



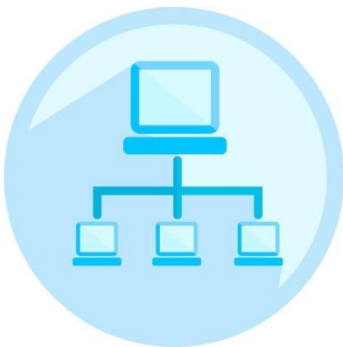
# 计算机网络



顾 军

计算机学院

[jgu@cumt.edu.cn](mailto:jgu@cumt.edu.cn)





# 专题5：如何保证端到端的可靠传输



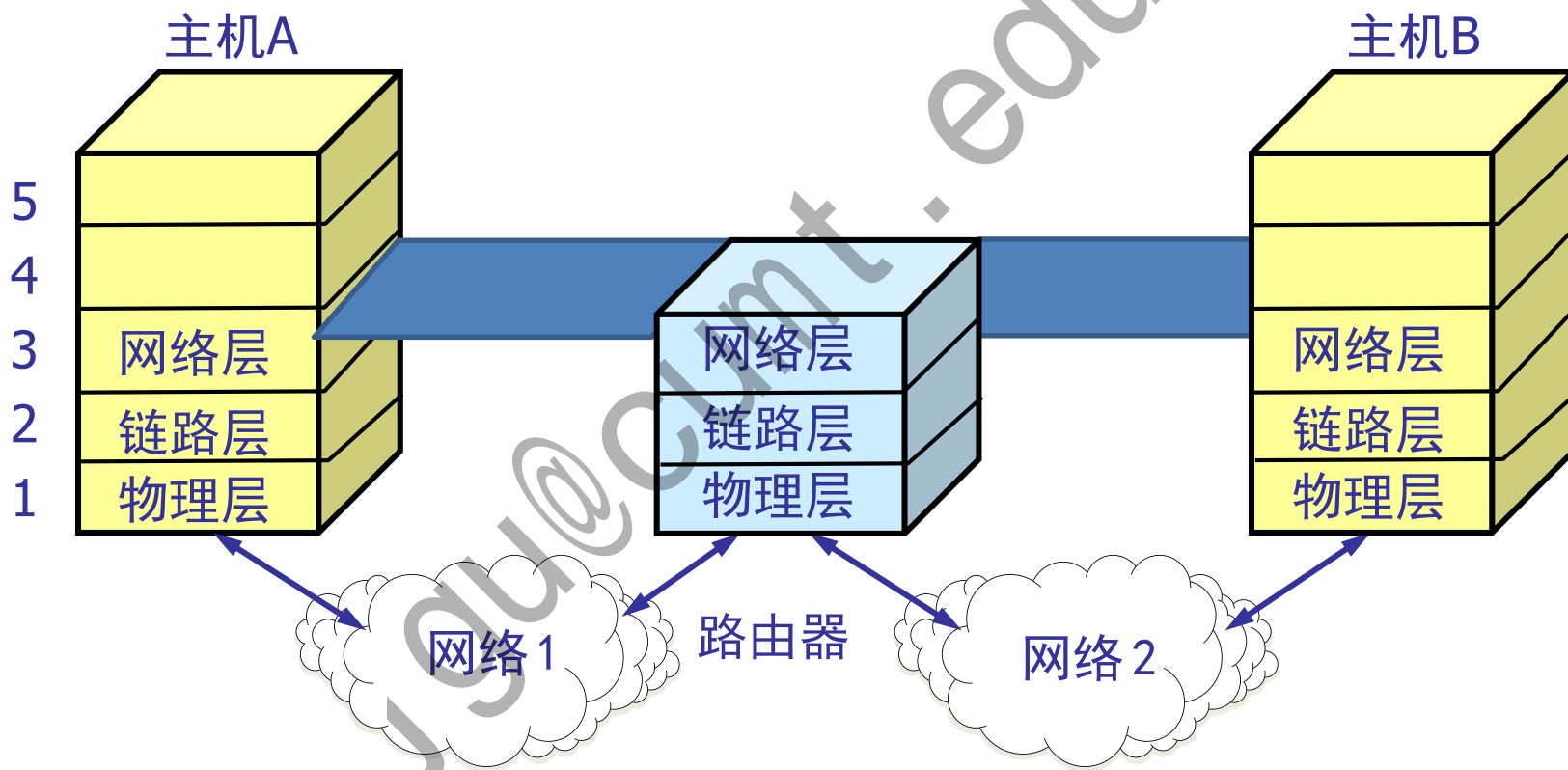
- 应用层(application layer)
- 运输层(transport layer)
- 网络层(network layer)
- 数据链路层(data link layer)
- 物理层(physical layer)





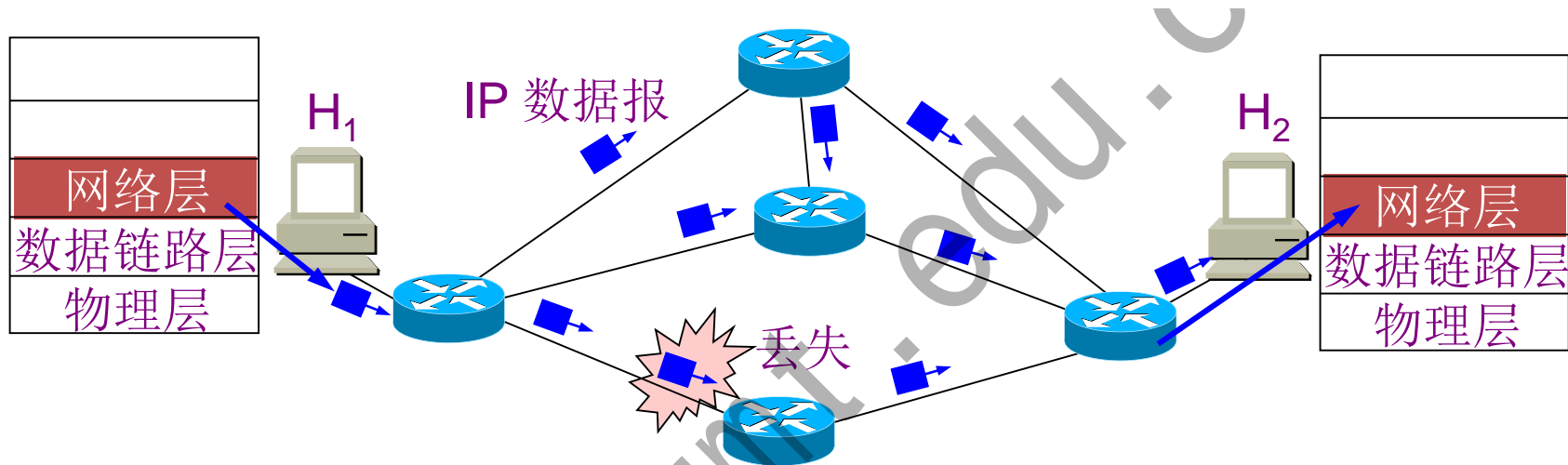
# Q1: 如何保证数据报的可靠交付?

网络层不能提供数据报的可靠交付服务





# 网络层的数据报服务



$H_1$  发送给  $H_2$  的分组可能沿着不同路径传送

网络在发送分组时不需要先建立连接。每一个分组（即 IP 数据报）独立发送，与其前后的分组无关（不进行编号）。

网络层向上只提供简单灵活的、无连接的、尽最大努力交付（Best Effort）的数据报服务。





## 网络层不可靠的数据报服务 与互联网应用的可靠交付需求 存在矛盾

- 在计算机网络中，可靠交付应当由谁来负责？

一是网络还是端系统？

### 过程控制

关注数据报转发的每一个步骤，需要对每一个网络交换设备进行管理。

### 结果驱动

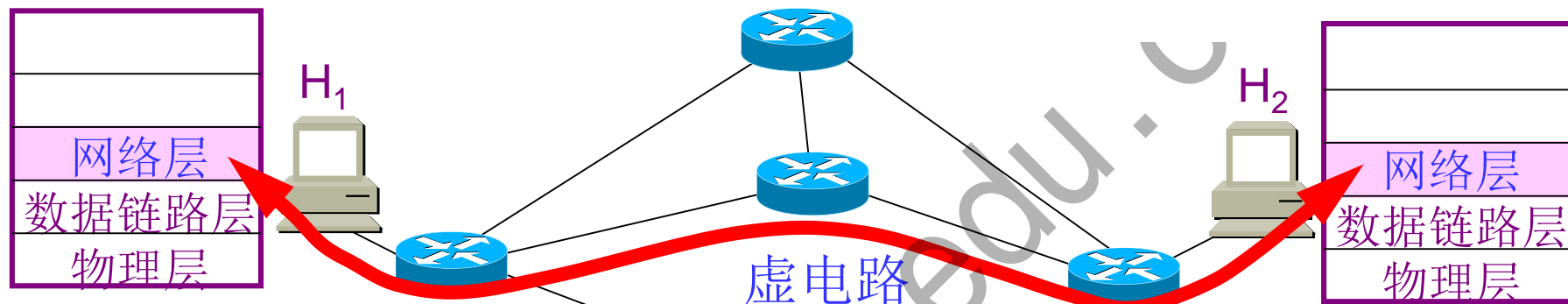
不关注数据报转发过程，只关注端设备的状态。





# 电信网的成功经验

## 让网络负责可靠交付



$H_1$  发送给  $H_2$  的所有分组都沿着同一条虚电路传送

- 电信网采用面向连接的通信方式，通过建立虚电路(Virtual Circuit)，以保证双方通信所需的一切网络资源。
- 虚电路表示这只是一条逻辑上的连接，分组都沿着这条逻辑连接按照存储转发方式传送，而并不是真正建立了一条物理连接。





# 电信网可以保证端到端可靠性

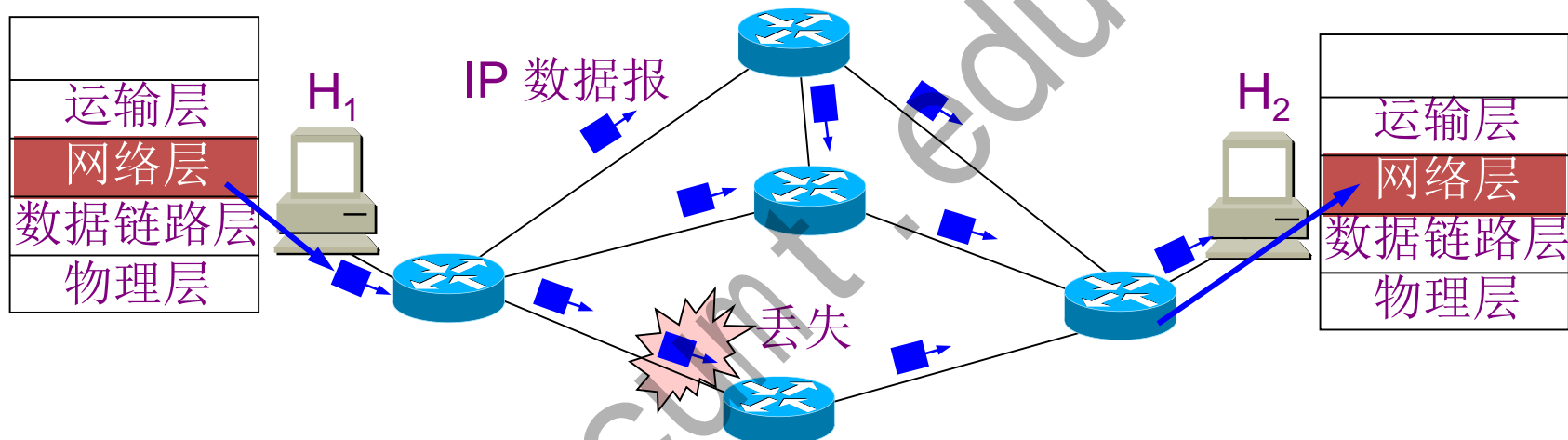
- 如果再使用可靠传输的网络协议，就可使所发送的分组无差错按序到达终点。
- 请注意，电路交换的电话通信是先建立了一条真正的连接。因此分组交换的虚连接和电路交换的连接只是类似，但并不完全一样。
- 但是虚电路技术需要加强对网络转发设备的控制，使得网络变得更复杂。





# 互联网由端系统负责可靠交付

## 运输层的引入



- 如果主机（即端系统）中的进程之间的通信需要是可靠的，那么就由网络的主机中的运输层负责（包括差错处理、流量控制等）。







# 互联网“尽力而为”设计思路的必要性

- **网络层不提供服务质量的承诺**。即所传送的分组可能出错、丢失、重复和失序（不按序到达终点），当然也不保证分组传送的时限。
- 由于传输网络不提供端到端的可靠传输服务，这就使网络中的路由器可以做得比较简单，而且价格低廉。
- 采用这种设计思路的好处是：**网络的造价大大降低，运行方式灵活，能够适应多种应用**。
- 因特网能够发展到今日的规模，充分证明了当初采用这种设计思路的正确性。





# 虚电路服务与数据报服务的对比

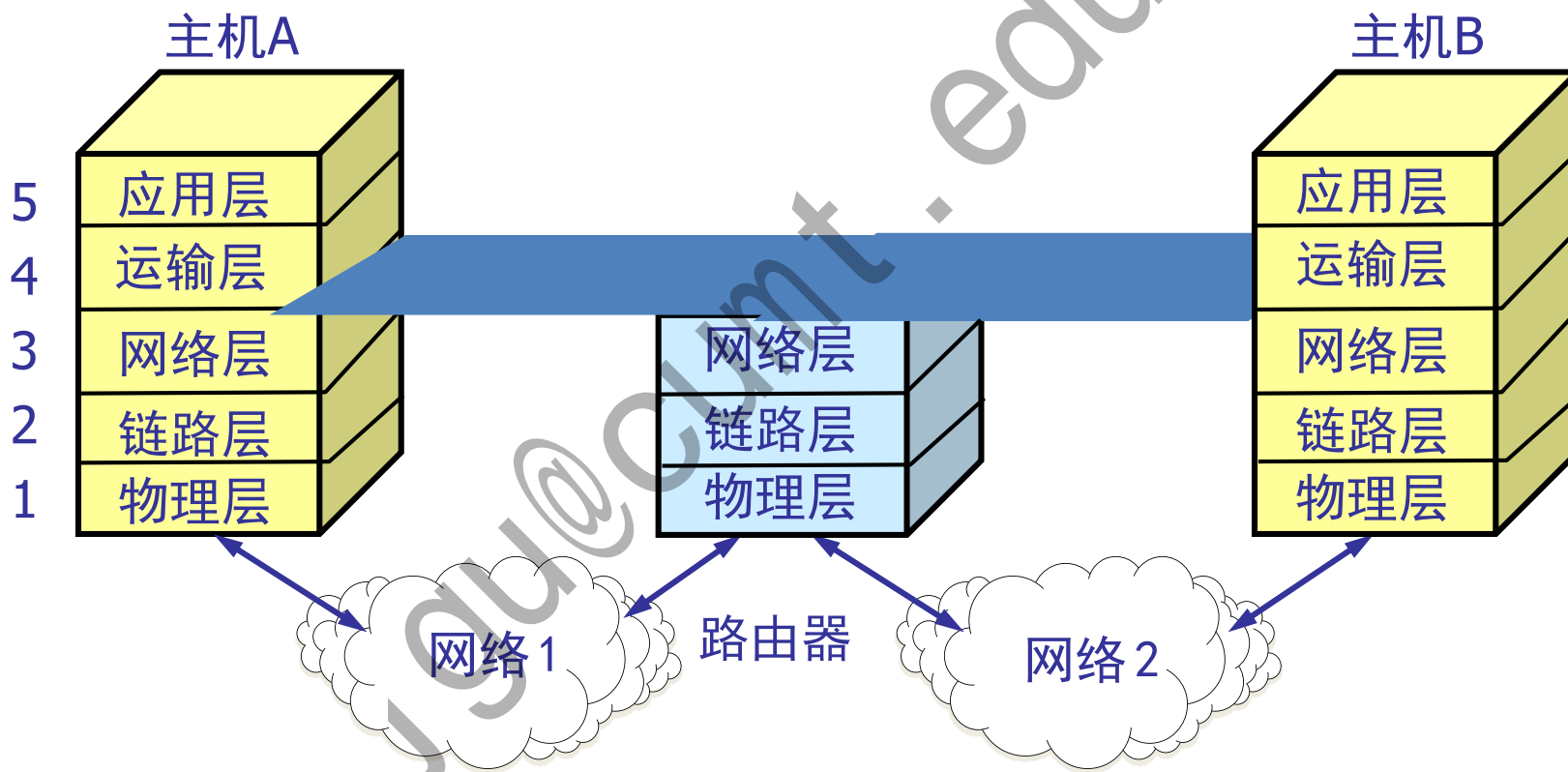
对比的方面	虚电路服务	数据报服务
思路	可靠通信应当由网络来保证	可靠通信应当由用户主机来保证
连接的建立	必须有	不需要
终点地址	仅在连接建立阶段使用，每个分组使用短的虚电路号	每个分组都有终点的完整地址
分组的转发	属于同一条虚电路的分组均按照同一路由进行转发	每个分组独立选择路由进行转发
当结点出故障时	所有通过出故障的结点的虚电路均不能工作	出故障的结点可能会丢失分组，一些路由可能会发生变化
分组的顺序	总是按发送顺序到达终点	到达终点时不一定按发送顺序
端到端的差错处理和流量控制	可以由网络负责，也可以由用户主机负责	由用户主机负责





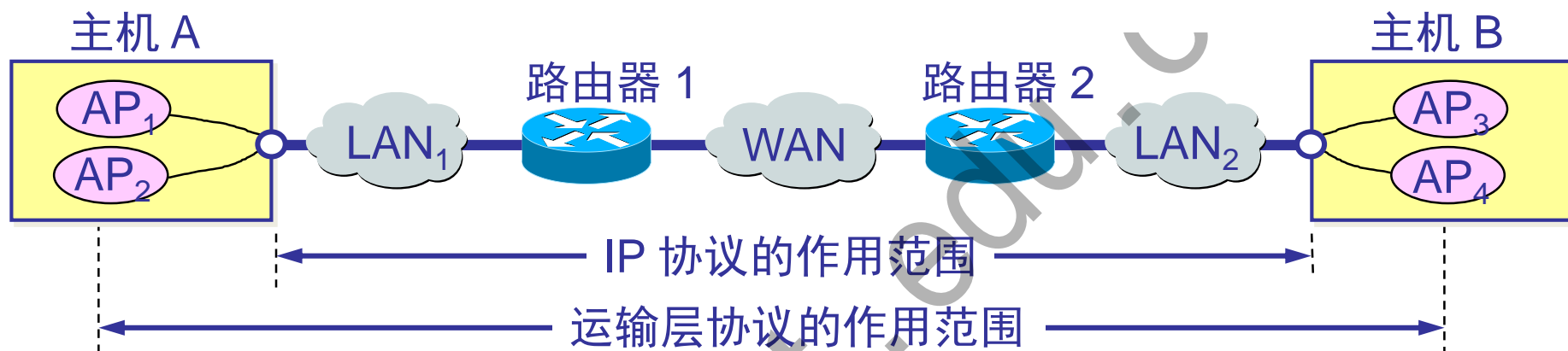
## Q2: 运输层将承担哪些功能?

功能一：运输层提供端到端控制机制，提供可靠传输保证，对收到的报文进行差错检测





## 功能二：运输层提供应用进程之间的逻辑通信



- IP 数据报要经过互联网中许多路由器的存储转发，但是这些路由器根本不知道上面的运输层是否建立了可靠连接。
- 运输层数据是在运输层的端到端抽象的逻辑信道中传送的，这样的信道也不知道究竟经过了哪些路由器。



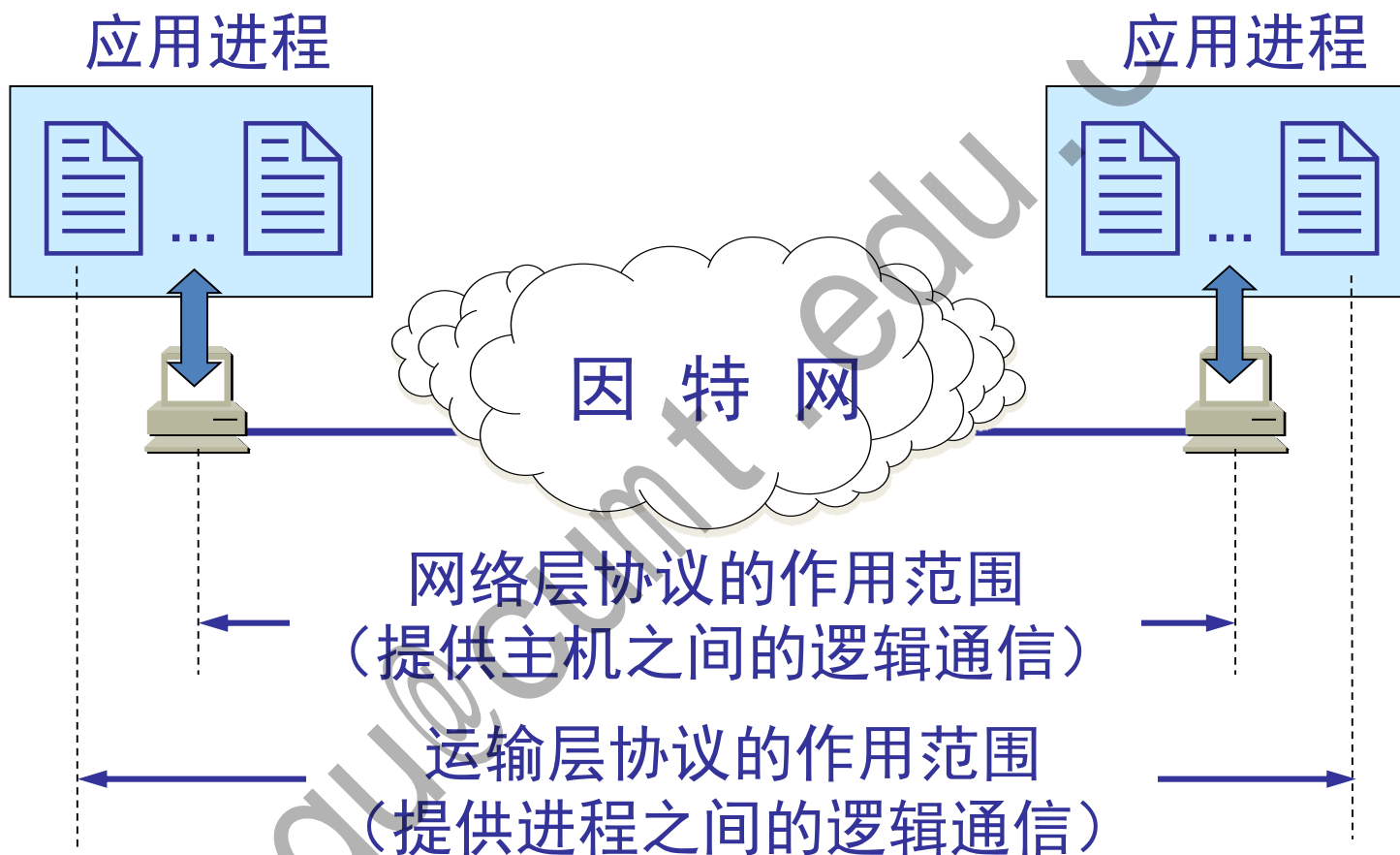


- 运输层向高层用户屏蔽了下面网络核心的细节（如网络拓扑、所采用的路由选择协议等），它使应用进程看见的就是好像在两个运输层实体之间有一条端到端的逻辑通信信道。





# 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**。





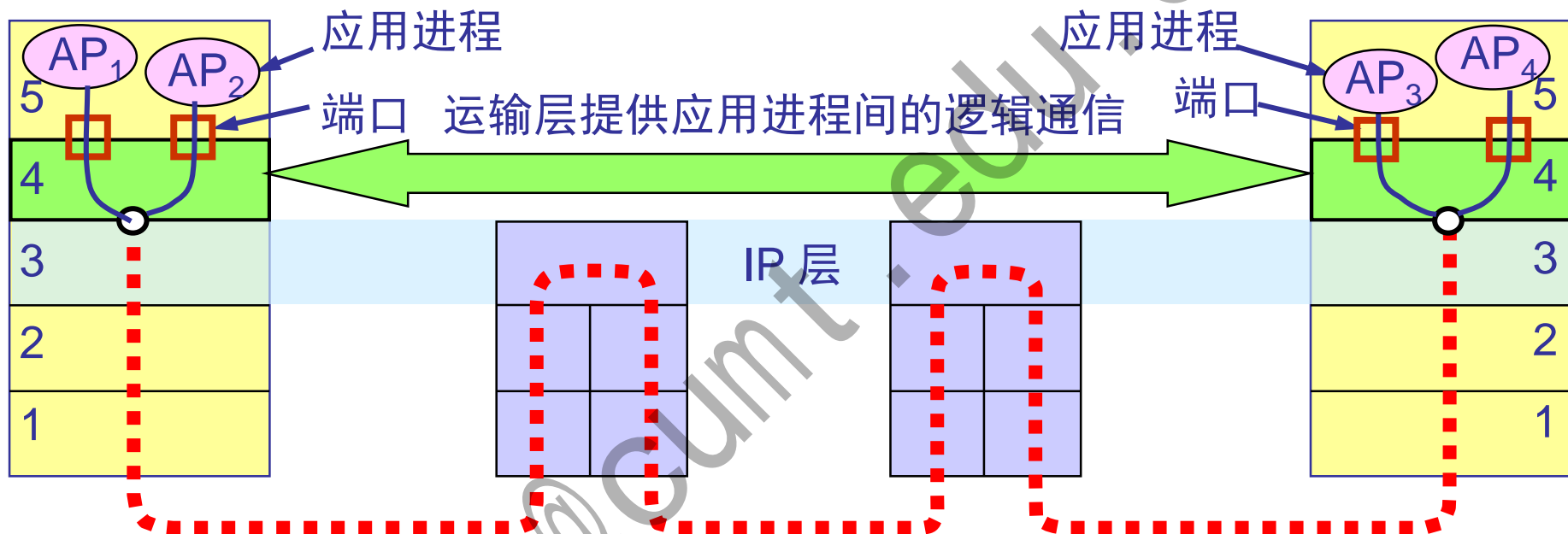
### 功能三：让应用进程在网络中可识别

- 运行在计算机中的进程是用**进程标识符**来标志的，但是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。
- 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**。





# 基于端口的应用进程间逻辑通信



“**逻辑通信**”：运输层之间的通信**好像**是沿水平方向传送数据。但事实上它们之间并没有一条水平方向的物理连接。

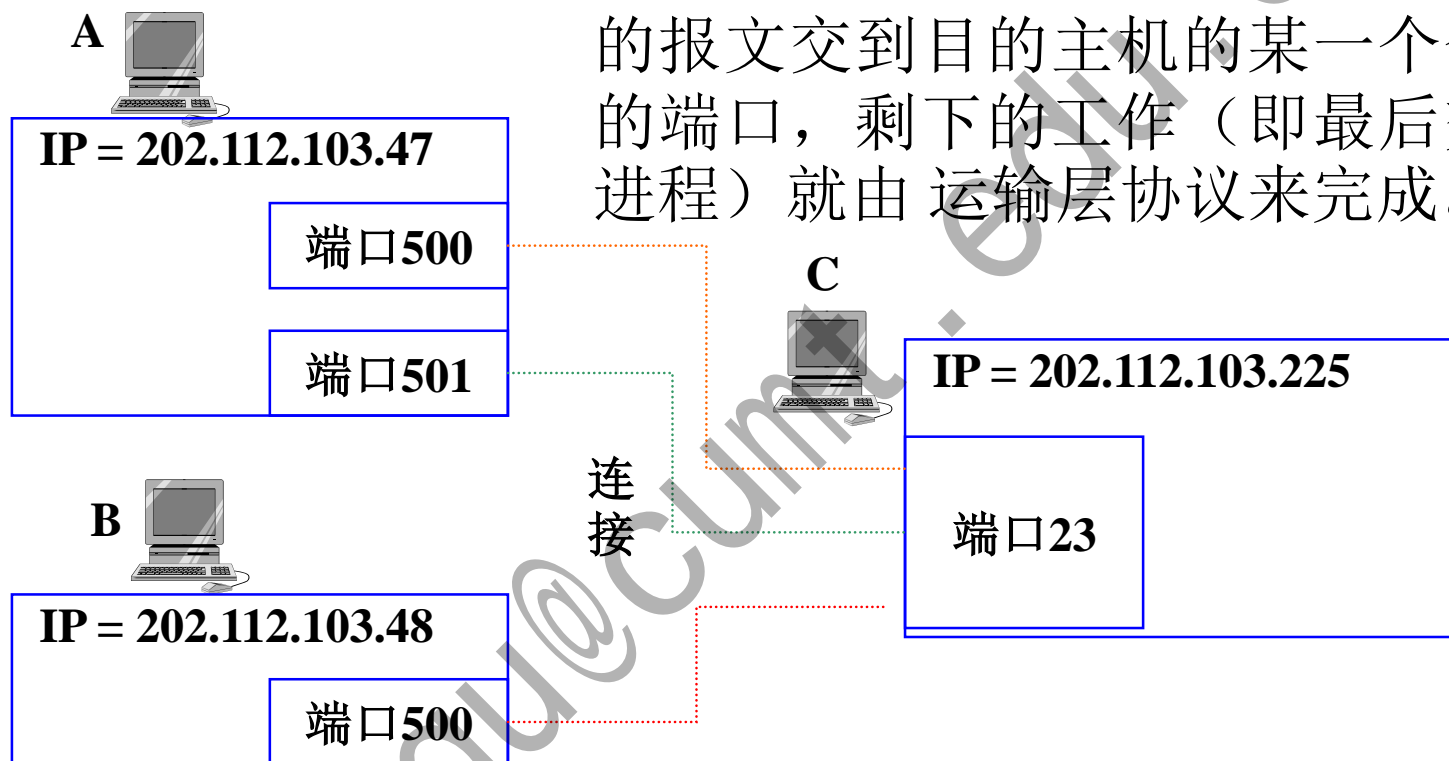






## Q3: 运输层怎么标识应用进程?

- 虽然通信的终点是应用进程，但也可把端口看作是通信的终点，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）就由运输层协议来完成。



- 端口用一个 **16 位** 端口号进行标志，而且只具有**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。





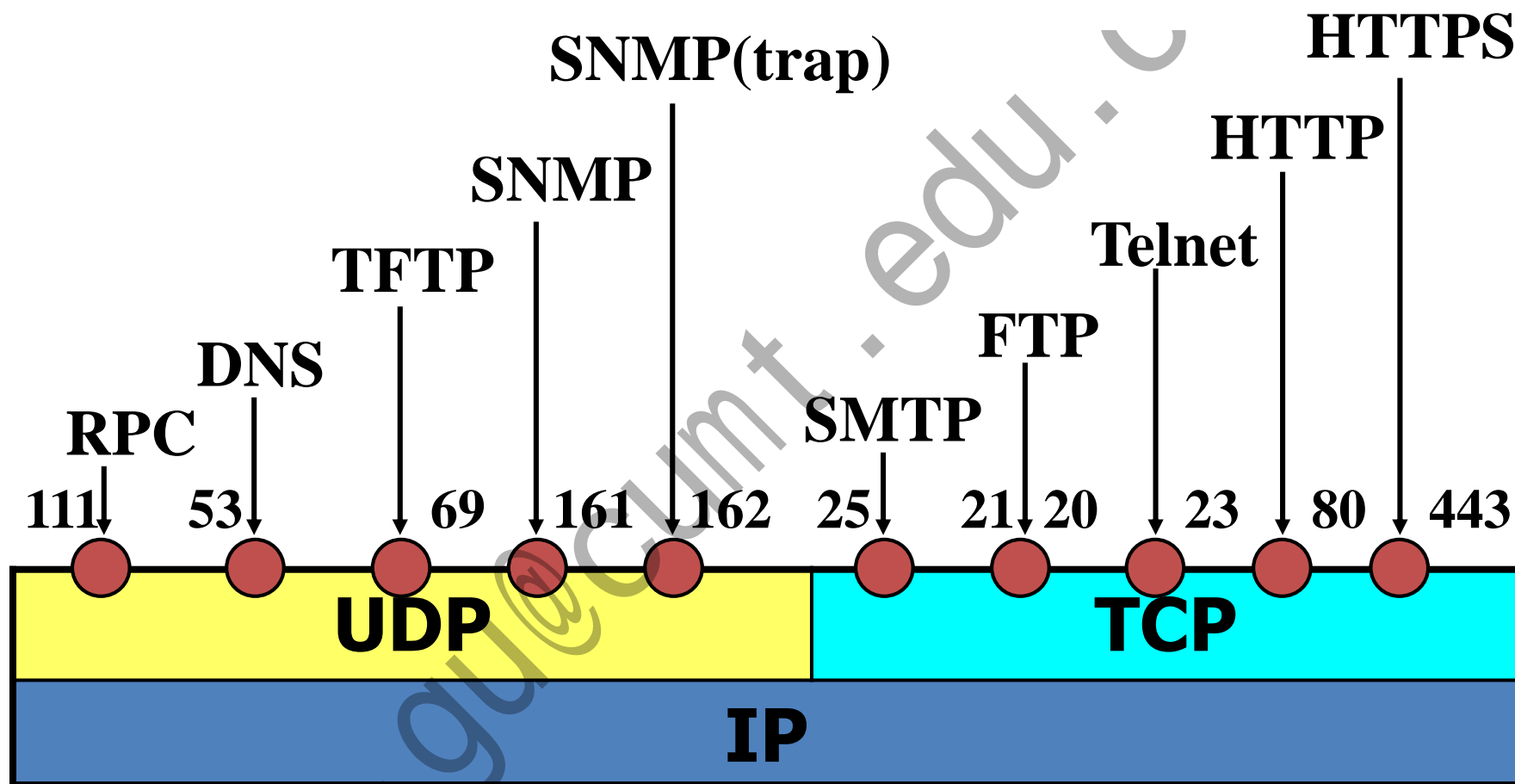
# 端口号的分类

- 服务器使用的端口号
  - 熟知端口(well-know port), 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。
- 客户端使用的端口号
  - 又叫短暂端口号(ephemeral port), 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。



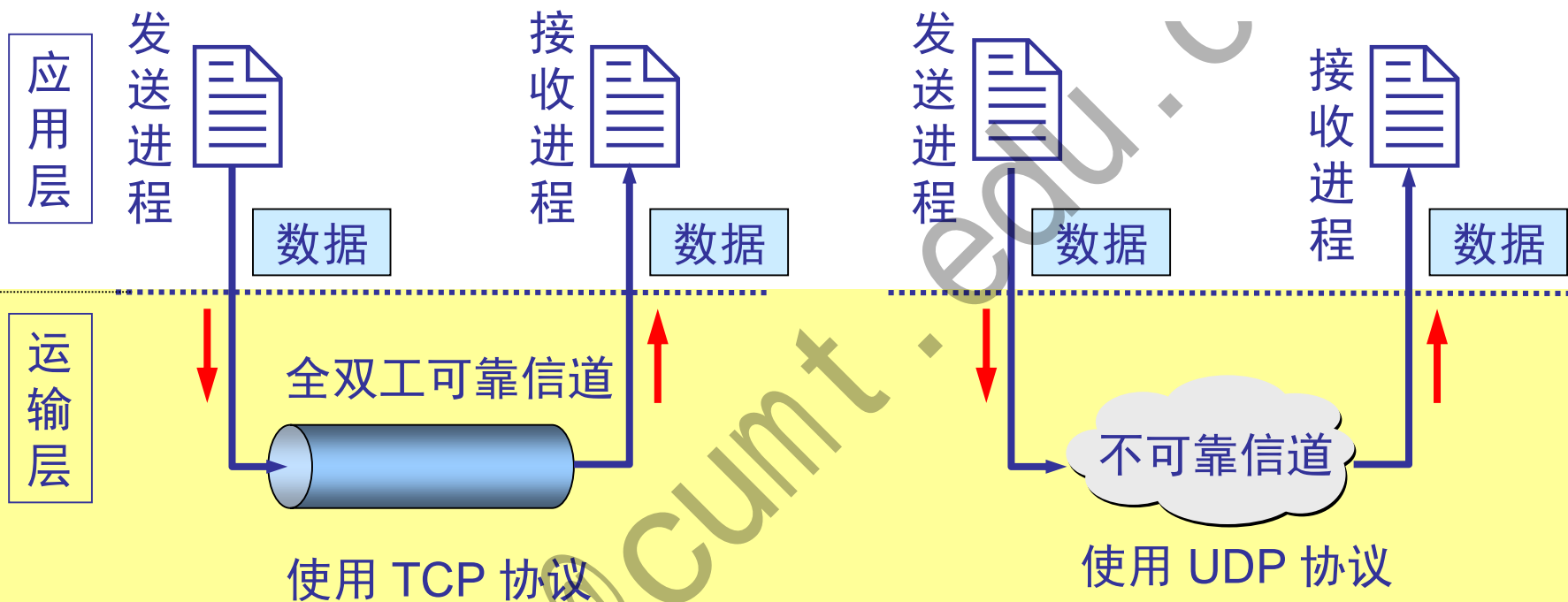


# 常用的熟知端口 (0~1023)





## Q4: 运输层逻辑信道的分类?



- 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**。

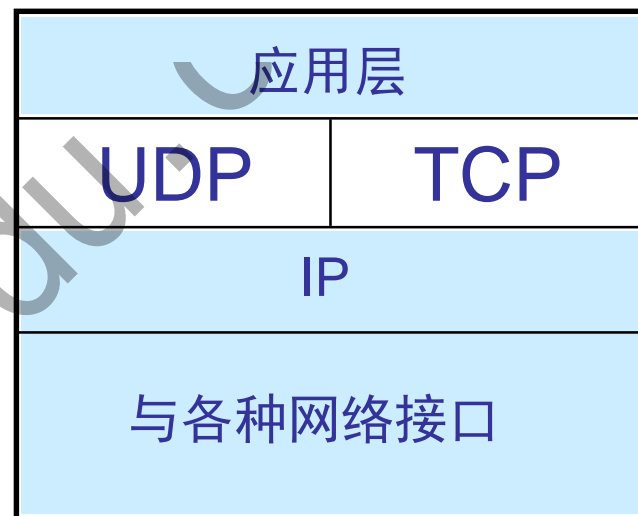
- 当运输层采用无连接的 UDP 协议时，这种逻辑通信信道是一条**不可靠信道**。





# 运输层协议

- 运输层需要有两种不同的运输协议，即面向连接的传输控制协议 TCP(Transmission Control Protocol)和无连接的用户数据报协议 UDP(User Datagram 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 传送的数据单位协议是 **TCP 报文段**(segment)
  - UDP 传送的数据单位协议是 **UDP 报文**或**用户数据报**





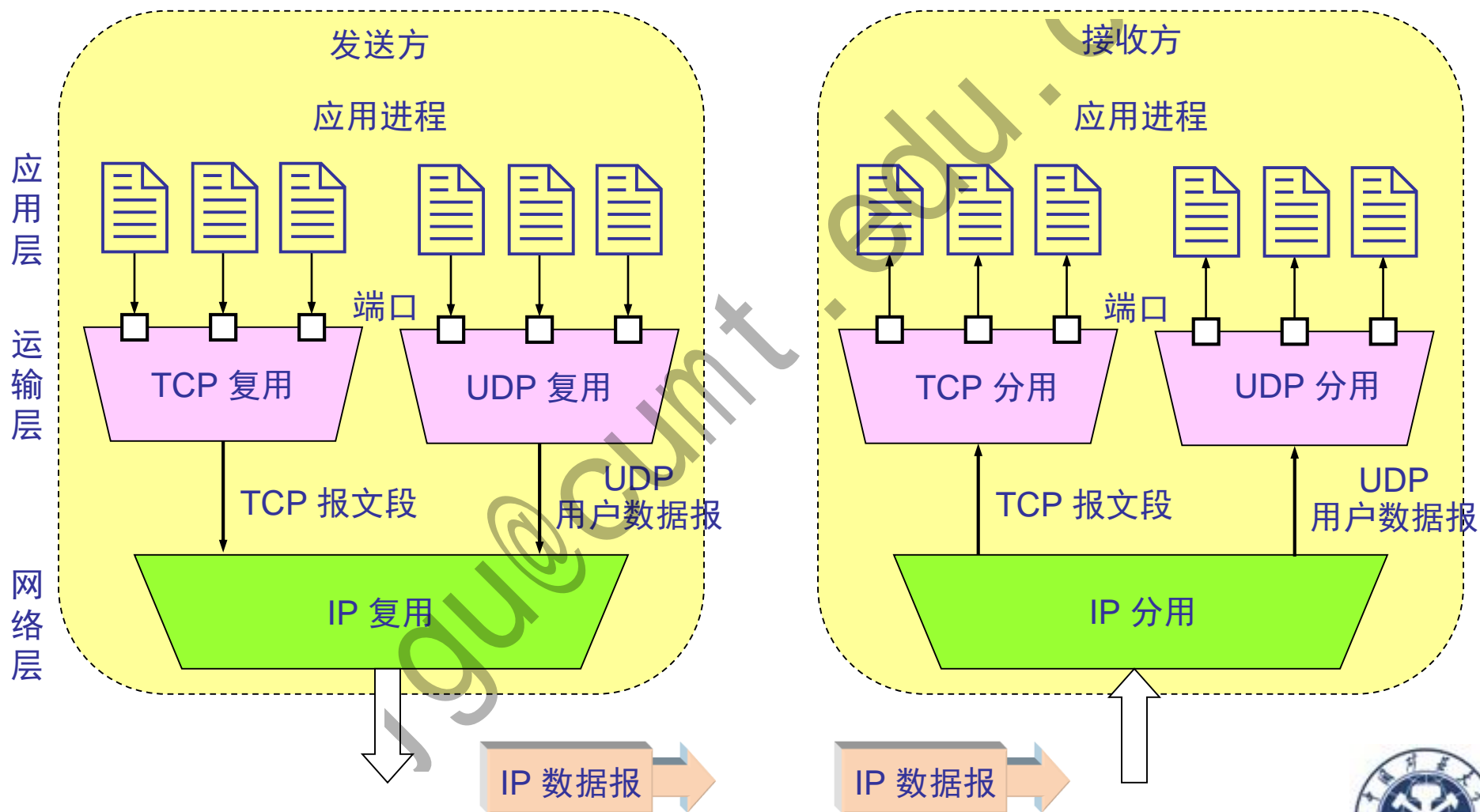
# UDP协议与TCP协议的区别

- 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，如视频应用。
- 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。
- 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**。
  - 允许通信双方的应用进程在任何时候都能发送数据。





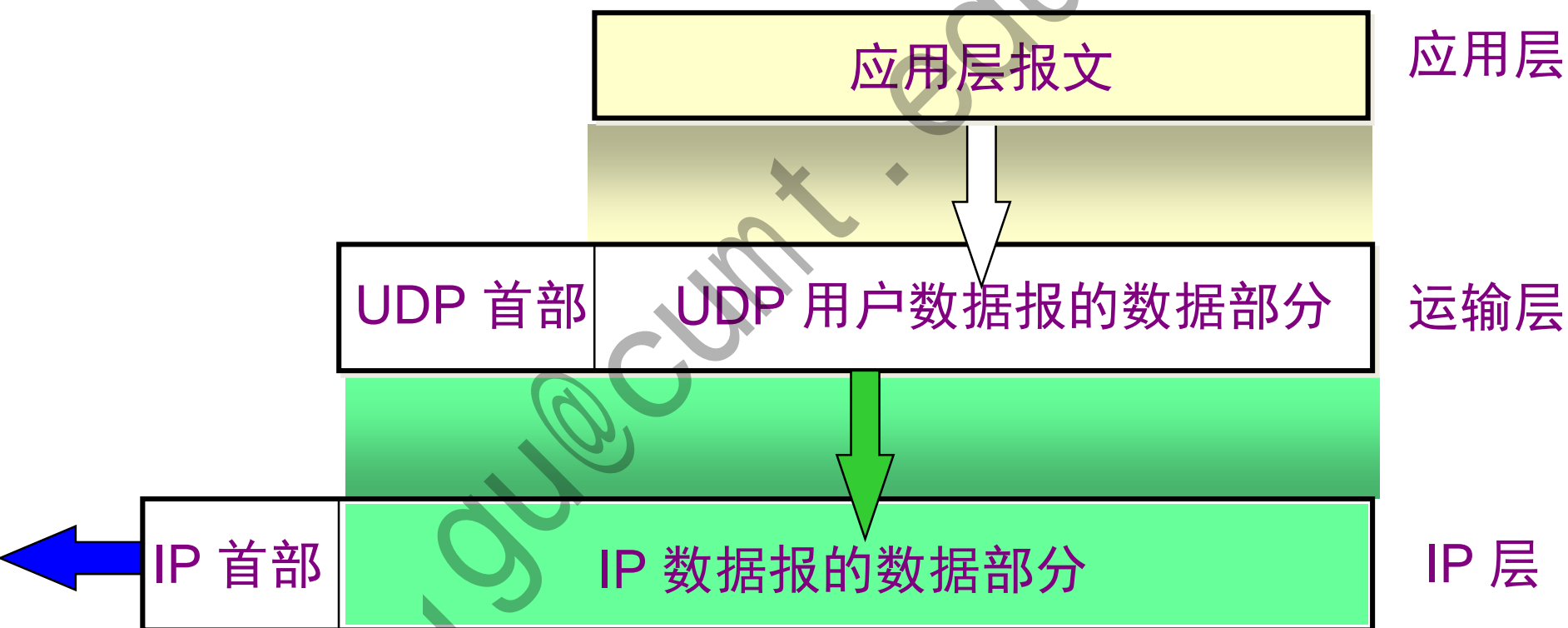
# 端口在进程之间的通信中所起的作用





## Q5: 用户数据报协议 UDP ?

- 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。



UDP 是面向报文的

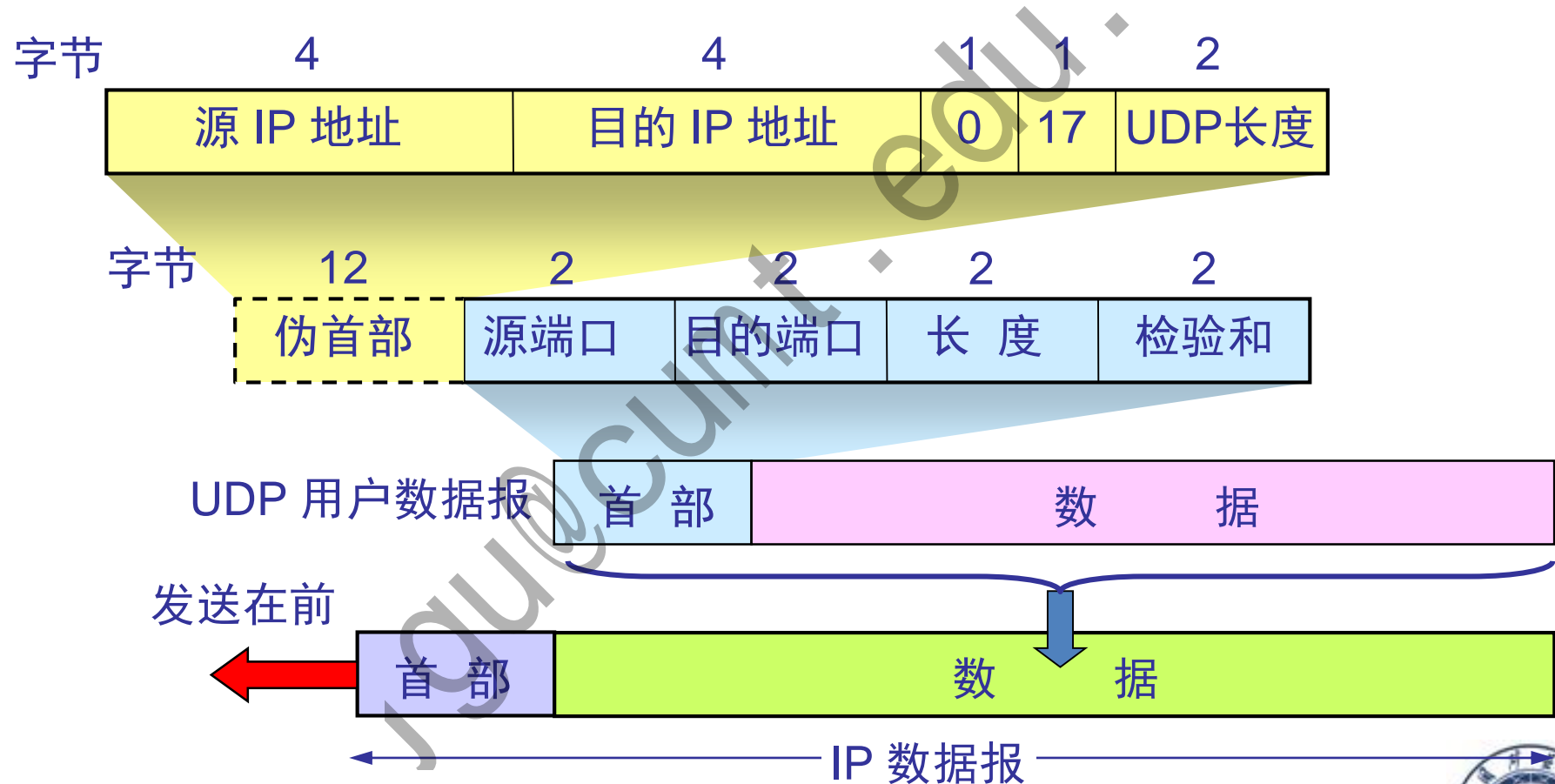






# UDP 的首部格式

用户数据报 UDP 有两个字段：首部字段和数据字段。

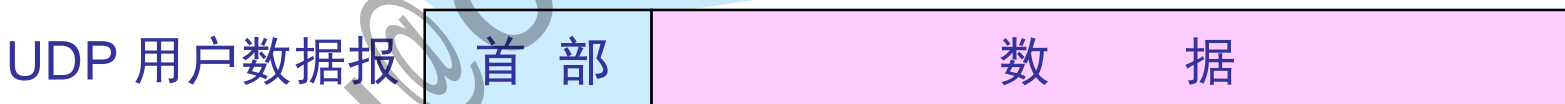
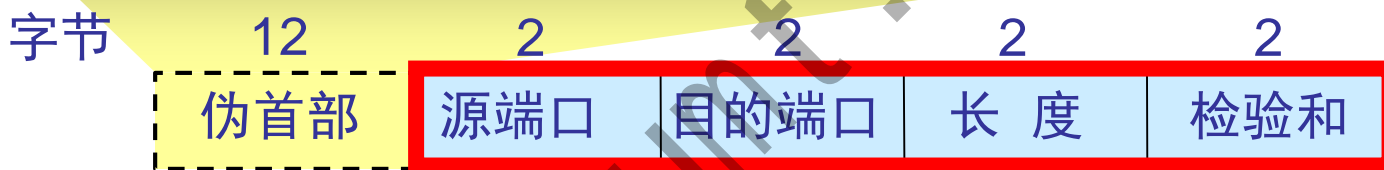




首部字段有 8 个字节，由 4 个字段组成，每个字段都是两个字节。

源端口：在需要对方回信时选用，不需要时可用全0。

目的端口：在终点交付报文时必须使用。



发送在前

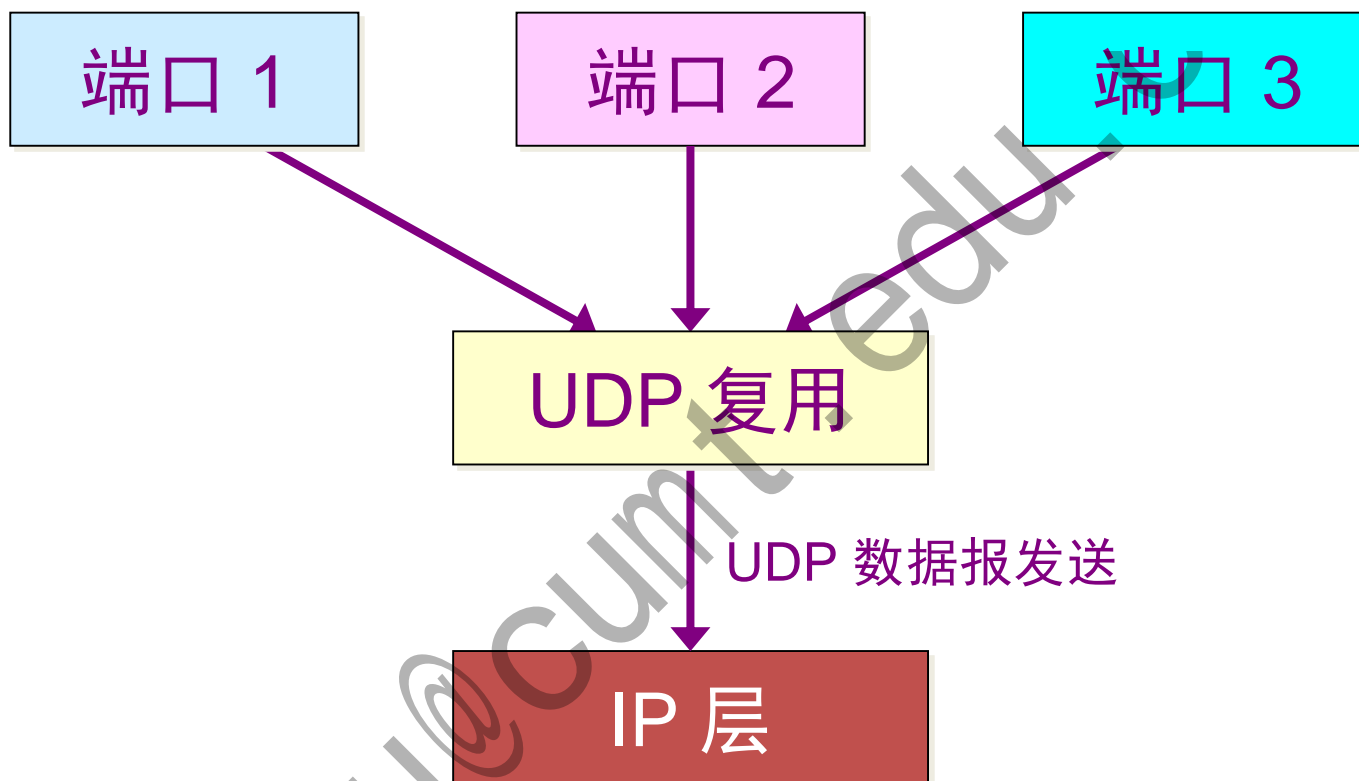


IP 数据报





# 基于端口的复用

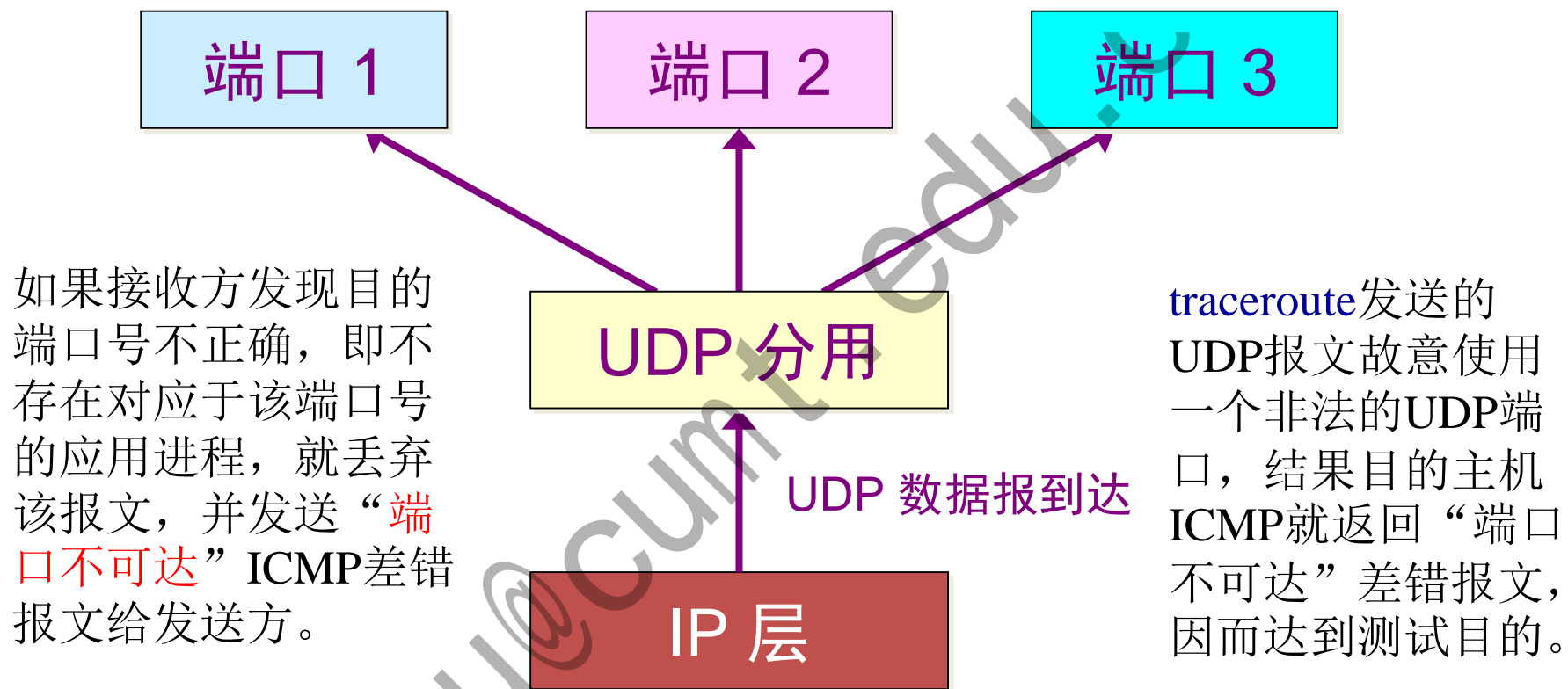


- 应用层不同进程的报文通过不同的端口向下交到运输层，再往下就共用网络层提供的服务。





# 基于端口的分用



- 运输层从IP层收到数据后必须通过不同的端口向上交付给指明的应用进程。

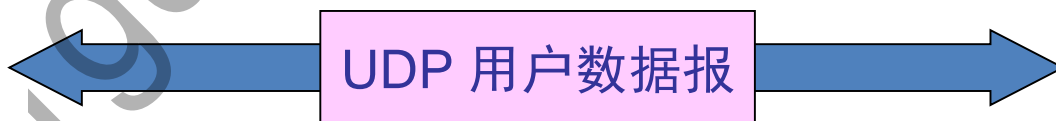
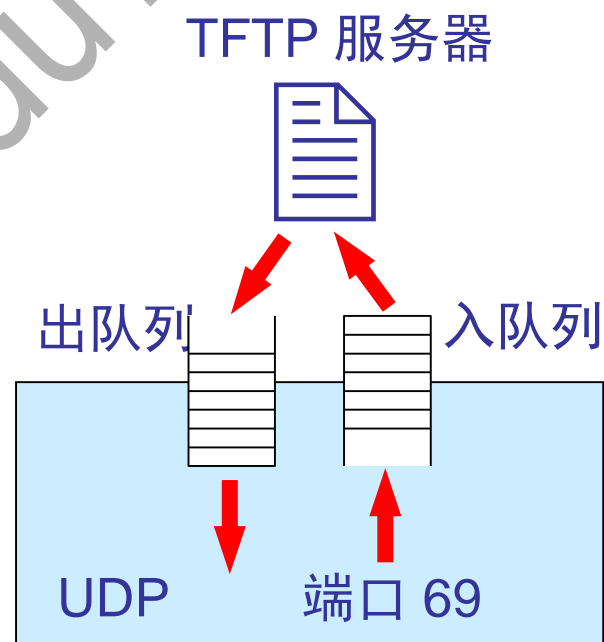
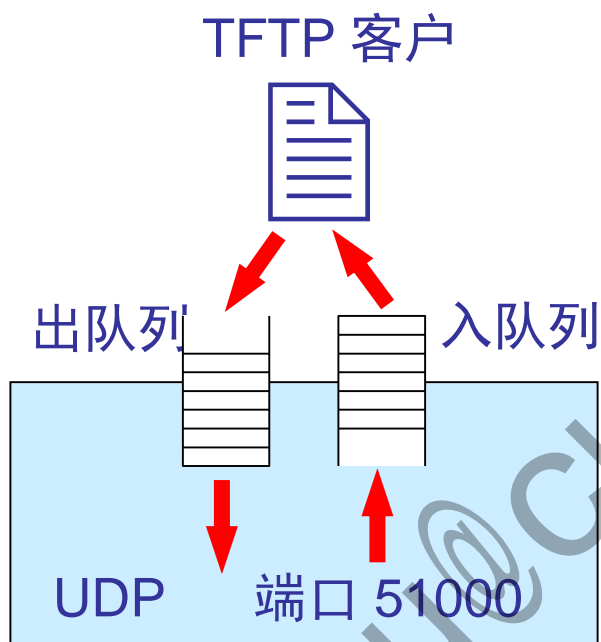




# 端口用报文队列来实现

应用层

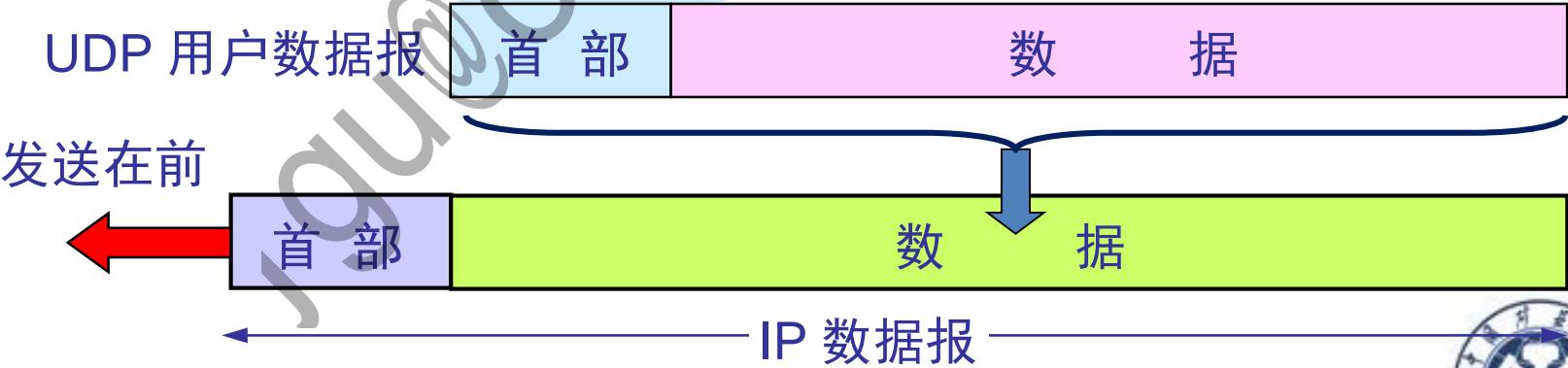
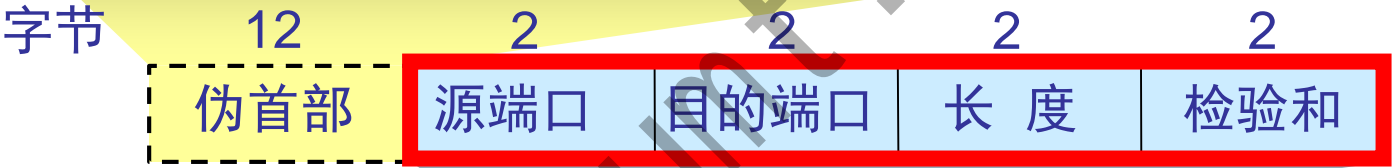
运输层





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。

长度：UDP用户数据报的长度，其最小值是8（仅有首部），最大为 $2^{16}-1=65535$ 字节。





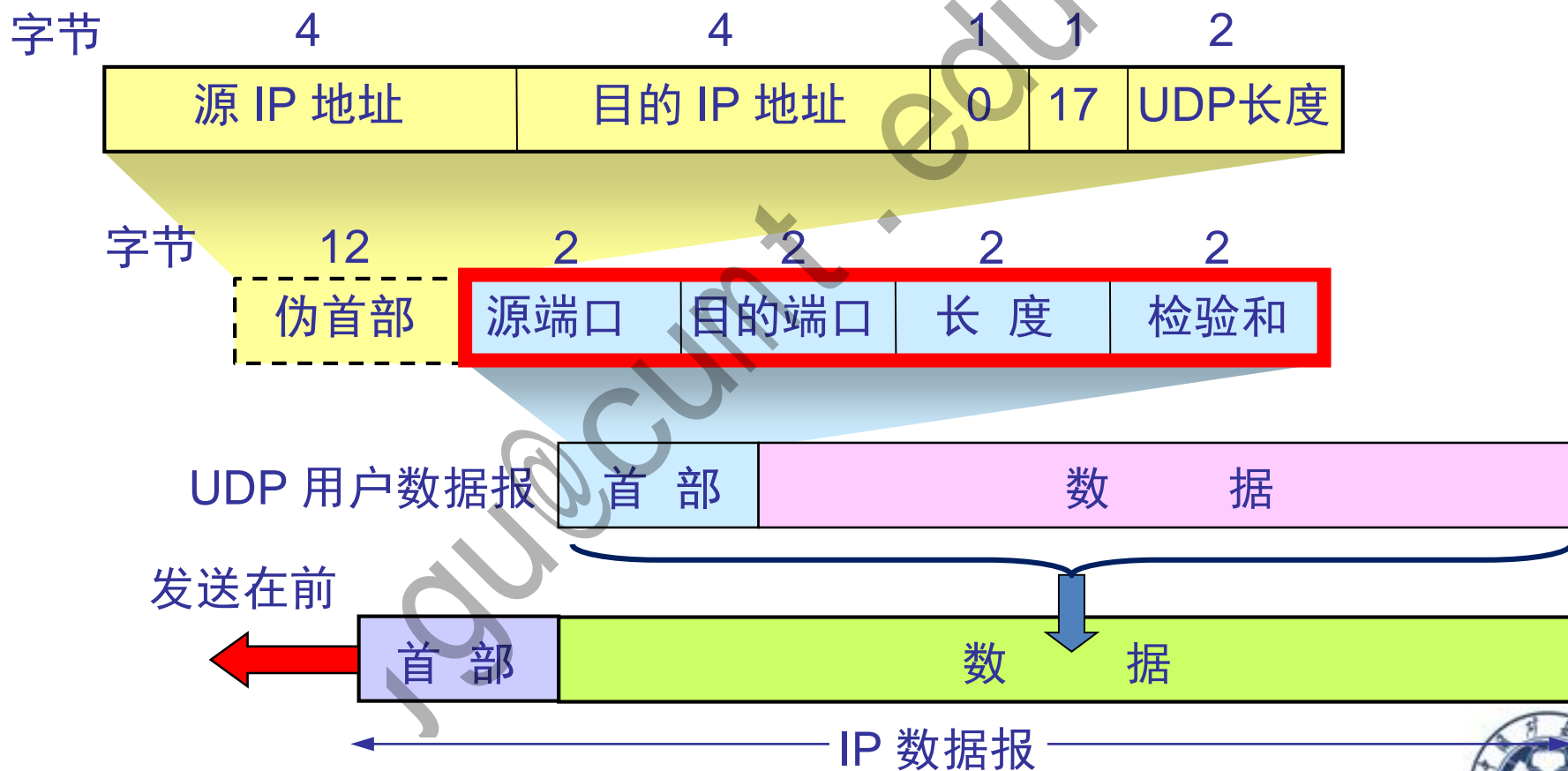
- 某客户有67000字节的分组。试说明怎样使用UDP数据报将这个分组进行传送？
- 解答：
  - 一个UDP用户数据报的最大长度是65535字节。
  - 现在的长度超过了这个限度，因此不能使用一个UDP用户数据报来传送。
  - 必须进行分割。例如，分割成为两个UDP用户数据报，使其长度不超过上述限度。



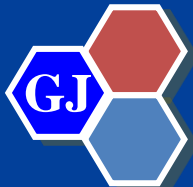


首部字段有 8 个字节，由 4 个字段组成，每个字段都是两个字节。

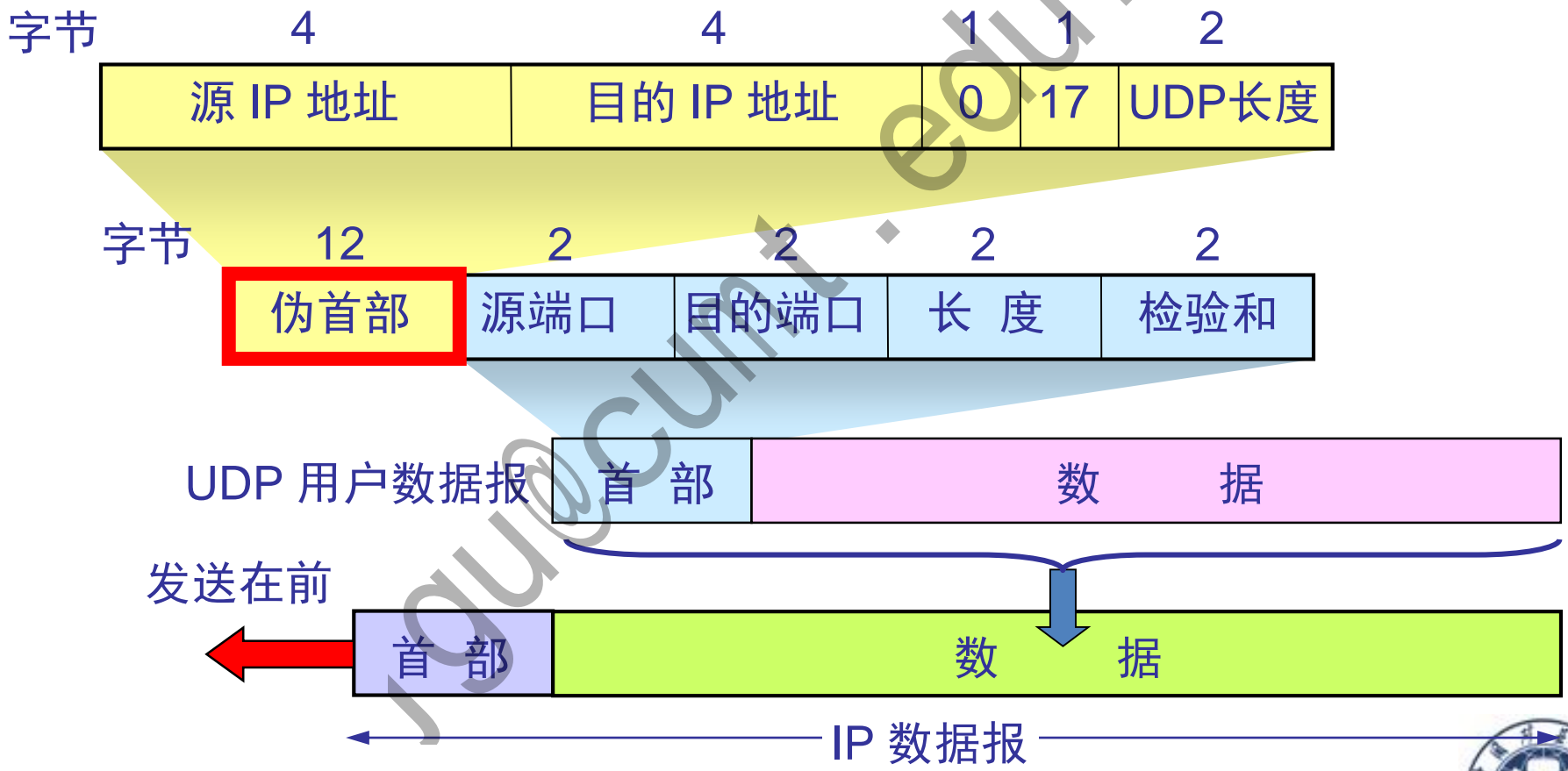
检验和：检测UDP数据报在传输中是否有错。有错就丢弃。







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				10011001 00010011	→ 153.19
	171.3.14.11				00001000 01101000	→ 8.104
	全 0	17	15		10101011 00000011	→ 171.3
8 字节 UDP 首部	1087		13		00001110 00001011	→ 14.11
	15		全 0		00000000 00010001	→ 0 和 17
7 字节 数据	数据	数据	数据	数据	00000000 00001111	→ 15
	数据	数据	数据	全 0	00000100 00111111	→ 1087
填充					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

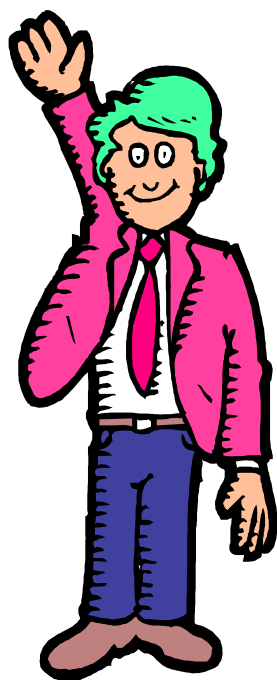




# UDP的优点

- ◆ 虽然 UDP提供不可靠交付，但有以下优点：
  - ▣ 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，即不保证可靠交付。
  - ▣ UDP 是面向报文的，不管应用层交给UDP多长的报文，UDP都照样发送，即一次发送或交付一个完整的报文。
  - ▣ 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，这对某些实时应用是很重要的。
    - ✓ 多媒体通信要求恒定速率，并允许拥塞时会有一些数据丢失。
  - ▣ 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。





**THANK  
YOU!**

