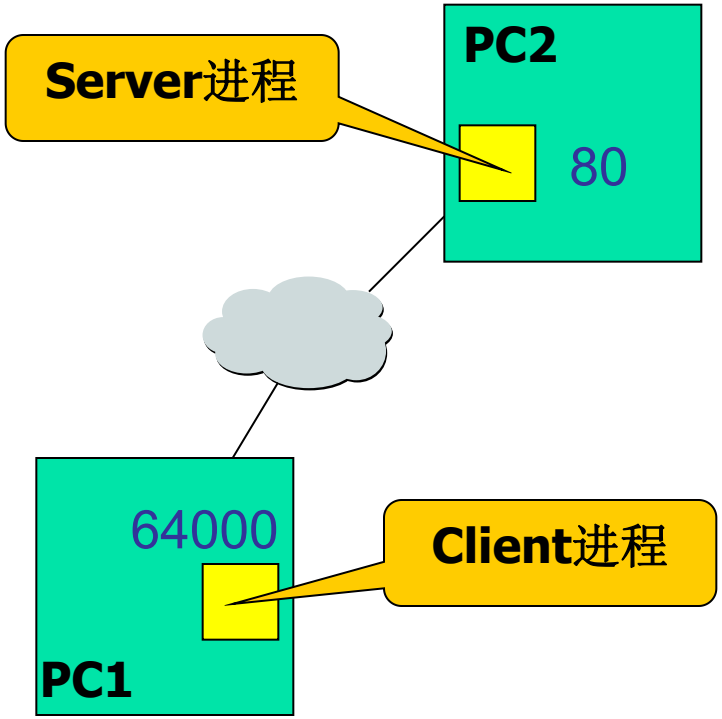
多对一通信关系

# 端口号分类

- 服务器端口号
  - 周知端口号：数值一般为 0~1023，IANA组织管理和分配。
  - 注册端口号：数值为1024~49151，为没有熟知端口号的应用服务器程序使用；这个范围的端口号必须在 IANA 登记，以防止重复使用。
- 客户端口号（或动态端口号）：数值为49152~65535，留给客户应用程序暂时使用，由操作系统临时分配。
  - 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，客户端口号立即释放，可供其他客户程序以后使用。

# 服务器周知端口号

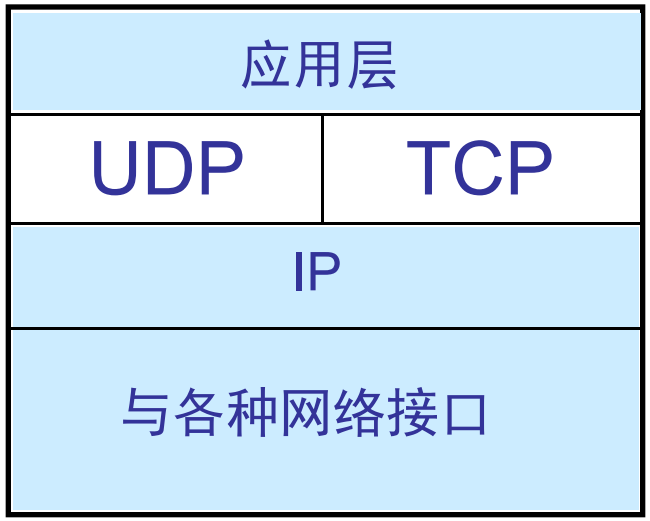
- 见教材PP184页, 表5. 3为TCP周知端口, 表5. 4为UDP周知端口.




| 应用层协议 | 周知端口 | 功能 |
|---|---|---|
| DNS UDP | 53 | 域名解析 |
| TFTP UDP | 69 | 简单文件传输 |
| DHCP UDP | 67/68（S/C） | 动态地址分配 |
| SNMP UDP | 161 | 简单网络管理 |
| SNMP UDP (Trap) | 162 | 简单网络管理 |
| HTTP TCP | 80 | 超文本传输 |
| FTP TCP | 20d, 21c | 文件传输 |
| Telnet TCP | 23 | 远程登录 |
| SMTP TCP | 25 | 简单邮件传输 |
| …… | …… | …… |
| | | |

# 6.2 用户数据报协议 UDP
## 6.2.1 UDP 概述

- 用户数据报协议UDP：在IP协议（无连接数据传输服务）基础上，增加了端口号和差错检测(可选)功能，提供端到端进程不可靠传输。
- 首部只有8个字节， TCP首部至少要20个字节。




# 6.2 用户数据报协议 UDP
## 6.2.1 UDP 概述

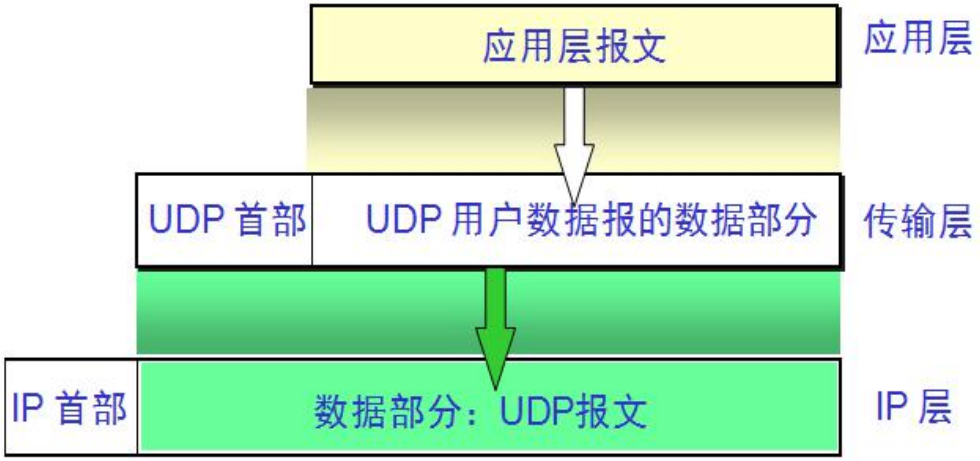
- 适用于实时通信应用系统(如IP电话, 视频会议)
  - 要求主机以恒定速率发送UDP报文, 允许一定程度UDP报文丢失, 但不允许端到端传输有太大延迟或延迟抖动时间。
- 在航空航天领域某些应用既需要实时性，也需要可靠性。
  - 解决方法：采用UDP协议，在不影响实时通信前提下，在应用层增加一些简单的可靠性措施，如对丢失UDP用户数据报进行简单重发;
  - 如果直接采用TCP协议，可靠性可以保证，由于TCP协议太复杂，并且系统资源开销比较大，无法保证实时性要求。

# UDP 协议通信特点

- UDP 是无连接：发送UDP报文之前不需要建立连接；数据传输结束后，不需要释放连接，减少了处理开销和发送延迟。
- UDP 使用尽最大努力交付，不保证可靠性，端节点不需要维护许多参数，定时器，复杂连接状态表等。
- UDP 支持一对一、一对多的交互通信。
- 不提供拥塞控制，网络发生拥塞发送端不会降低发送速率，此时网络会出现什么现象？
  - 如果网络发生拥塞，TCP有拥塞控制机制，会适当的降低发送方发送速率；UDP协议发送速率不会降低，造成网络中存在大量UDP报文-UDP/TCP通信的不公平现象。

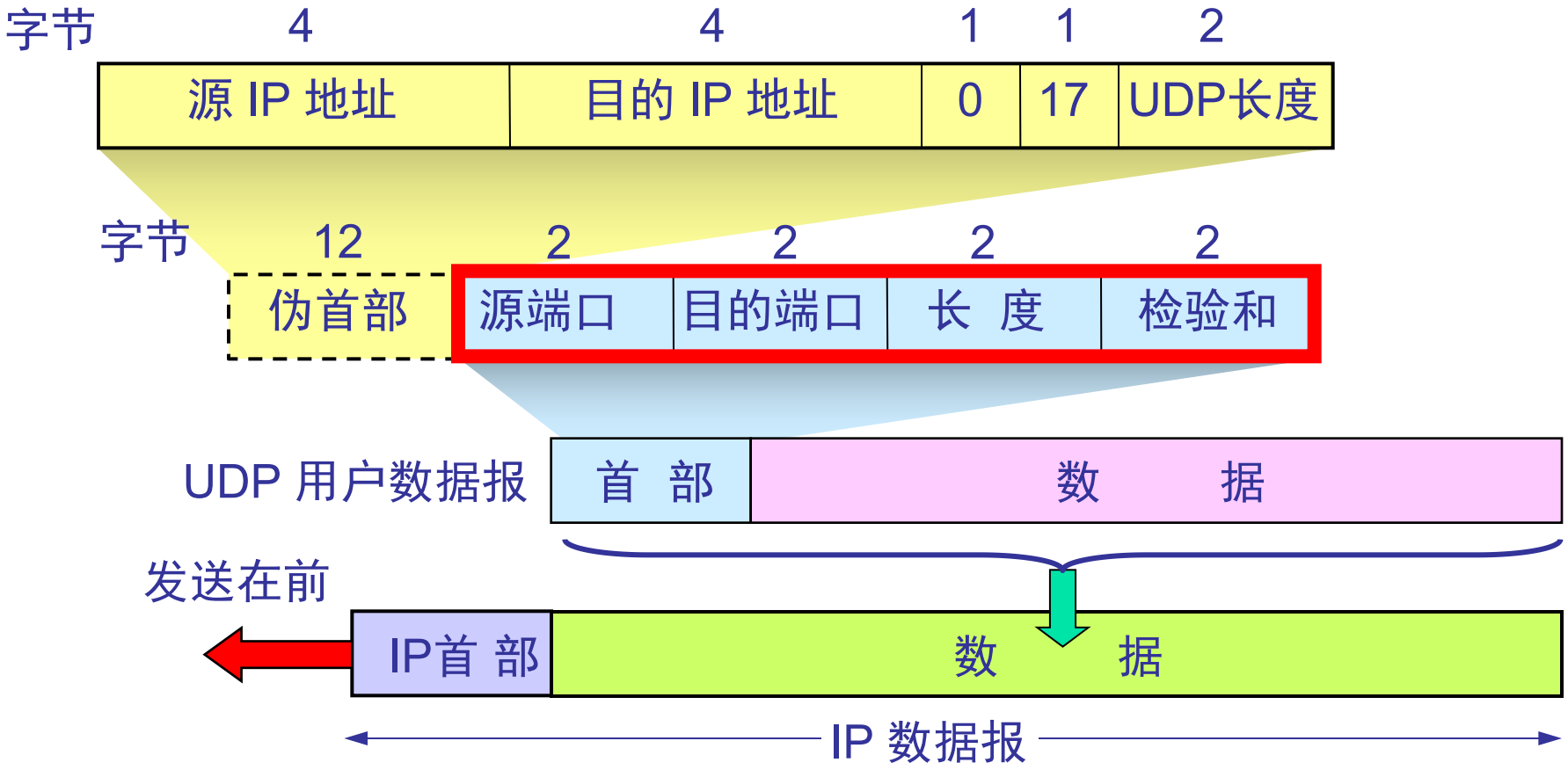
# UDP用户数据报

- 发送方 UDP 对应用进程交付下来协议数据单元，在添加首部后就向下交付 IP 层。
- 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。

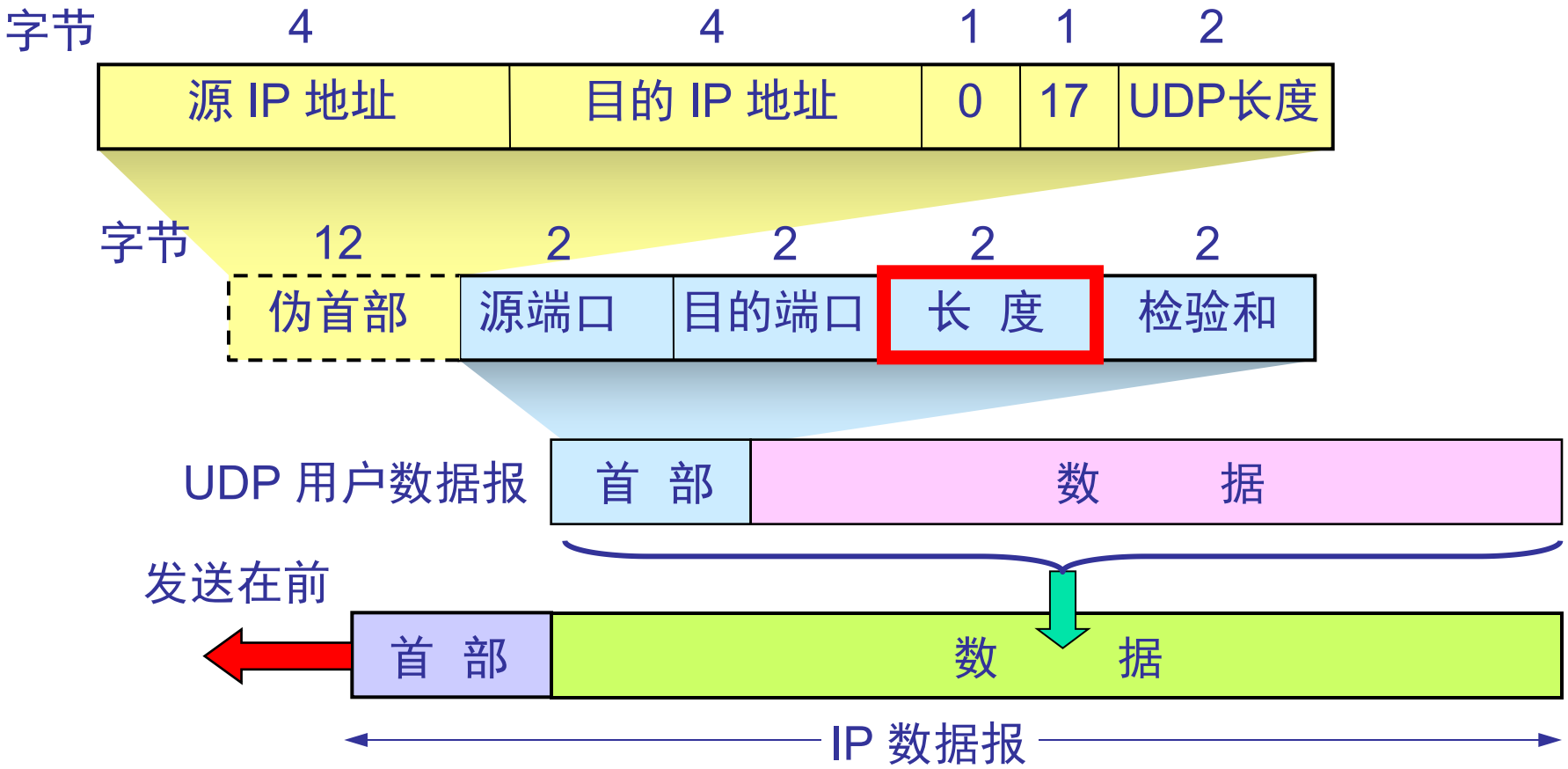



应用层进程产生的协议数据单元大小要合适：如果太大在IP层需要分片；如果太小传输层首部相对太大，降低了网络传输效率。

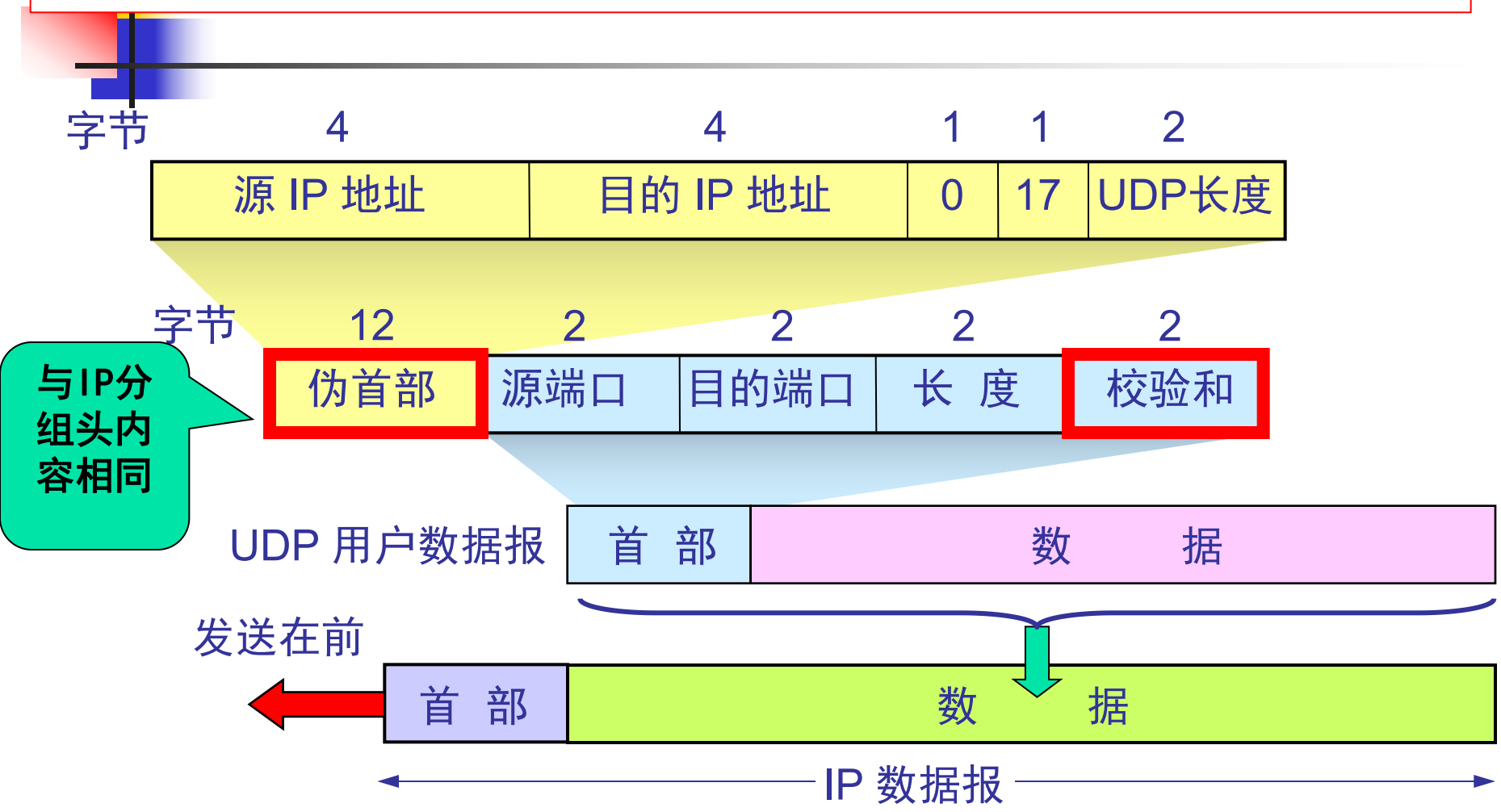
用户数据报 UDP 有两个部分：首部和数据；首部有 8 个字节，由 4 个字段组成，每个字段都是2个字节。



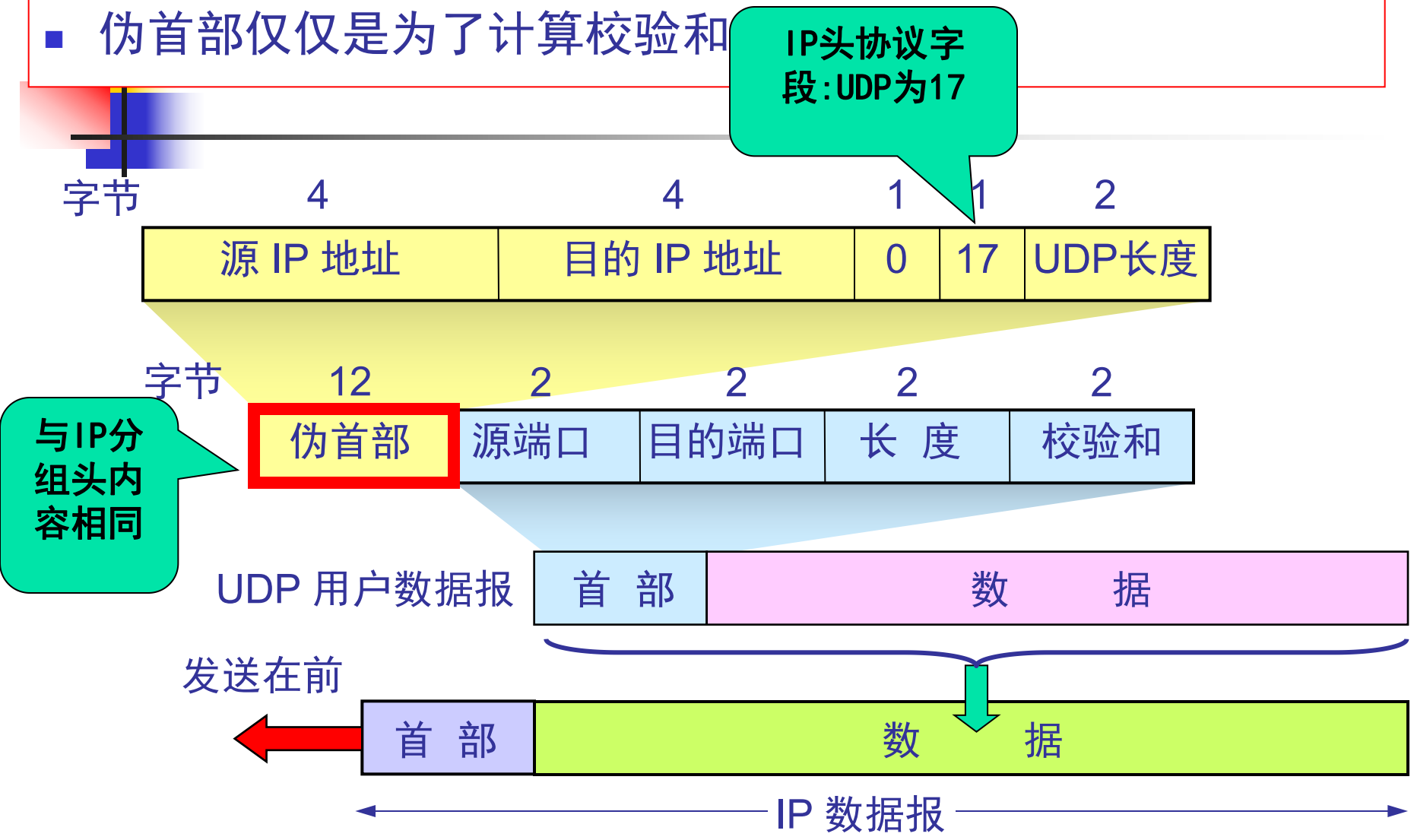
长度:用户数据报长度,包括UDP首部+数据,以字节为单位,最小为8个字节(UDP头长度).



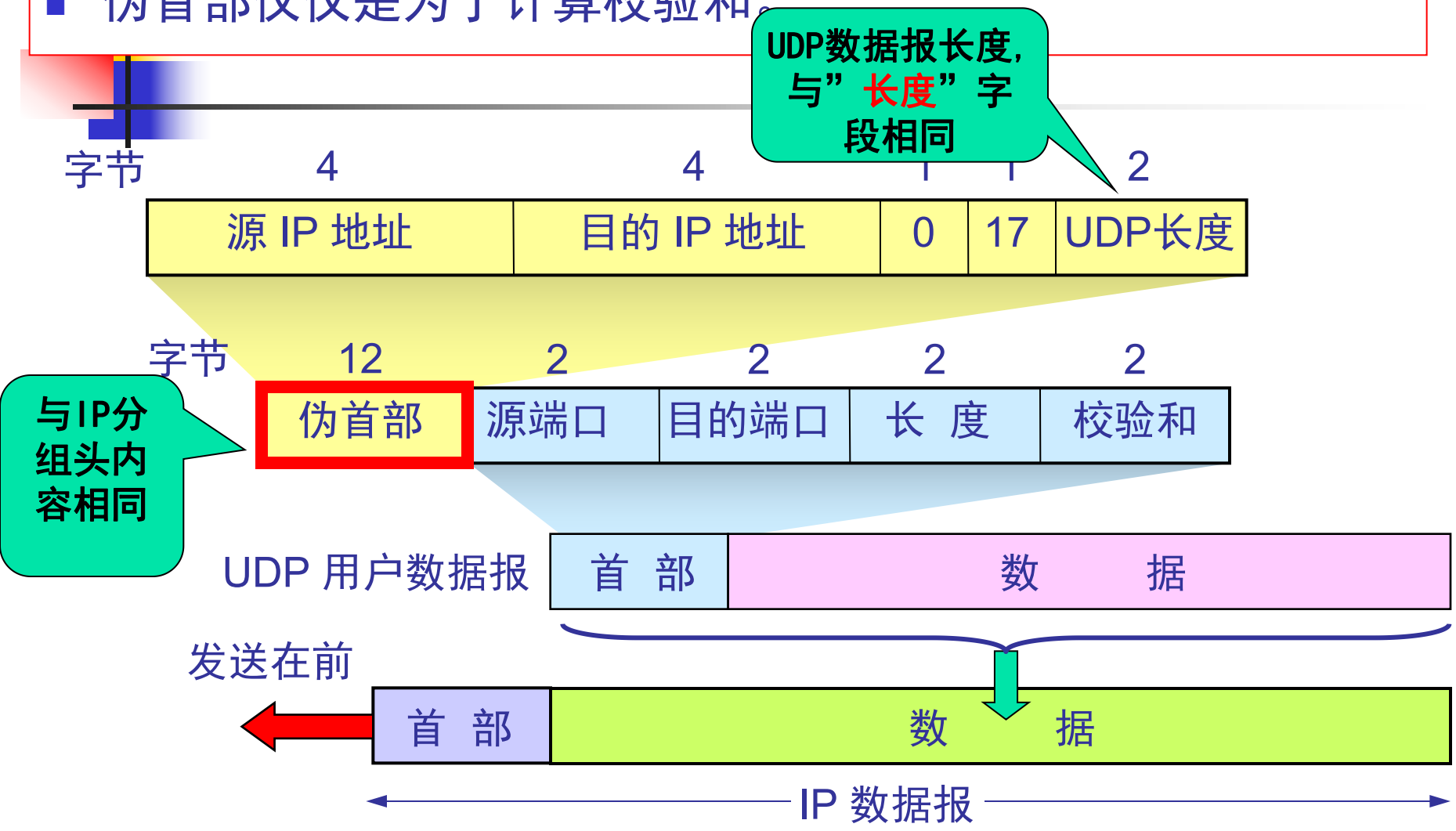
- 计算检验和时，临时把“伪首部”和 UDP 用户数据报连接在一起计算：伪首部+UDP首部+数据
- 伪首部仅仅是为了计算校验和，计算结束后丢弃。



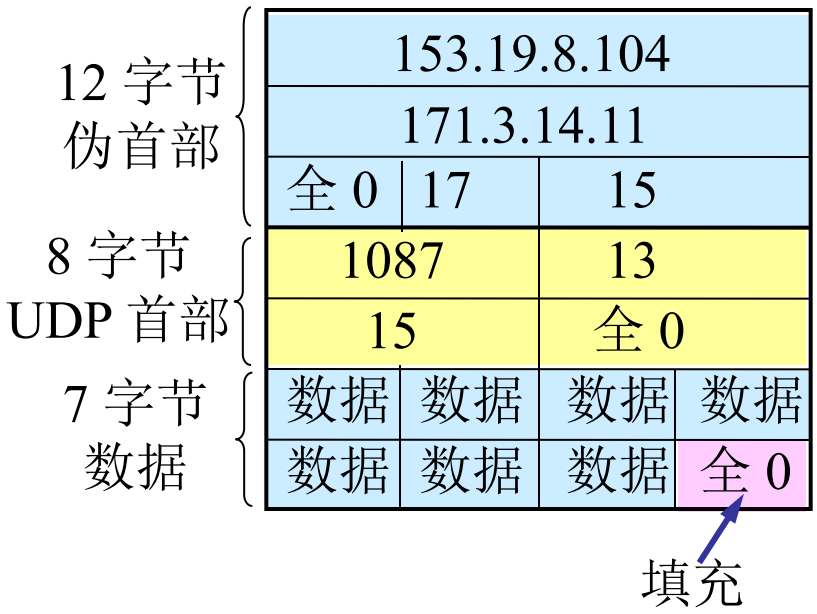
- 计算检验和时，临时把“伪首部”和 UDP 用户数据报连接在一起。
- 伪首部仅仅是为了计算校验和



- 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校验和。



# 计算 UDP 检验和的例子




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）

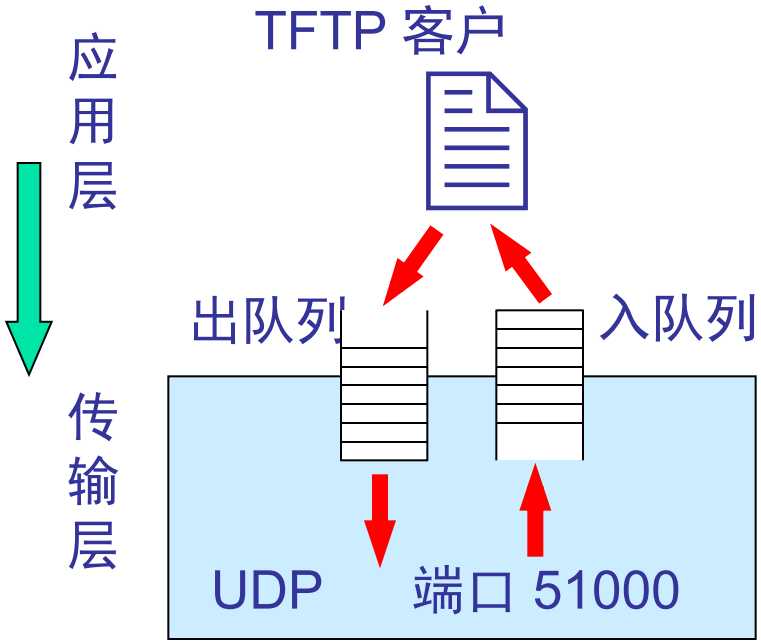
---

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

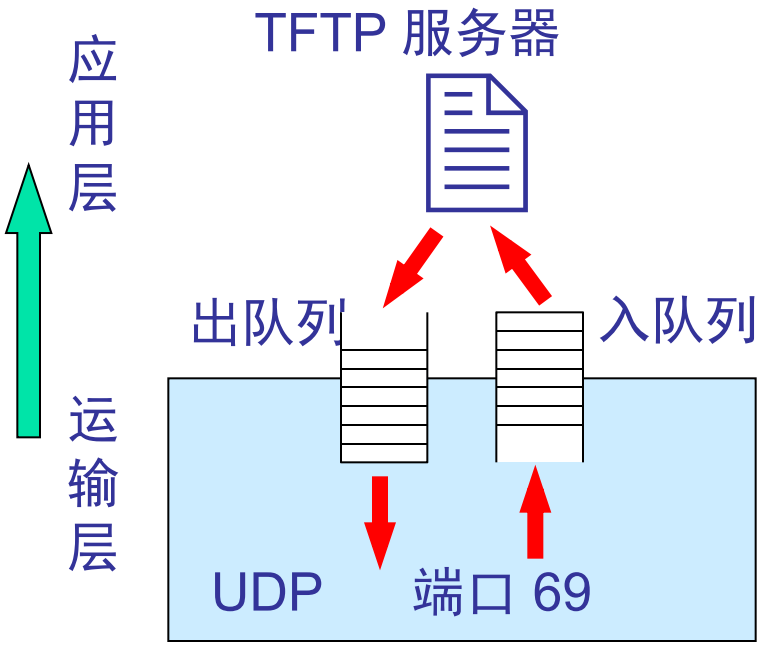
# （1）UDP用户数据报发送

- UDP用户数据报发送
  - 发送方应用进程发送数据到出队列（有边界），UDP按照队列顺序，取出数据（按照应用层交付数据大小）加上UDP头(填写源与目的端口号)，交付给IP层。
  - IP协议将整个UDP数据报封装在IP分组中，交付到下层。
- 注意事项：
  - 如果出队列溢出，UDP告诉应用层客户进程暂停发送数据。
  - 网间寻址由IP地址完成，进程间寻址则由UDP端口来实现。
  - 一个客户进程与多个远程进程通信使用一个出队列。



# (2)UDP数据报接收

- UDP数据报接收
  - UDP进程实体接收数据报时,首先检查校验字段(可选),如果出错,丢弃但不发送ICMP差错报告;
  - 否则，UDP实体根据目的端口号查找对应的入队列;
  - 如果找到，则将数据放入相应接收队列，向上层交付按照原始数据大小交付。
  - 否则丢弃，并向源端发送一个“端口不可达”的ICMP差错报告报文;
  - 当接收队列已满时，数据也要丢弃,并向源端发送一个“端口不可达”的ICMP差错报告报文;
  - 多个客户向一个服务器发送用户数据报,都在同一服务器入队列统一接收排队。



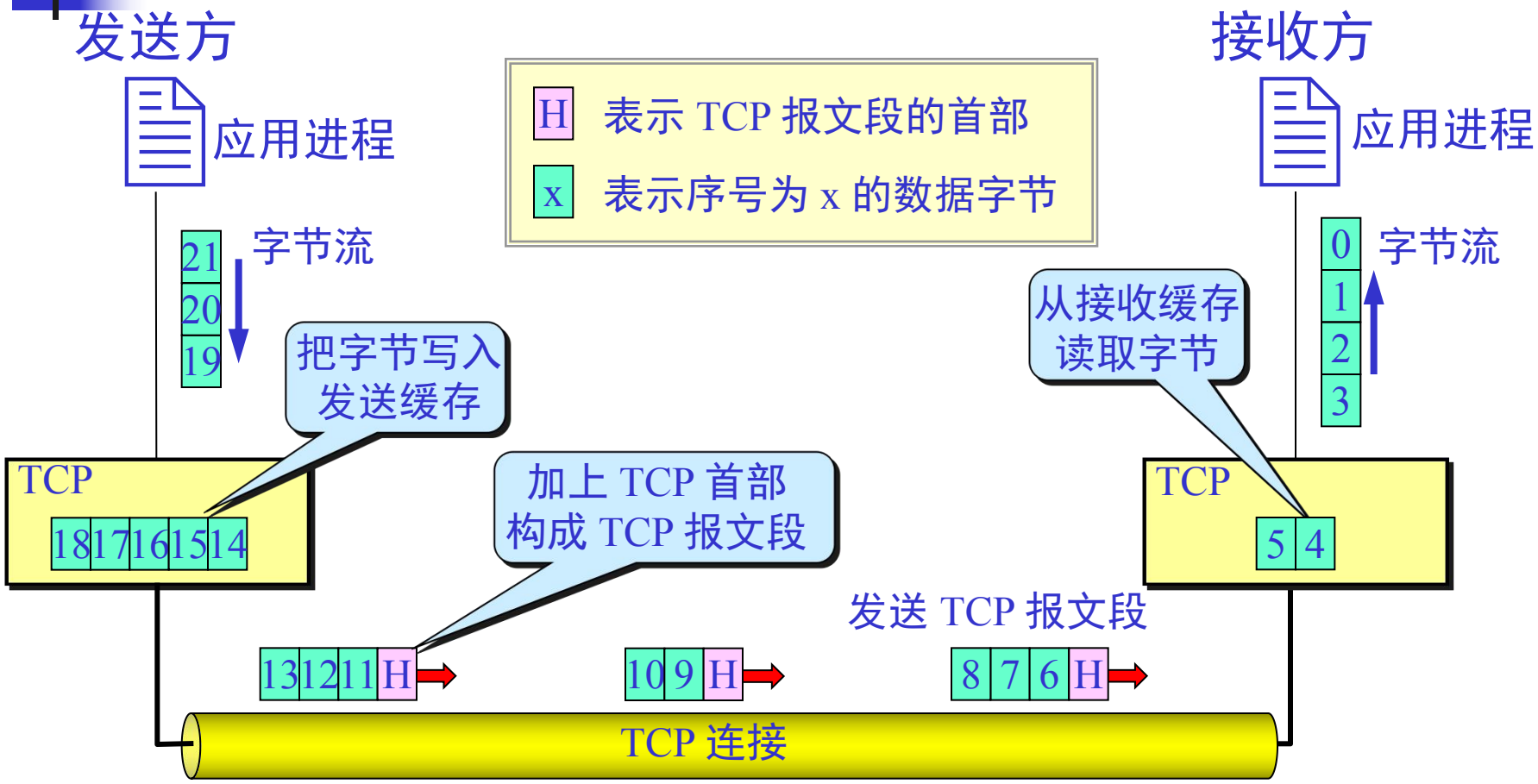
# 6.3 传输控制协议 TCP 概述
## 6.3.1 TCP 最主要的特点

- TCP是面向连接的传输层协议。
  - 通信前通过三次握手建立连接；通信过程中对该连接进行维护；通信结束后释放连接。
  - 连接是一个虚连接，不是真真的物理连接；
- 每一条 TCP 连接只能有两个端节点(end-point)，每一条 TCP 连接只能是一对一通信（单播通信）。

# 6.3 传输控制协议 TCP 概述
## 6.3.1 TCP 最主要的特点

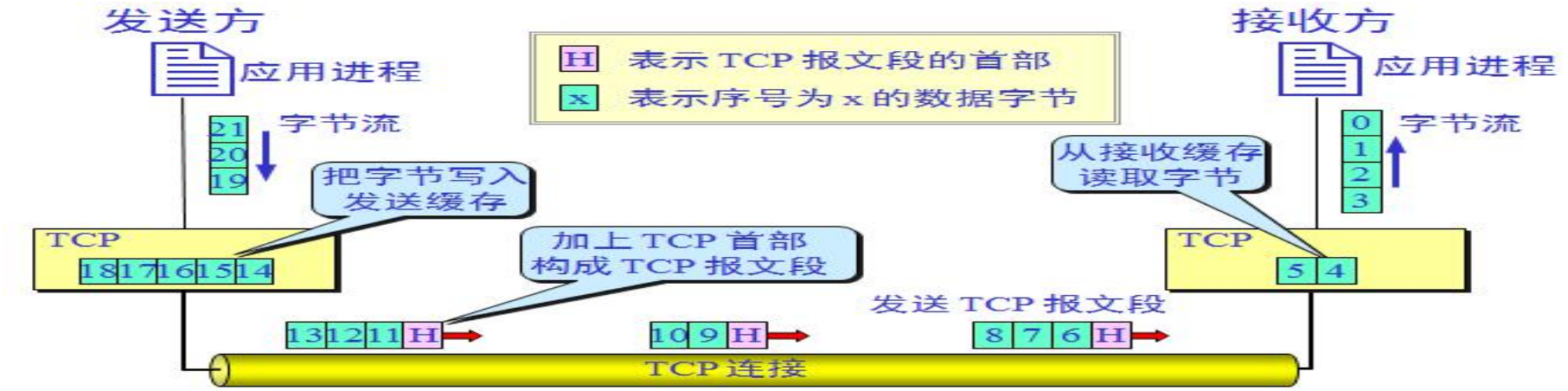
- TCP 提供传输层可靠通信服务。
  - 差错控制：序号+确认+超时重发；
  - 流量控制：采用可变大小的滑动窗口机制
  - 拥塞控制:慢启动、拥塞避免、快速重传、快速恢复
- TCP 提供全双工通信。
- 面向字节流，对TCP数据段中数据部分每一个字节都编有序号。

# TCP 面向字节流概念

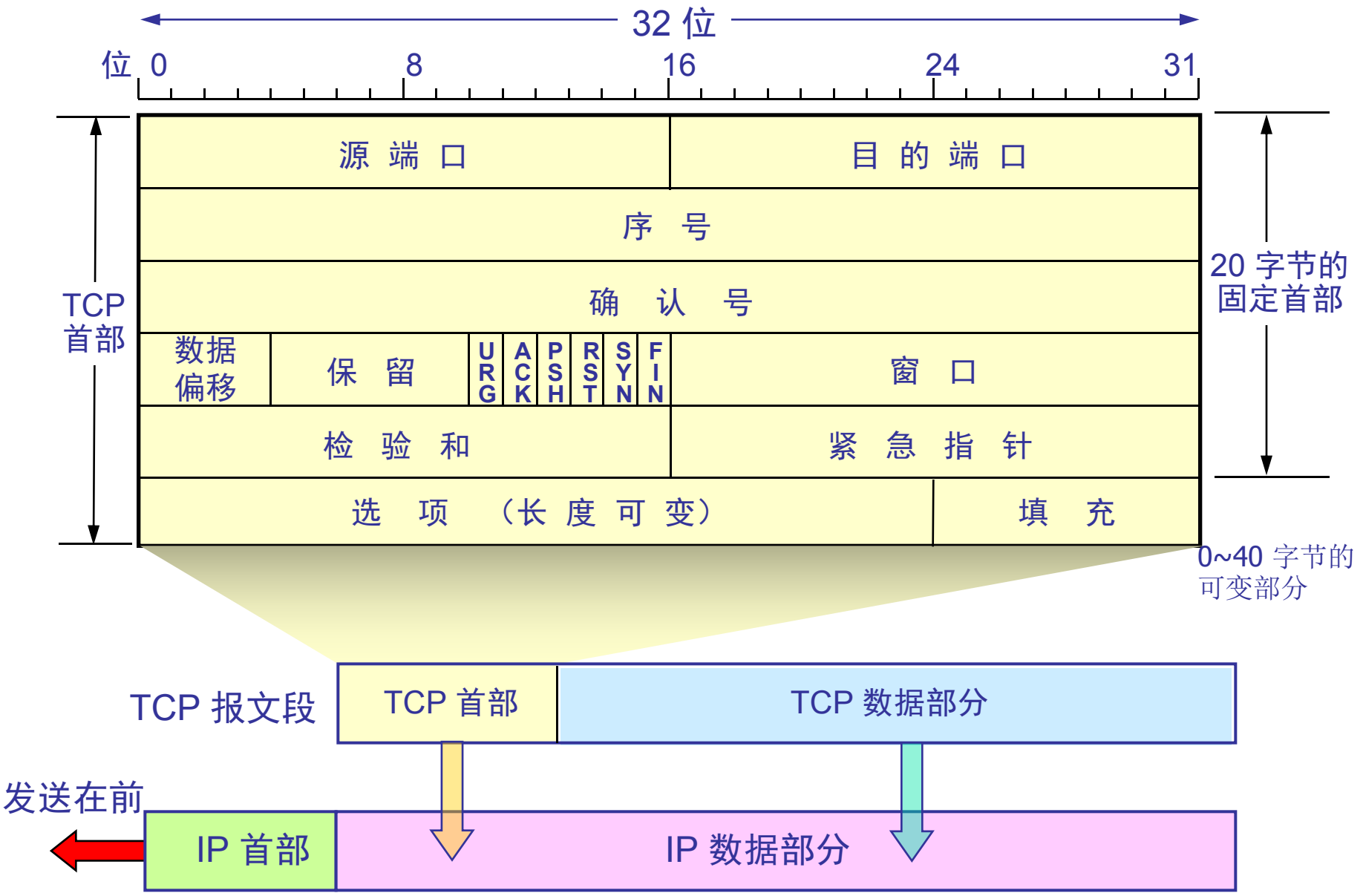

发送方
应用进程
字节流
21
20
19
把字节写入
发送缓存
TCP
18 17 16 15 14
H 表示 TCP 报文段的首部
x 表示序号为 x 的数据字节
加上 TCP 首部
构成 TCP 报文段
13 12 11 H
10 9 H
8 7 6 H
发送 TCP 报文段
TCP 连接
接收方
应用进程
字节流
0
1
2
3
从接收缓存
读取字节
TCP
5 4

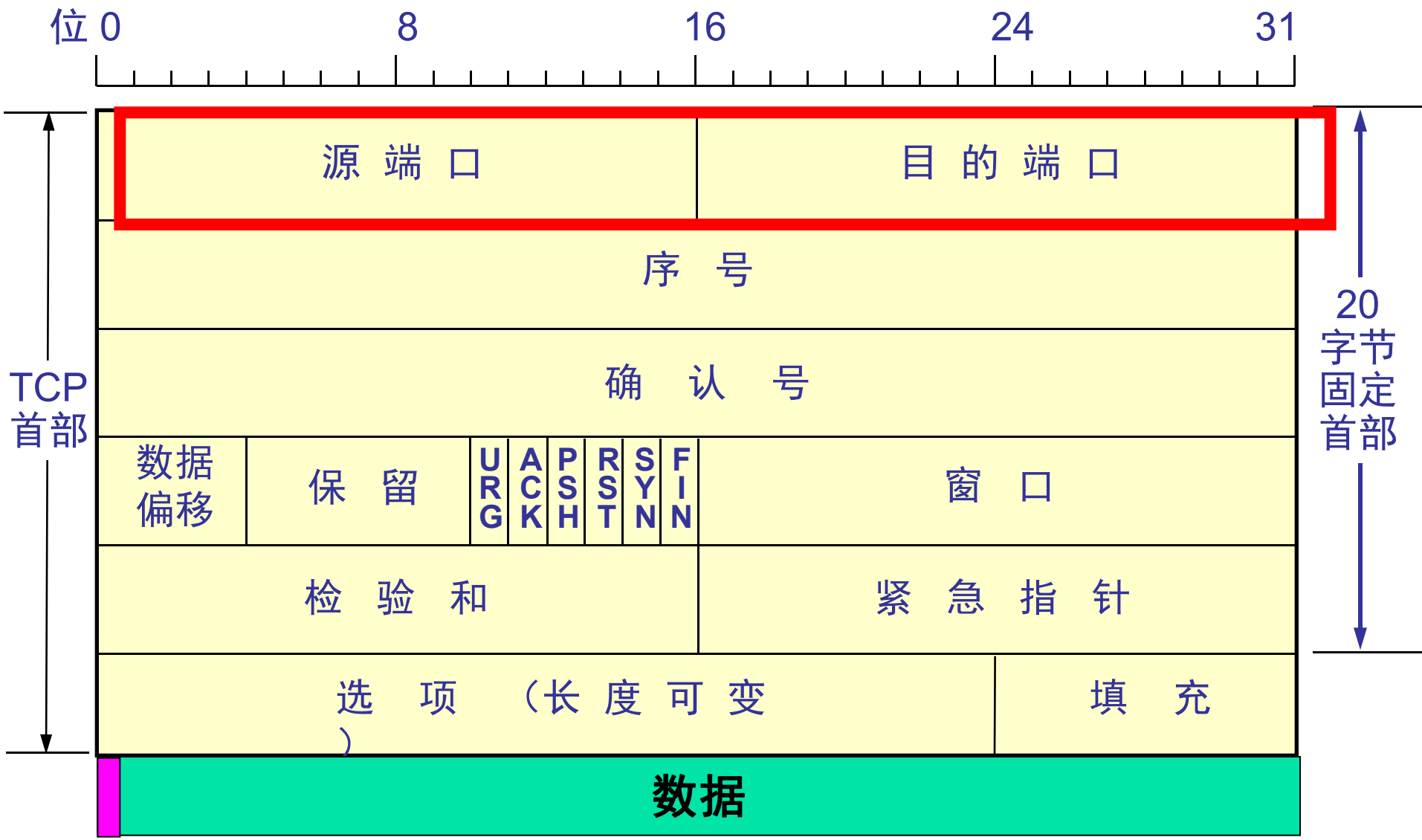
# 注意要点

- TCP对应用进程一次把多大数据交付到TCP 发送缓存中不关心
  - 发送方应用进程可能交付给传输层TCP有10个数据块；但TCP有可能只用了3个TCP报文段就发送完。
  - TCP 会根据接收方给出窗口值、当前网络拥塞程度和MSS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确定）。
- 结论：TCP 可把发送缓存中一个大数据块分割为若干个TCP报文段再发送；也可等待发送缓存中积累有足够多字节的数据后后再构成一个TCP报文段发送出去。

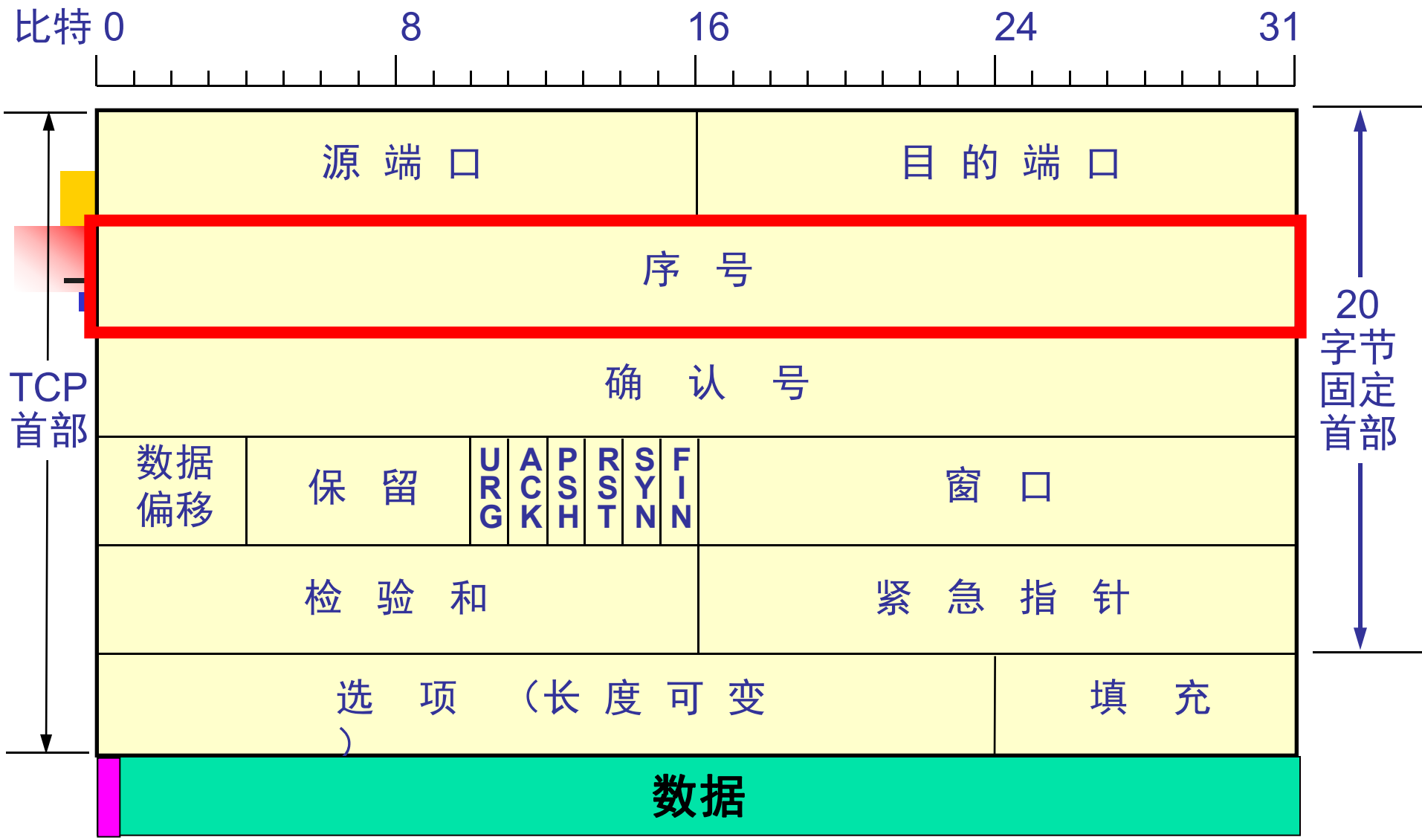


# 6.3 TCP 报文段语法与语义



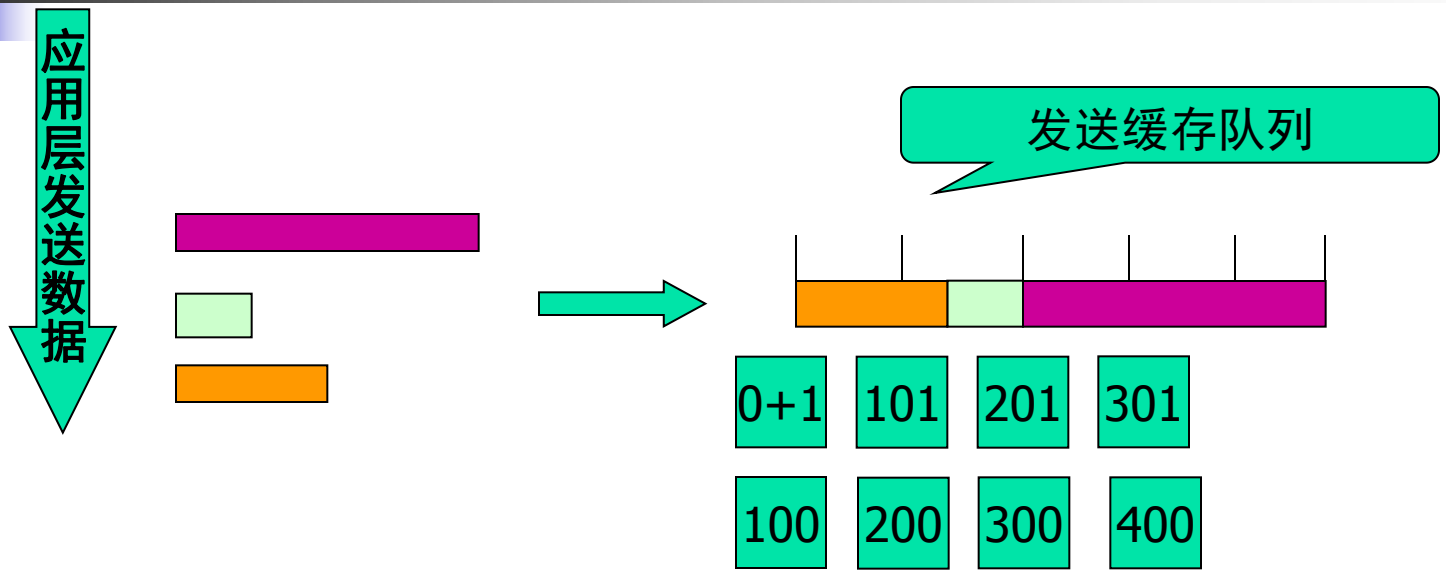



- 源端口、目的端口：各占 2 字节
  - 端口是传输层提供给应用层的TSAP地址；
  - 传输层复用和分用功能都要通过端口号实现。



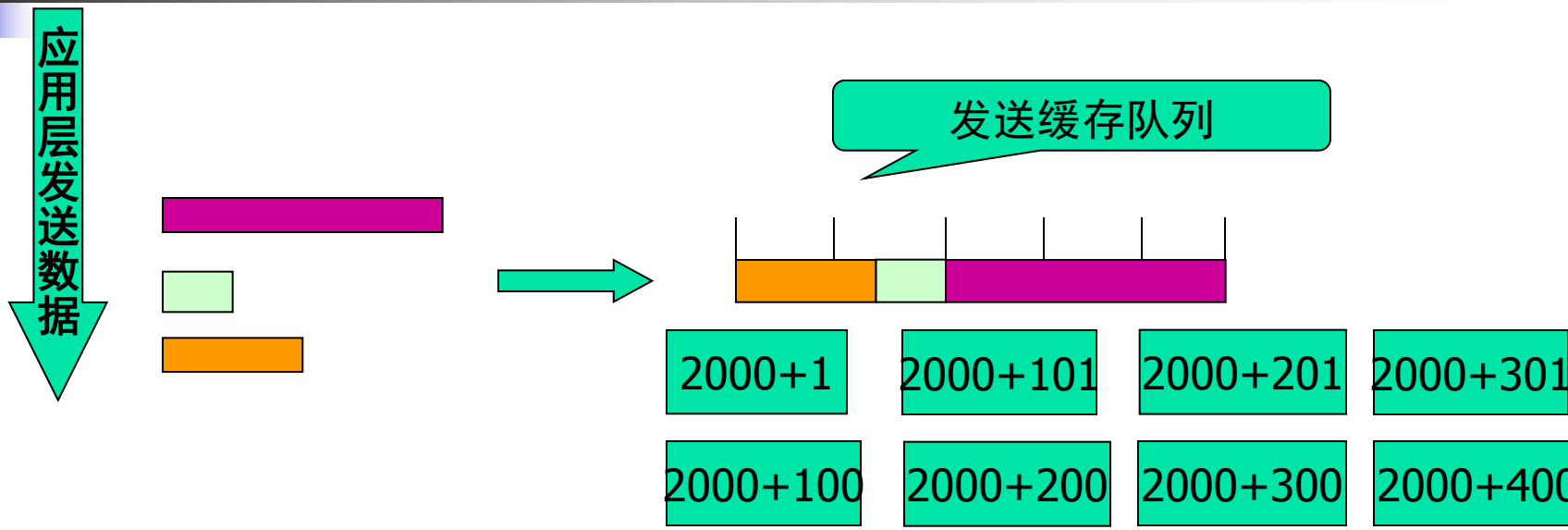

- 序号：占 4 字节。
  - TCP 连接中传送的数据流（字节流）中每一个字节都编上一个序号。
  - 含义：TCP数据段的数据字段第一个字节在字节流中相对序号。

# 举例：序号（A发送TCP报文段给B）

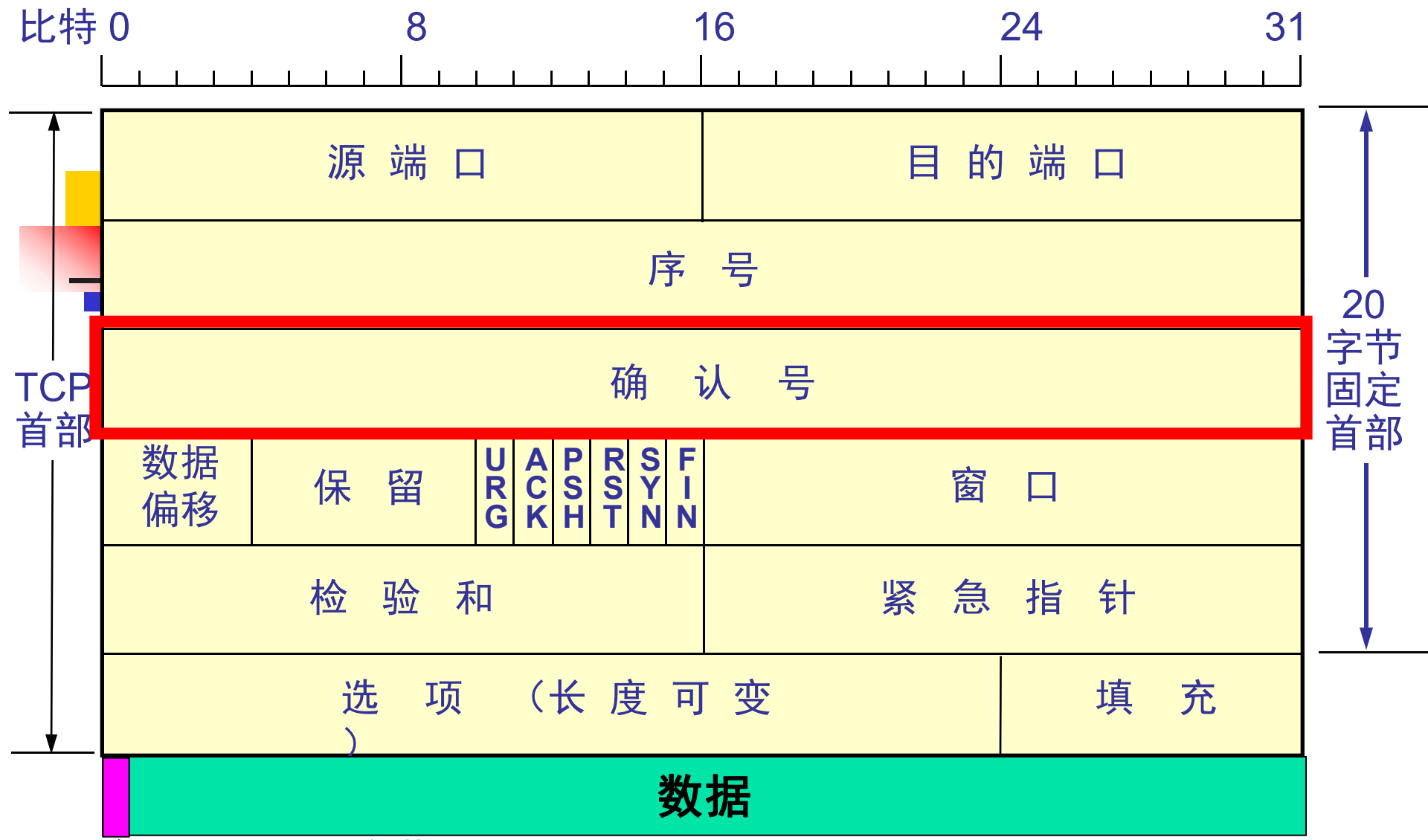



- 如果协商A发送的初始序号为0，数据长度100字节。
  - 字节流第1个字节序号从0+1=1开始；
  - 四个TCP报文段序号：1, 101，201, 301

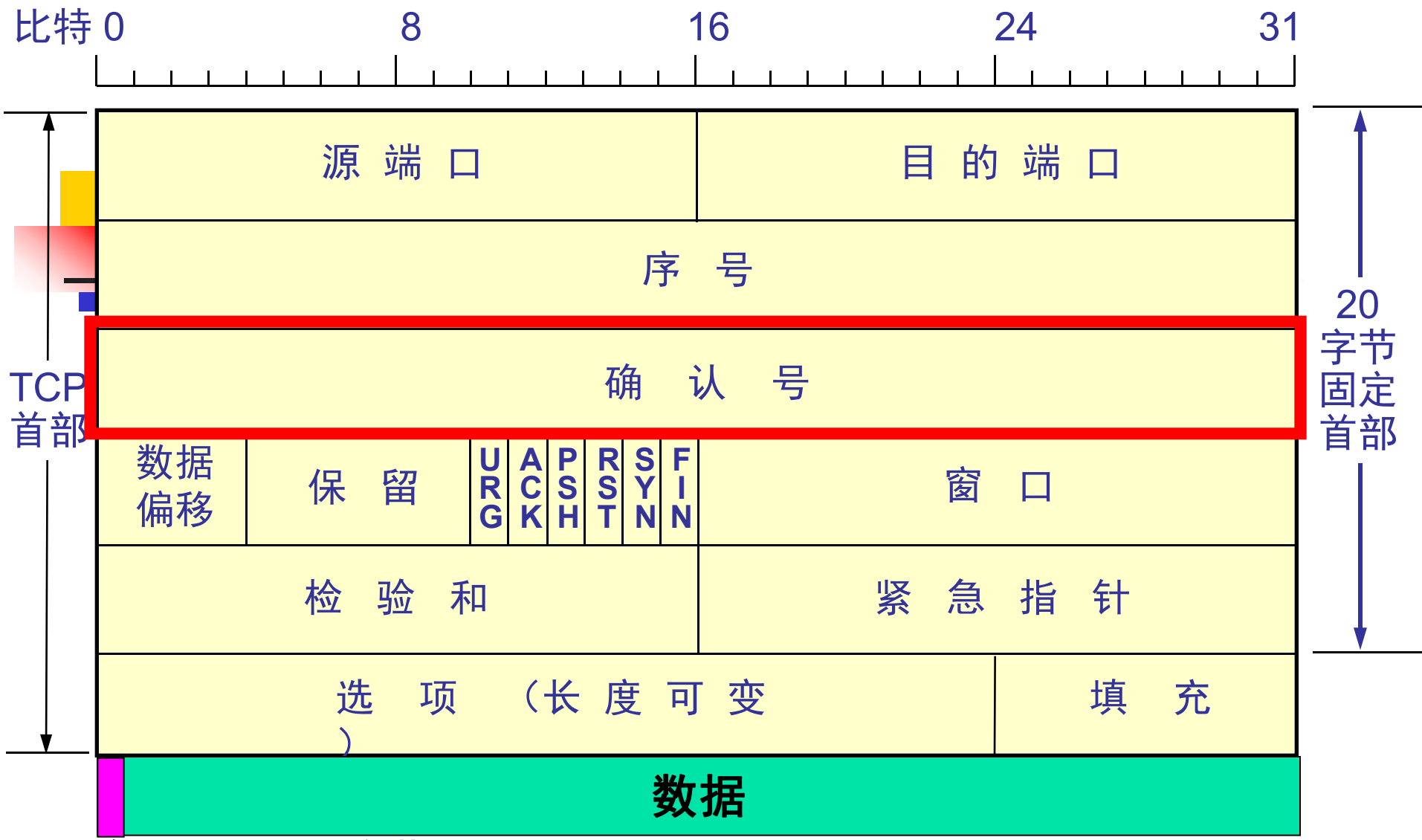
# 举例：序号（A发送TCP报文段给B）




- 如果协商A发送的初始序号为2000，数据长度100字节
  - 字节流第1个字节序号从2000+1=2001开始；
  - 四个TCP报文段序号：
    2000+1, 2000+101, 2000+201, 2000+301, ……

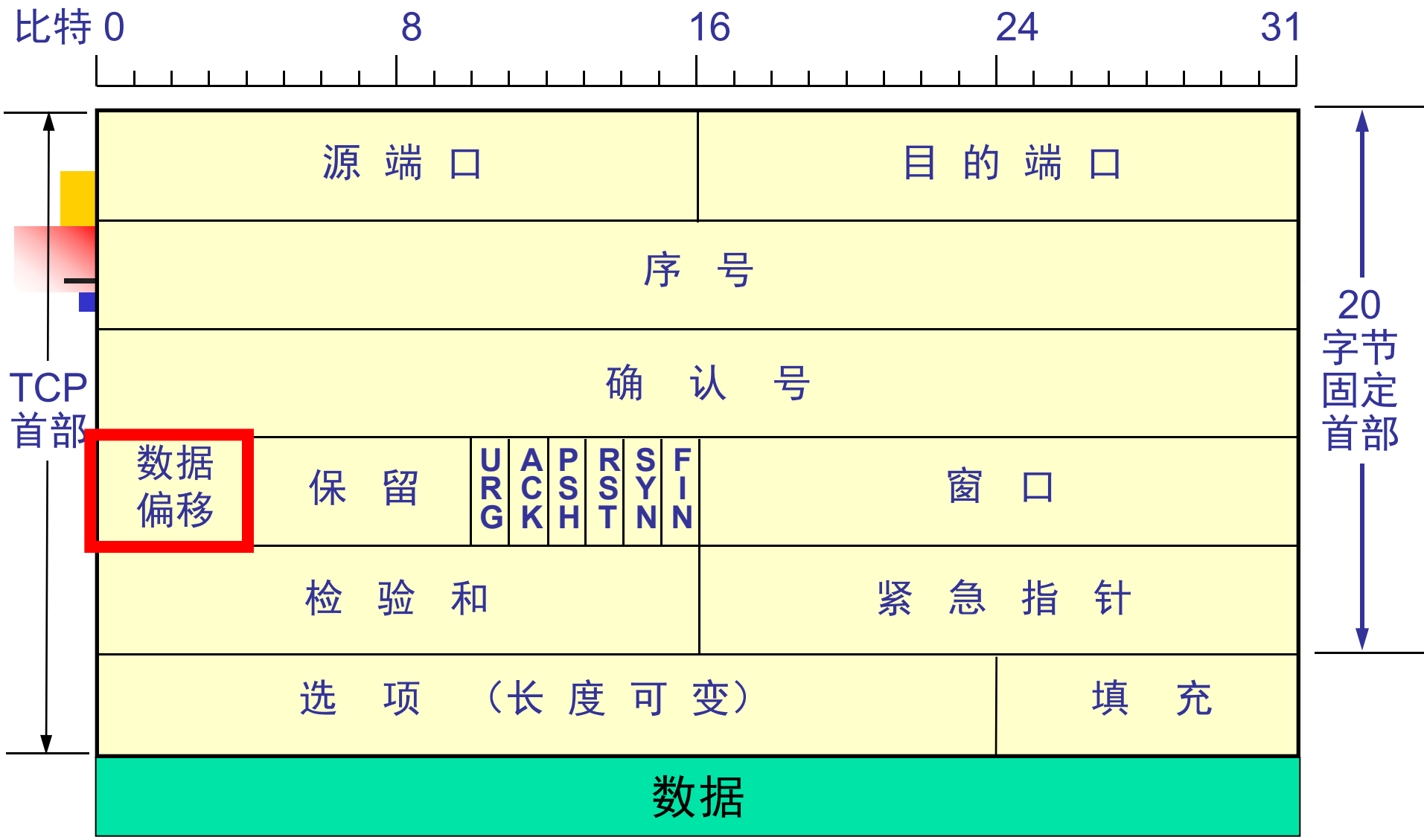



- 确认号：占 4 字节
  - 含义：接收方希望收到发送方发送的下一个TCP段数据第一个字节序号，ACK = 1情况下有效。
  - 如果确认号=N，表示N-1号以前字节已经成功接收到，期望接收到的下一个TCP段数据第一个字节序号为N；实际上等于期望接收的下一个TCP段序号字段为N



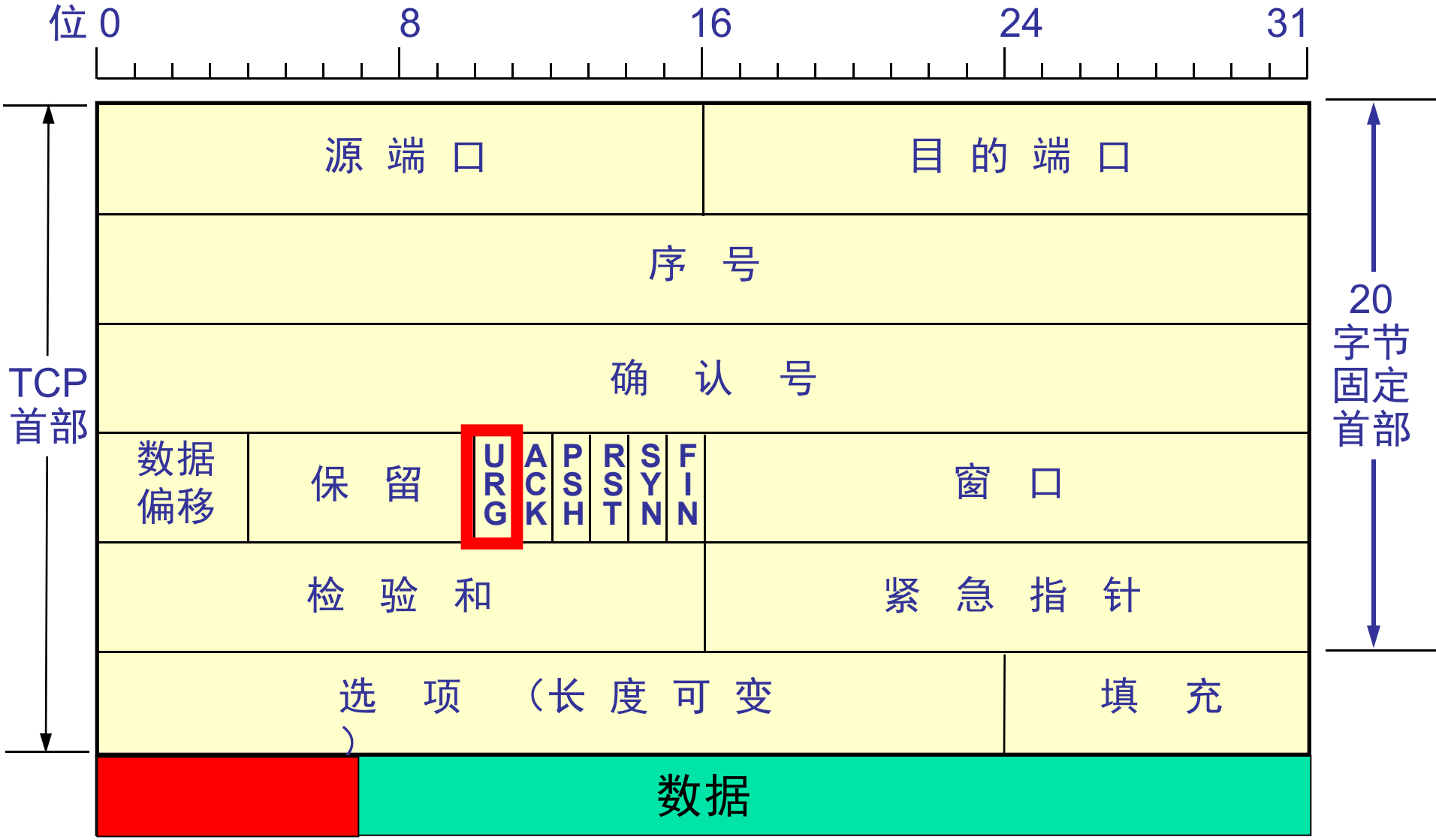

- 确认号：占 4 字节
  - 举例：如果A发送给B一个TCP报文段，其序号为501，数据长度为200个字节(该数据段序号为501~700)；则此时B希望收到A的下一个TCP段序号应该是？

    701(而不是501，700)

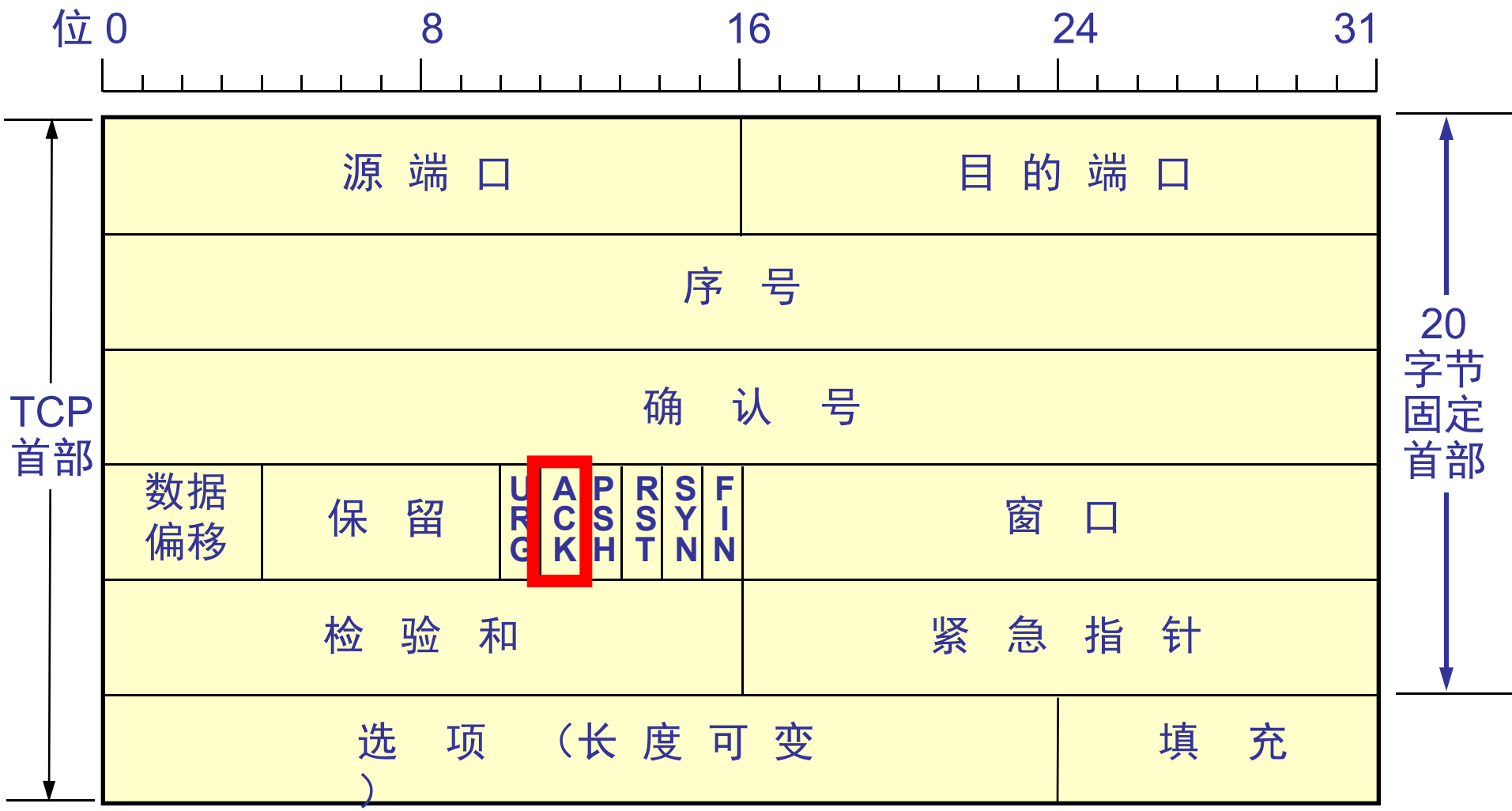



- 数据偏移（首部长度）：占 4 bit
  - 含义:TCP报文段首部长度,以4 字节为计算单位;最大值为60字节.
  - 首部长度范围：20~60个字节；选项字段范围：0~40字节。

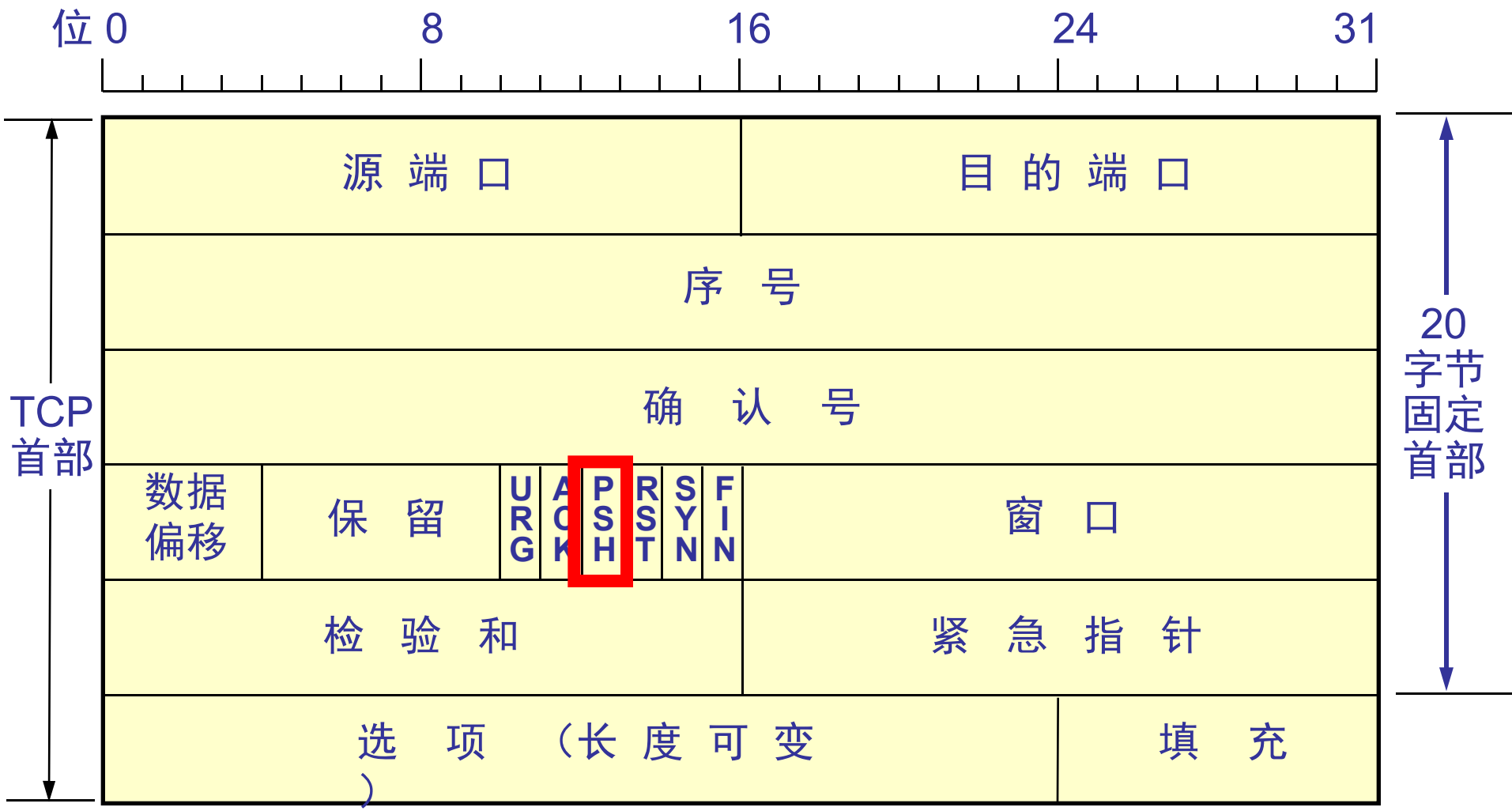

- 保留字段：占 6 bit，
  - 含义：保留为今后使用，但目前全为0。




- 6比特标志位（Frags）:UAPRSF
- 紧急比特 URG ：当 URG = 1 时，表明紧急指针字段有效，报文段中有紧急数据，应尽快传送(相当于高优先级的数据)，紧急数据在数据字段的位置由紧急指针（urgent pointer）字段给出。




- 确认比特 ACK
  - 当 ACK = 1 时确认号字段才有效;当 ACK = 0 时，确认号无效。

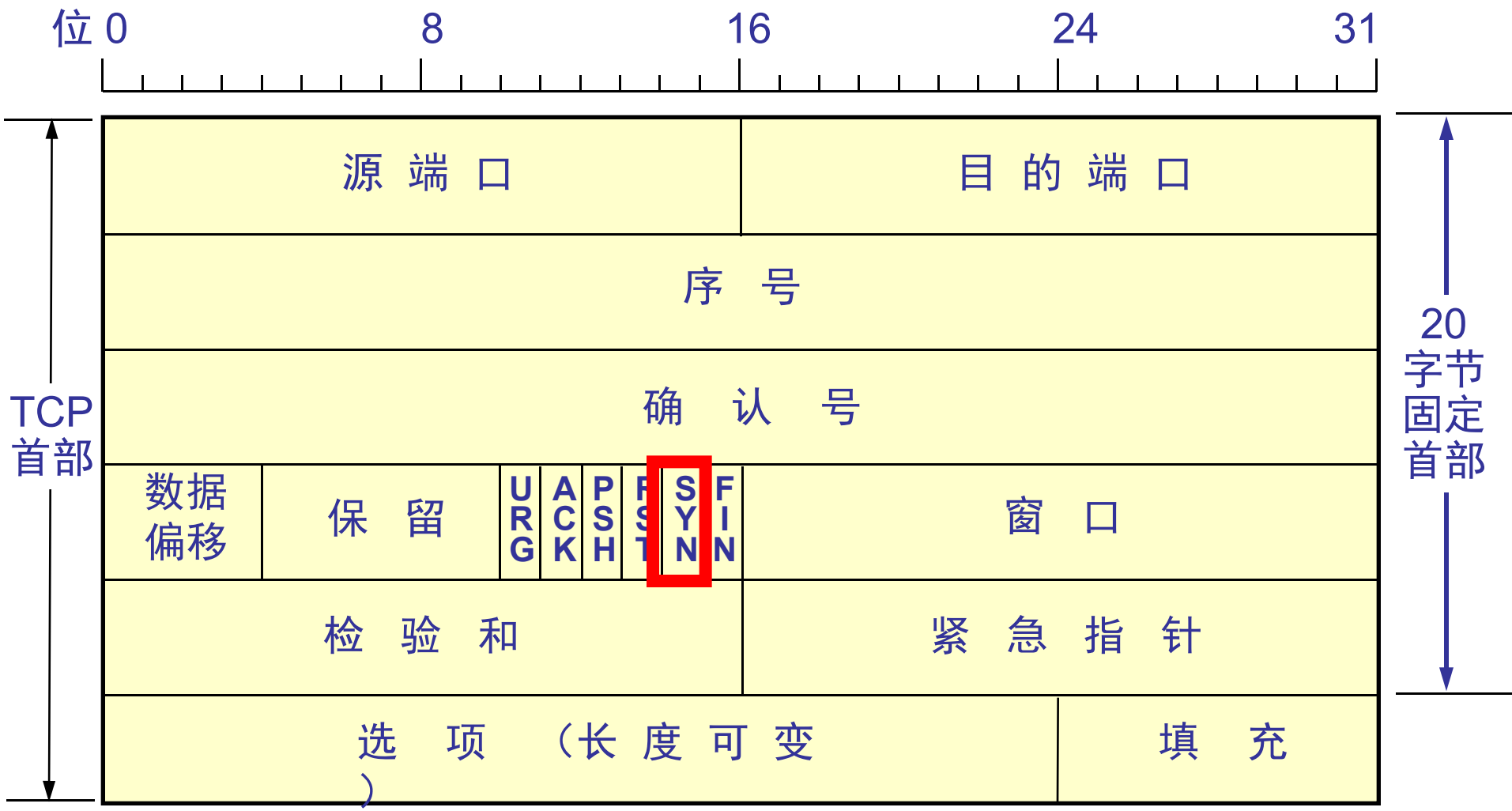



- 推送比特 PSH （PUSH操作）
  - 发送方Push=1，发送缓冲区即使有发送窗口限制，也要立即发送；
  - 接收方TCP 收到Push=1的数据不需要在接收缓冲区中排队，可尽快地交付给应用进程。

# PUSH操作

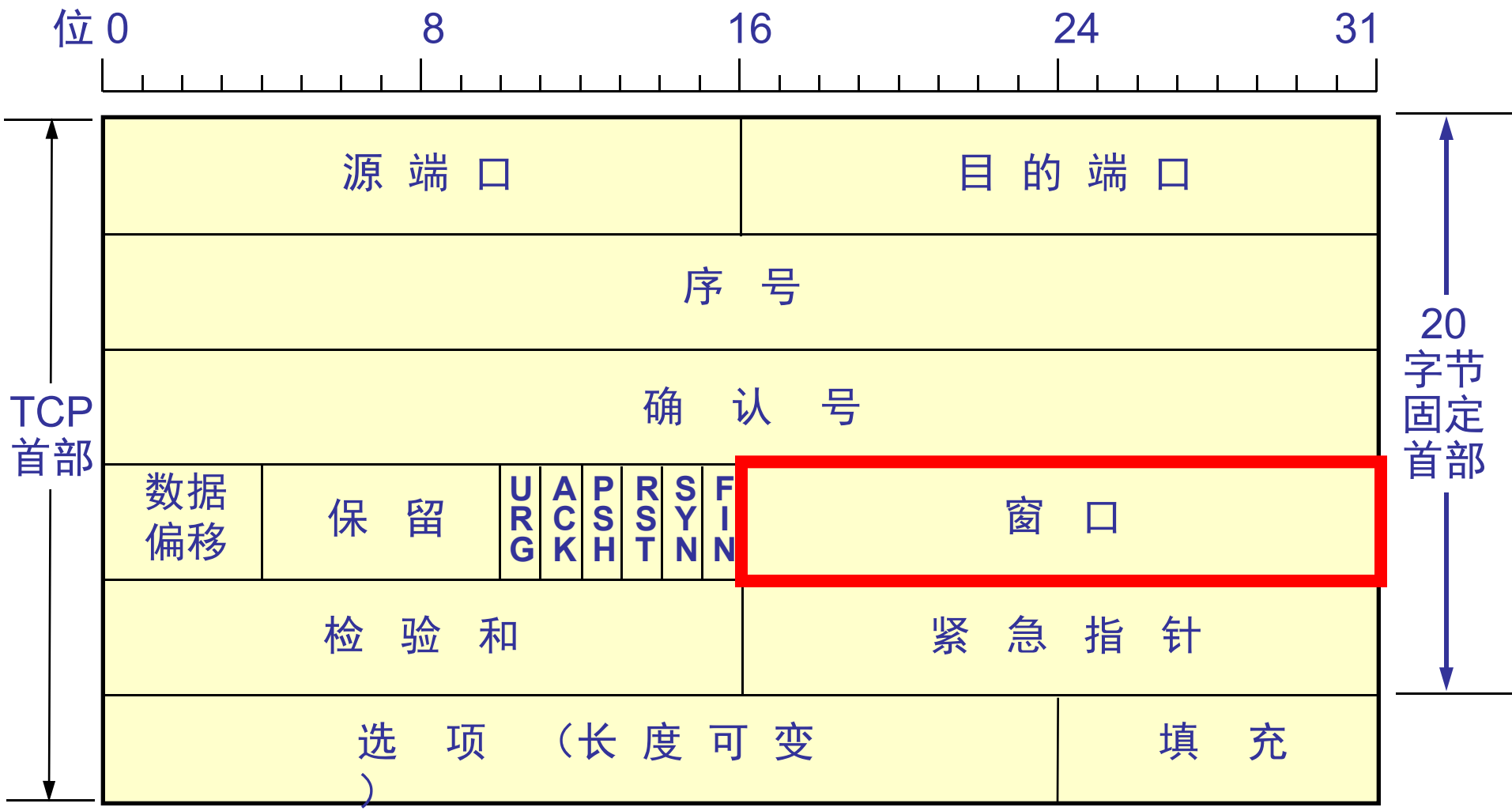
- 问题：
  - TCP连接中，发送方接收到上层数据后，先缓存到出队列，然后根据发送窗口大小分段发送上层数据。
  - 应用层在数据发送完成前，无法知道自身递交的数据是否已被发送（如下班前突然关机）。
- 解决方法：
  - 如果发送方TCP接收到上层PUSH操作，可使TCP将出队列所有数据迅速地分段发送出去，不受发送方当前发送窗口大小限制。
  - 接收方TCP接收到带PSH标志的TCP数据段后，将迅速把这些数据段快速递交给上层协议，而不用在接收缓存中排队。


- 复位比特 RST (ReSeT)
  - 当 RST = 1 时，表明 TCP 连接中出现严重差错，对错误连接复位，重新回到初始状态。



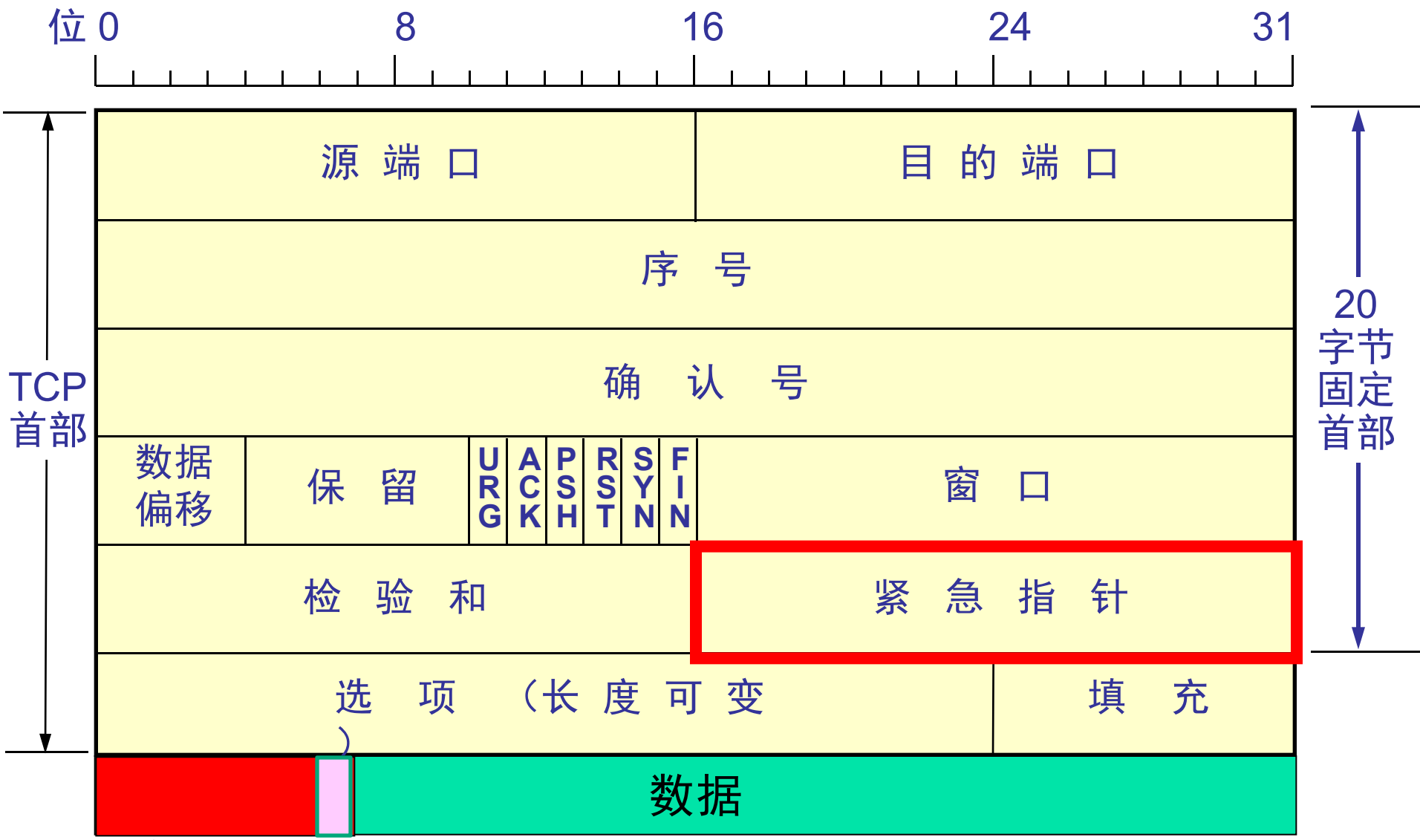

- 同步比特 SYN
  - SYN 置为 1，就表示这是一个TCP连接请求。


- 终止比特 FIN（FINal）
  - FIN = 1 时，表明发送方的数据已发送完毕，要求释放TCP连接，该TCP报文段是一个释放连接请求。




- 窗口：占 2 字节。
  - 含义：接收方通知发送方接收缓存的有效大小，单位为字节。
  - 目的：发送方接收到对方“窗口信息”后，通过调整发送方发送窗口大小，控制发送速率，实现端到端流量控制。


- 校验和：占 2 字节。
  - 检验和范围： TCP伪首部（ 12 字节）+ TCP首部 + 数据。
  - TCP伪首部：源IP地址+目的IP地址+1字节保留（全0）+协议号（TCP=6）+TCP首部长度字段，共为12个字节。
  - 方法：简单校验和，与IP协议（UDP协议）计算校验和方法相同。

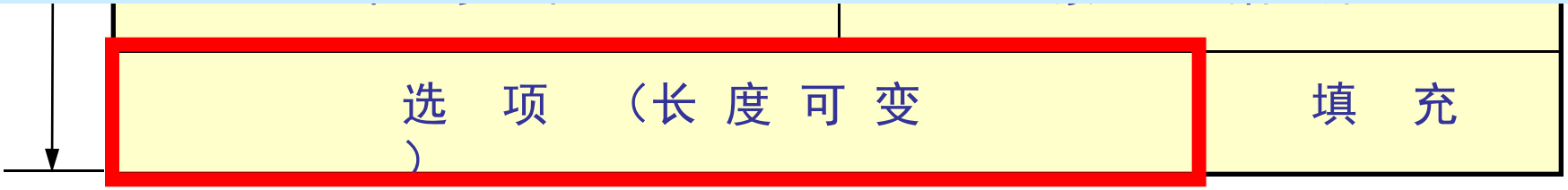



- 紧急指针： 占 16 bit;
  - 含义：TCP报文段中紧急数据字节数，URG=1时有效。
  - 紧急数据一般放在TCP报文段数据最前面。
  - 序号+紧急指针字段的值-1：本TCP段中的紧急数据最后一个字节序号。

# 紧急数据操作

- 任何一个TCP段都可以携带紧急数据(Urgent Data)，以支持上层协议紧急数据的快速传递与处理。
- 紧急数据有关TCP段相关字段:
  - URG标志位:当前数据段中携带有紧急数据。
  - 紧急指针:TCP报文段中紧急数据字节数。
- 举例
  - 假设发送方正在发送数据,然后等待接收方处理结果返回;
  - 发送方突然发现有错误,希望将此过程异常终止；如果此时发送终止命令(Control+C),这两个字符必然排在接收方队列尾,等所有数据处理完,才将这两个字符交上层.
- 解决方法
  - 发送方将URG比特置1,告诉TCP该数据为紧急数据,发送端TCP将该含有紧急数据数据块就安排在发送队列队首，以便尽快发送出去;
  - 接收方TCP收到URG=1的报文段后,利用紧急指针的值将该段中紧急数据提取出来,并不按缓冲排队顺序，提前交给应用进程处理，避免对错误数据处理。

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。



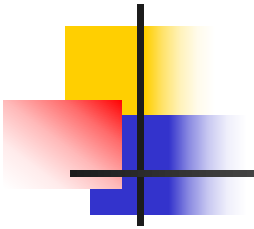

选项字段 —— 长度可变（0~40B）。TCP 最初只规定了一种选项，即最大报文段长度 MSS；MSS 告诉对方 TCP：“我的缓存所能接收的报文段的数据字段的最大长度是 MSS 个字节。”

# MSS选择

- 如果太小,数据传输效率低。
  - 假设传送一个字节数据,在网络层数据传输效率率1/41(不包括链路层开销).
- 如果太大,网络处理时间长，开销大。
  - 在网络层一个大的TCP报文段又要分片,构成若干个小IP分组;
  - 接收方对小的分组重新组装;
  - 如果有一个小的IP分组出现差错，丢失，超时，整个TCP报文段需要重传;
- 选择原则
  - 一般认为MSS尽可能大,只要在IP层不分片就可以了。
- 具体实现
  - 建立连接时,双方都要将自己可支持的MSS写入可选字段进行协商；通信时两个方向的MSS可以不同。
  - 如果主机没填写选项MSS字段，目前在Internet,MSS一般采用默认值为536字节,传输层默认TCP报文段长为:536+20（固定首部）=556个字节。
- 注意：由于IP分组是独立路由，不同分组每次选择的路径可能不同，在某一条路径上按确定的MSS不需要分片，但改选另一条路径可能需要分片，所以最佳的MSS很难确定。


- 填充字段
  - 为了使整个TCP首部长度是 4 字节的整数倍，按照32比特位对齐。

Sniffer Tool to capture TCP SEGMENT: www.nwpu.edu.cn(HTTP)



# 本节内容

6.1 传输层协议概述
- 6.1.1 进程之间的通信
- 6.1.2 传输层的两个主要协议
- 6.1.3 传输层的端口

6.2 用户数据报协议 UDP
- 6.2.1 UDP 概述
- 6.2.2 UDP 的首部格式

6.3 传输控制协议 TCP 概述

6.4 TCP 报文段的首部格式

# 下节预习内容

6.5 可靠传输的工作原理

6.5.1 停止等待协议

6.5.2 连续 ARQ 协议

6.6 TCP 可靠传输的实现

6.6.1 以字节为单位的滑动窗口

6.6.2 超时重传时间的选择

6.6.3 选择确认 SACK

6.7 TCP的流量控制

6.7.1 利用滑动窗口实现流量控制

6.7.1 必须考虑传输效率

# 传输层的主要功能

- 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）。
- 传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），使应用进程看见的就是好像在两个传输层实体之间有一条端到端的逻辑通信信道。
- 传输层对收到的报文段（UDP用户数据报）进行差错检测。
- 传输层需要有两种不同的协议，即面向连接的 TCP 和无连接的 UDP。

# 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比特，S最大为14)，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：时间戳字段（4 字节）和时间戳回送应答字段（4 字节）。
  - 计算往返时间RTT：发送方发送TCP段时，把发送时间写在时间戳字段；接收方发送其确认应答报文段时，将时间戳字段值复制到时间戳回送应答字段，发送方可计算RTT；
  - 防止序号绕回：一次TCP连接中，序号（32比特）很可能重复，如果发送速率为1Gb/s,不到35秒序号就会重复，为了使接收方能将新报文段与延迟报文段分开，可利用时间戳区分。
- 选择确认选项——在后面的 6.6.3 节介绍。