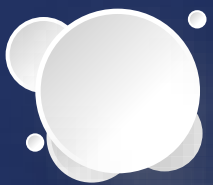


A faint, light gray world map serves as the background for the slide, centered behind the text.

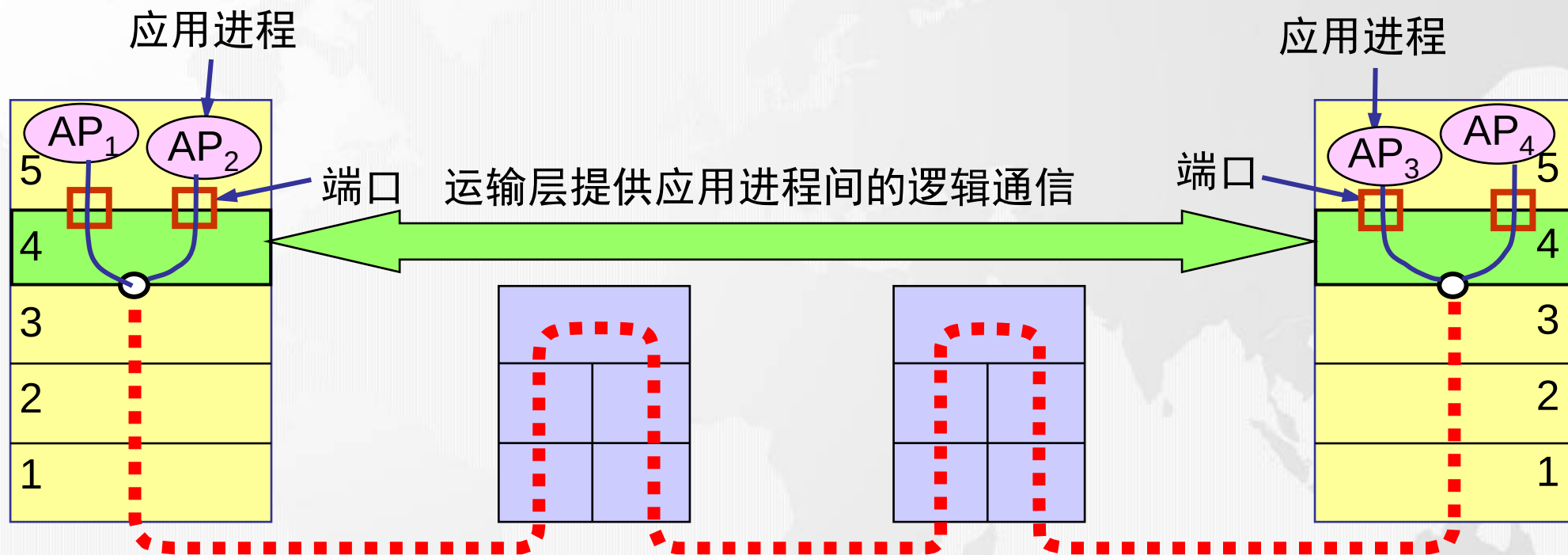
# 计算机网络

运输层

苏铅坤



# 运输层



# 目录

## CONTENTS



1

运输层作用：进程的通信

2

运输层作用：端口

3

运输层作用：可靠传输

4

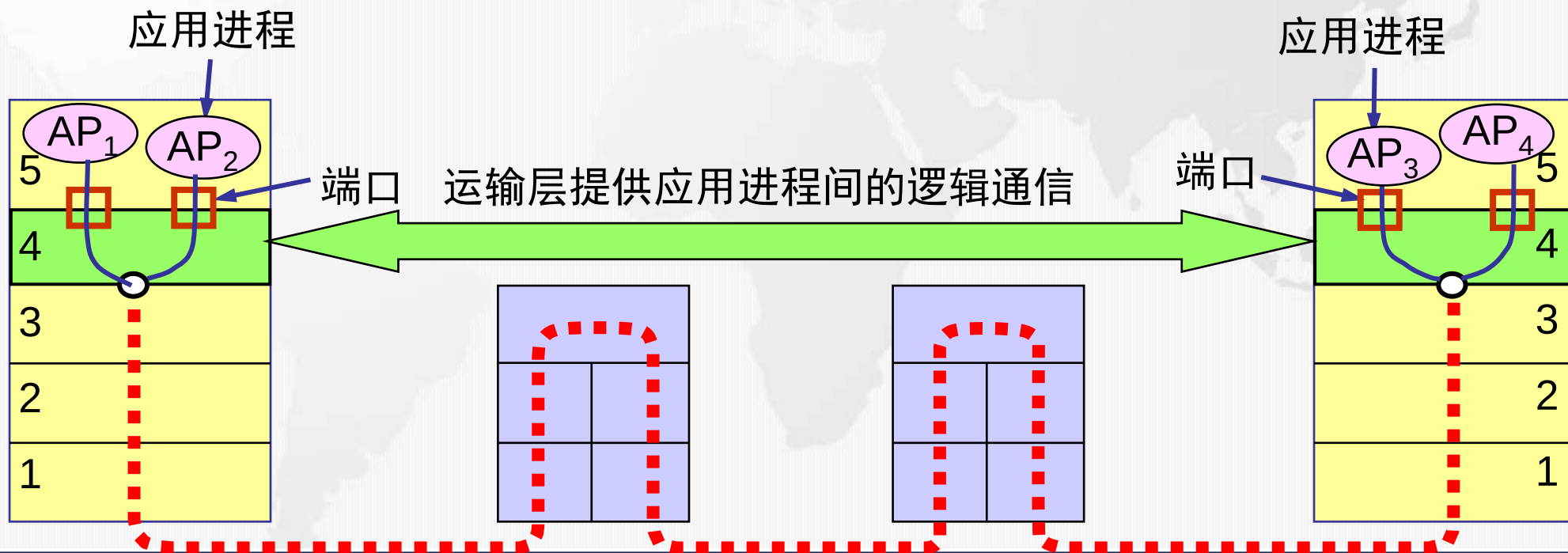
运输层协议

5



# 进程间的通信

- ☺ 两台主机进行通信实质上是两台主机中的应用进程进行通信
  - 通信的真正端点并不是主机而是主机中的进程，即端到端的通信是应用进程之间的通信



# 目录

## CONTENTS



1

运输层作用：进程的通信

2

运输层作用：端口

3

运输层作用：可靠传输

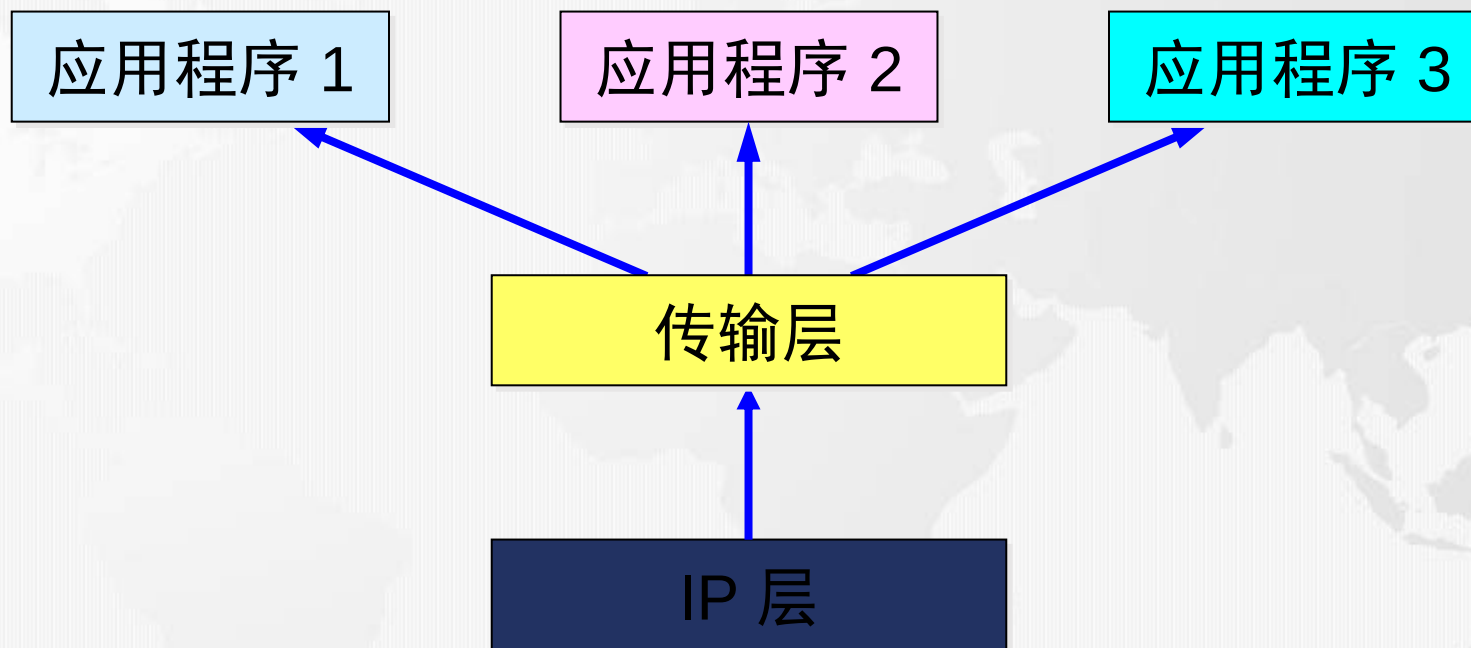
4

运输层协议

5

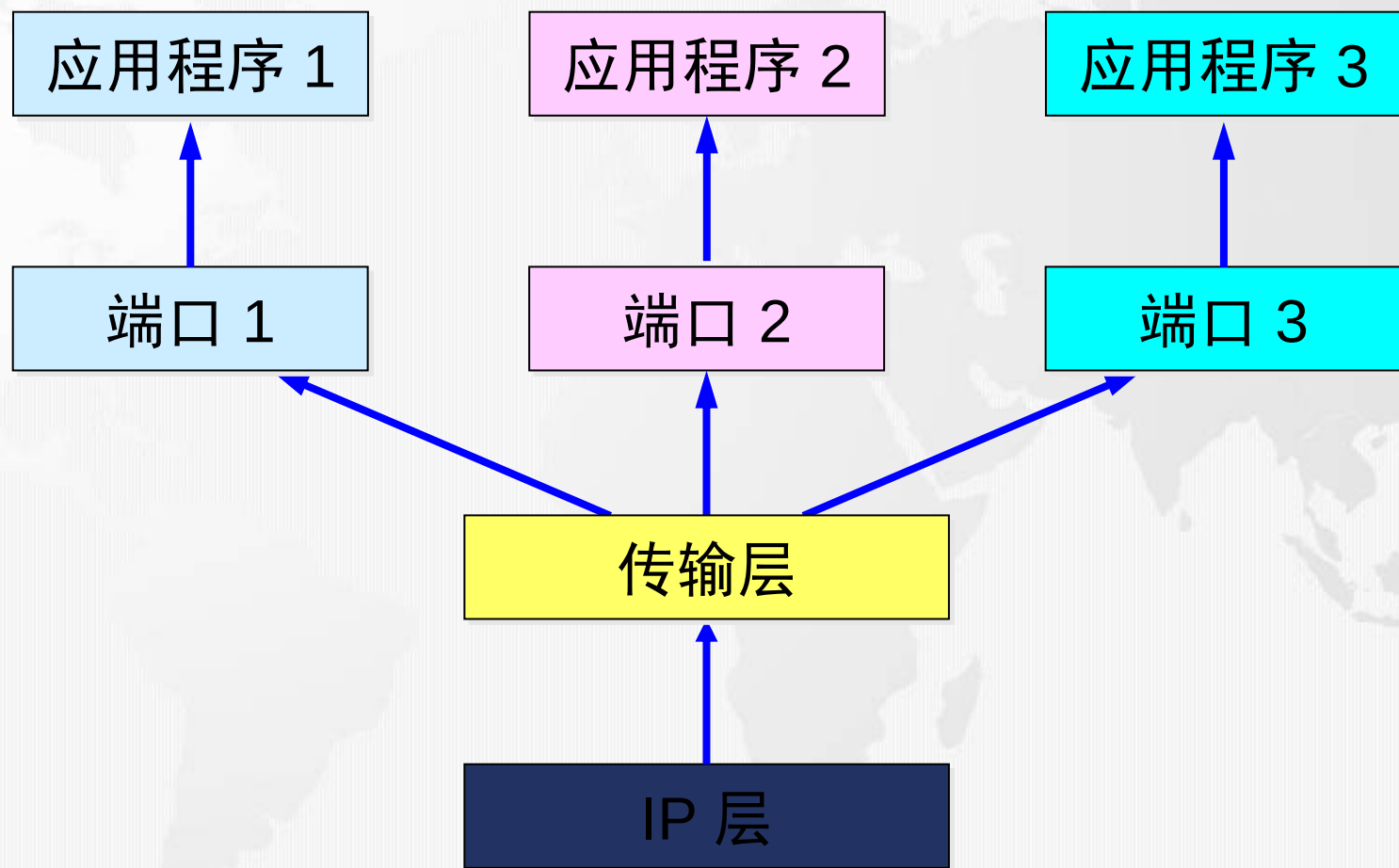


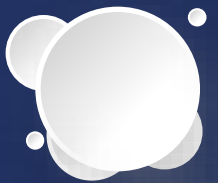
# 报文交给哪个应用程序?





# 通过端口分用

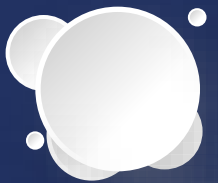




# 运输层的端口

- ☺ 运行在计算机中的进程是用进程标识符来标志的
  - 进程的创建和撤销都是动态
  - 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
- ☺ 为了使运行不同计算机网络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
  - 利用目的主机提供的功能来识别终点，而不需要知道实现这个功能的进程
  - 运输层使用协议端口号 (protocol port number)，简称端口 (port)





# 软件端口与硬件端口

## ☺ 软件端口

- 在协议栈层间的抽象的协议端口
- 软件端口是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

## ☺ 硬件端口

- 路由器或交换机上的端口
- 硬件端口是不同硬件设备进行交互的接口



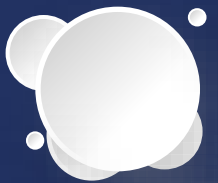
# 运输层的端口

## ☺ 端口用一个16位整数来标识

- 端口号只具有本地意义，即端口号只是为了标志本计算机应用层中的各进程
- 在互联网中，不同计算机的相同端口号是没有联系的

## ☺ 两个计算机中的进程要互相通信

- 不仅必须知道对方的 IP 地址（为了找到对方的计算机）
- 而且还要知道对方的端口号（为了找到对方计算机中的应用进程）



# 代码实例

```
import socket

# Create a TCP/IP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM) # socket.SOCK_STREAM

# Bind the socket to the port
server_addr = ('localhost', 20000)
sock.bind(server_addr)
```

```
try:
    while True:
        # Receive
        data, client_addr = sock.recvfrom  import socket
        if data:
            print('The client {} say : {}'.format(client_addr, data))
```

```
# Create a TCP/IP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

try:
    while True:
        # send
        message = input()
        server_addr = ('localhost', 20000)
        sent = sock.sendto(message.encode(), server_addr)
```



# 还记得这个例子吗?

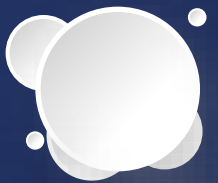


☺ <https://www.baidu.com/>

☺ <https://www.baidu.com:80/index.html>

端口如何分配？





# 端口分配

## ☺ 服务器端使用的端口号

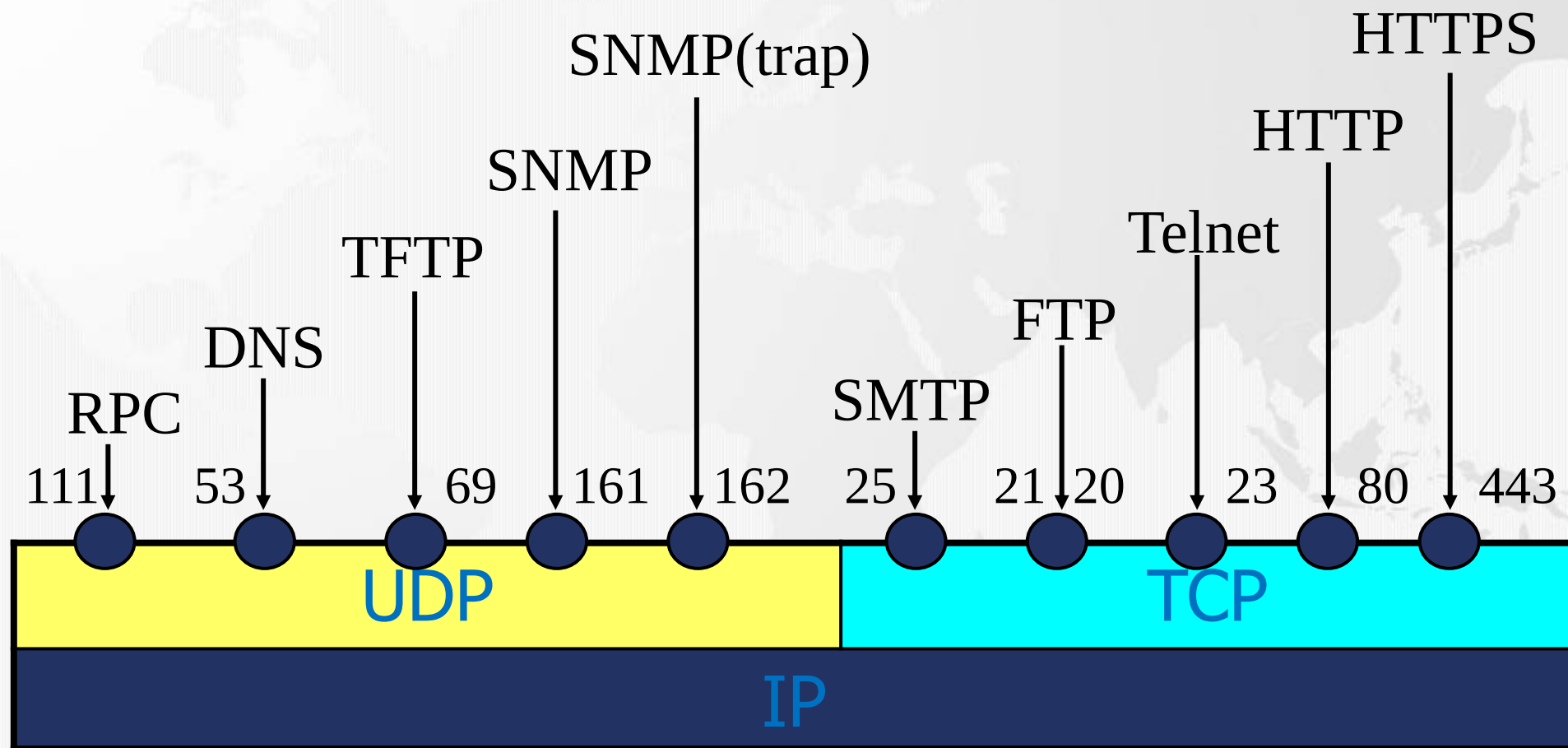
- 熟知端口，数值一般为0 ~ 1023
- 登记端口号，数值为1024 ~ 49151，为没有熟知端口号的应用程序使用的。使用这个范围的端口号必须在 IANA 登记，以防止重复

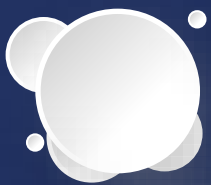
## ☺ 客户端使用的端口号

- 又称为短暂端口号，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。

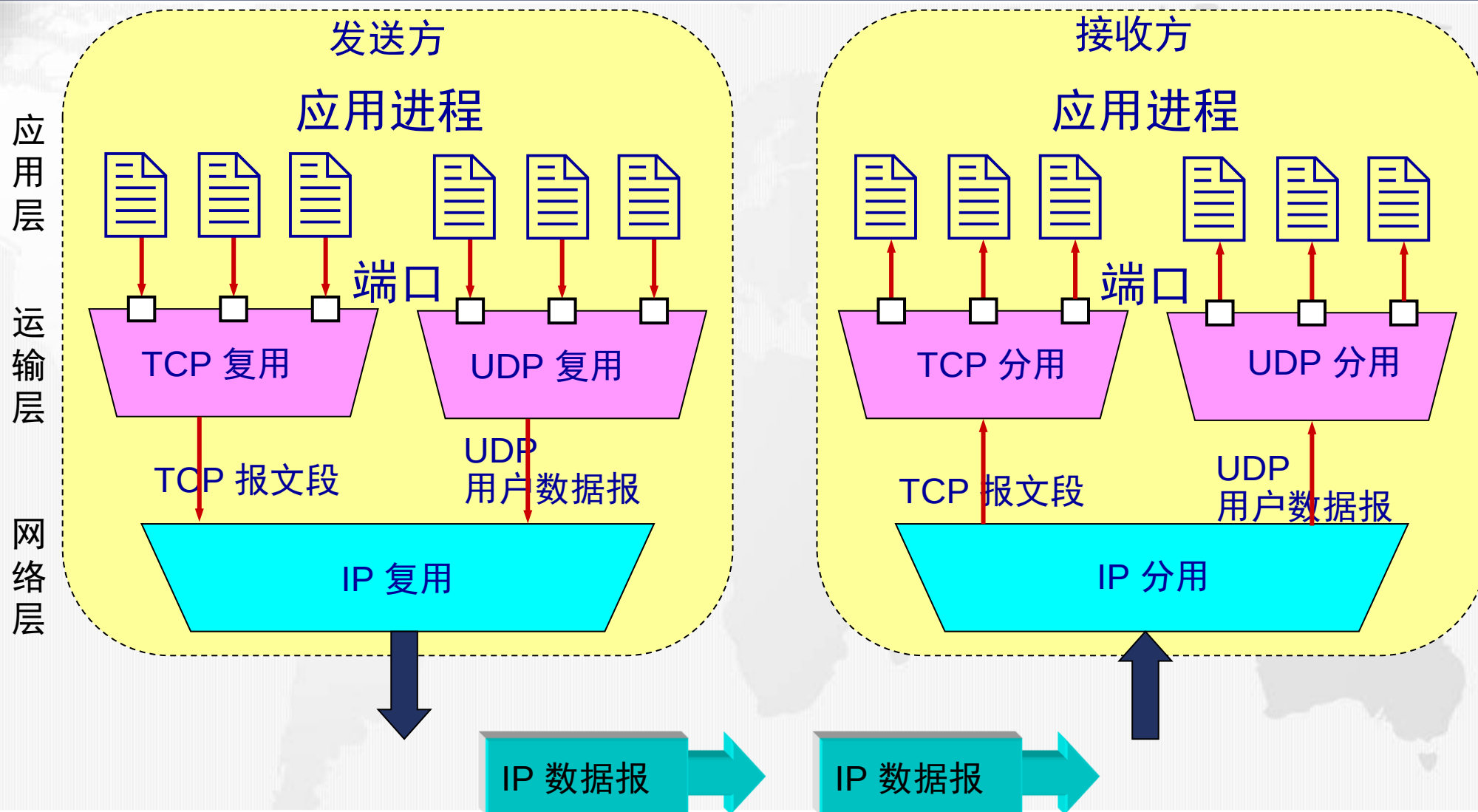


# 常用的熟知端口





# 基于端口的分用与复用





# 目录

## CONTENTS



1

运输层作用：进程的通信

2

运输层作用：端口

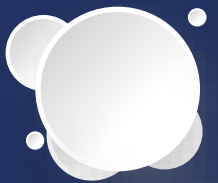
3

运输层作用：可靠传输

4

运输层协议

5



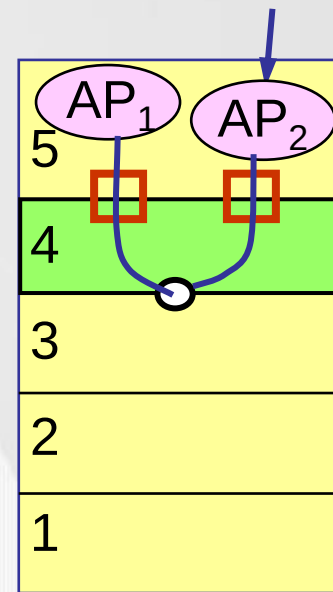
# 应用不同

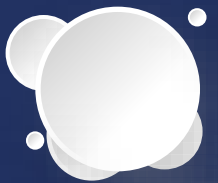
## ☺ 传输存在

- 丢包：网络太拥挤、链路太差
- 错包：对收到的报文进行差错检测

## ☺ 是否需要可靠传输

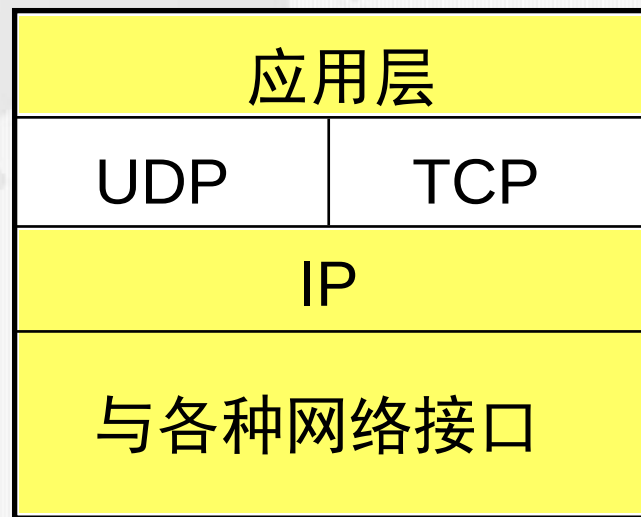
- 网页浏览 vs 视频播放





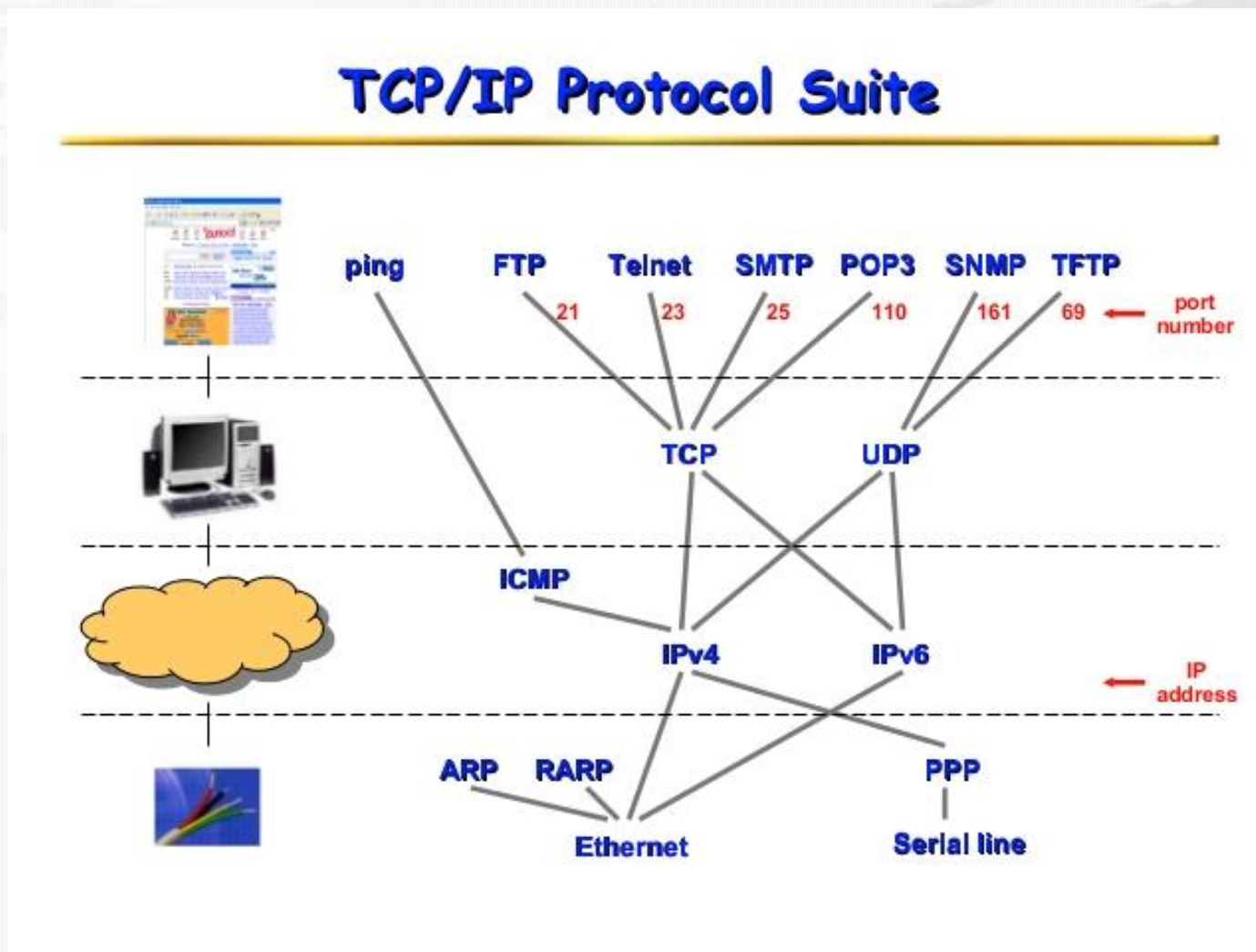
# TCP与UDP

- ☺ 根据应用程序的不同需求，运输层需要有两种不同的运输协议：
- 用户数据报协议 UDP (User Datagram Protocol)
  - 传输控制协议 TCP (Transmission Control Protocol)





# 所学应用层协议用的是TCP还是UDP?



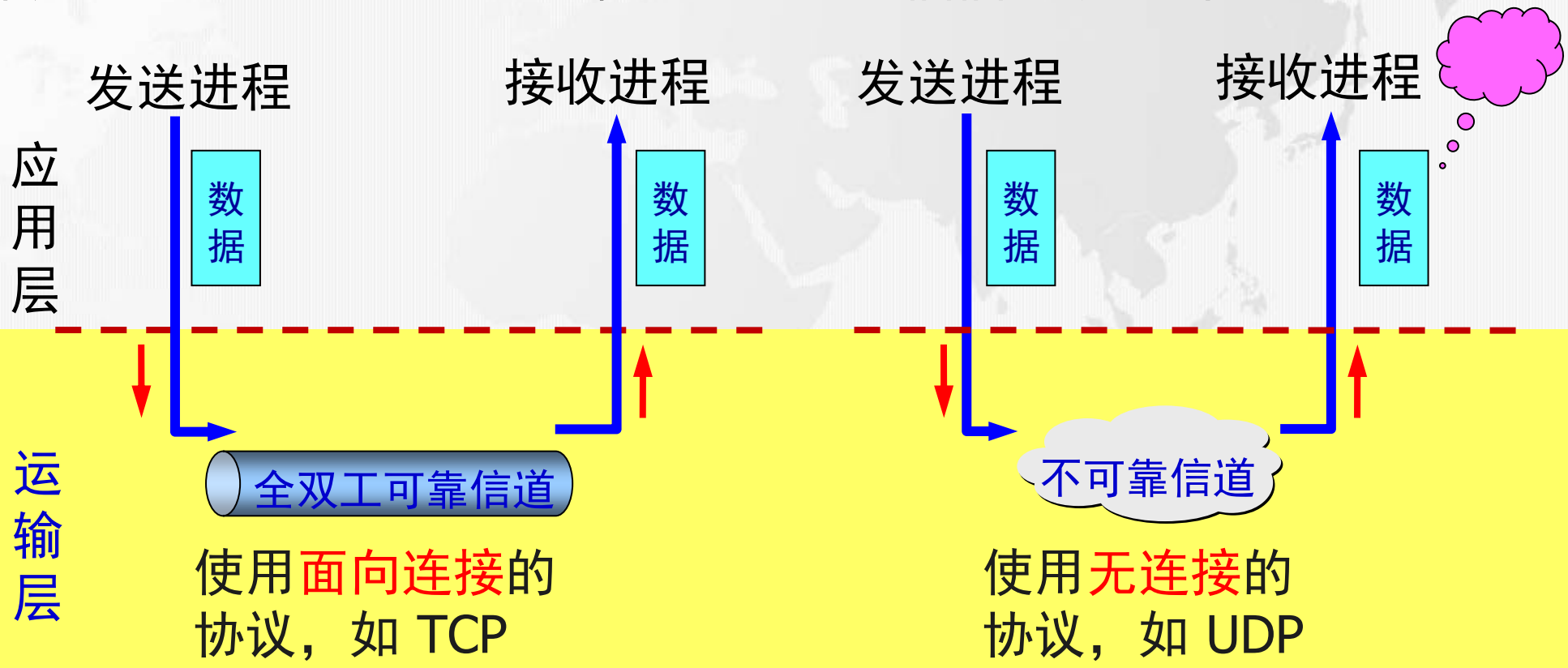


# 可靠信道和不可靠信道

☺ 运输层下面的网络是不可靠的（只提供尽最大努力服务）

○ 当运输层采用面向连接的TCP协议时，这种逻辑通信信道就相当于一端全双工的

可靠信道



- ☺ 一种无连接协议
- ☺ 提供无连接服务
- ☺ 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

- ☺ 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。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。



# Q & A

[qiankun.su@jmu.edu.cn](mailto:qiankun.su@jmu.edu.cn)

苏铅坤