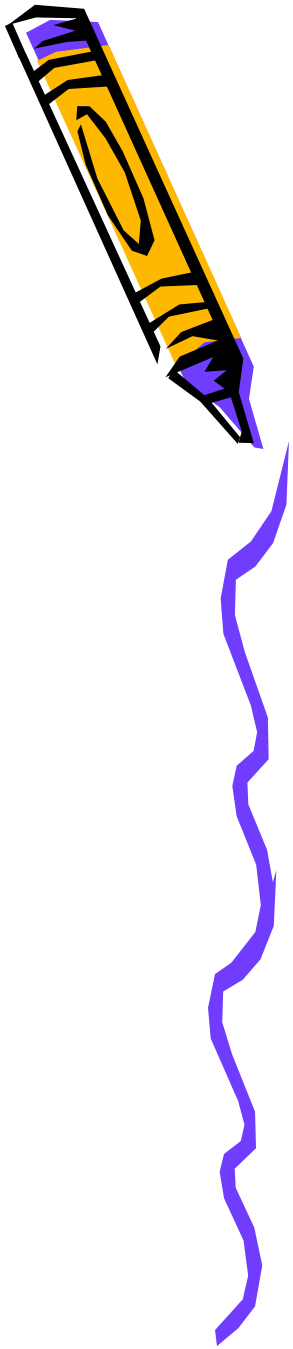
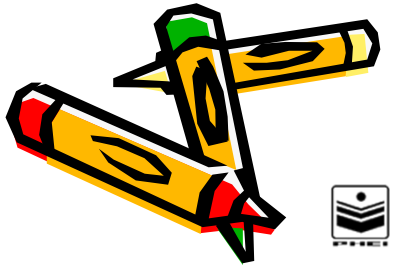
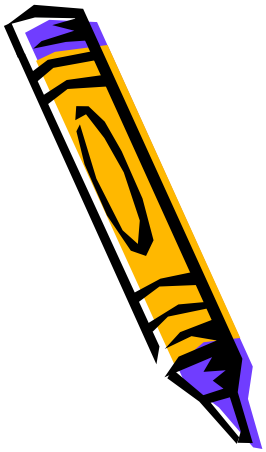


第9章 网络通信

- 9.1 网络编程基础
- 9.2 使用URL访问网络资源
- 9.3 TCP Socket通信
- 9.4 UDP数据报通信



第9章 网络通信



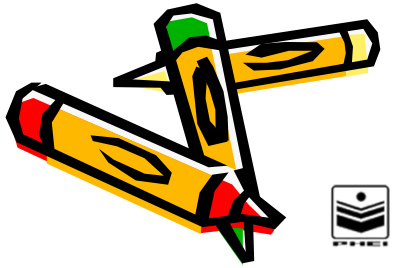
内容和要求：

1．熟悉URL访问网络资源。

2．掌握TCP Socket通信和UDP数据报通信。

重点： TCP Socket， UDP Socket。

难点： TCP Socket， UDP Socket。

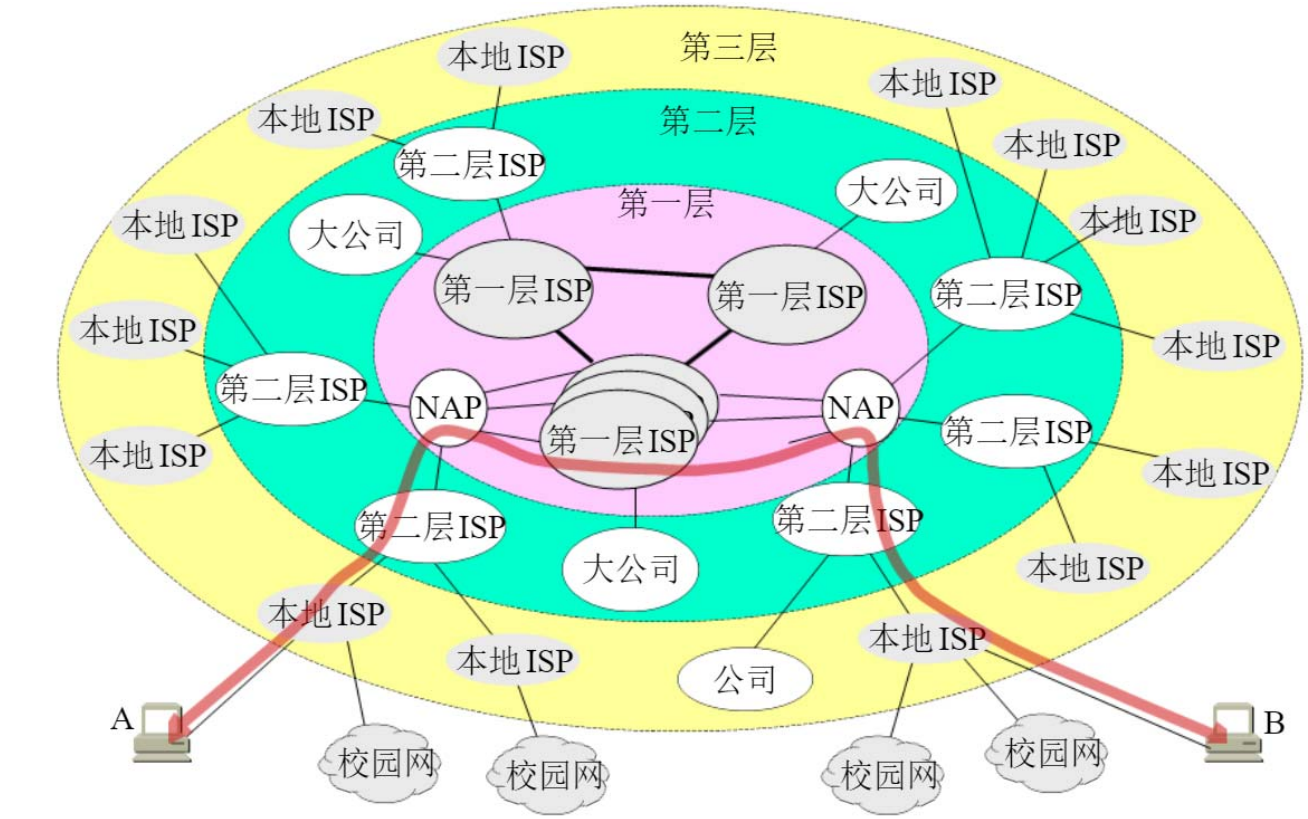


9.1 网络编程基础

1. 计算机网络与**Internet**
2. **TCP/IP**协议
3. **Internet**地址
   - ① **IP**地址，形如**xxx.xxx.xxx.xxx**
   - ② 域名系统。例如**www.edu.cn**
4. **URL**（统一资源定位符）

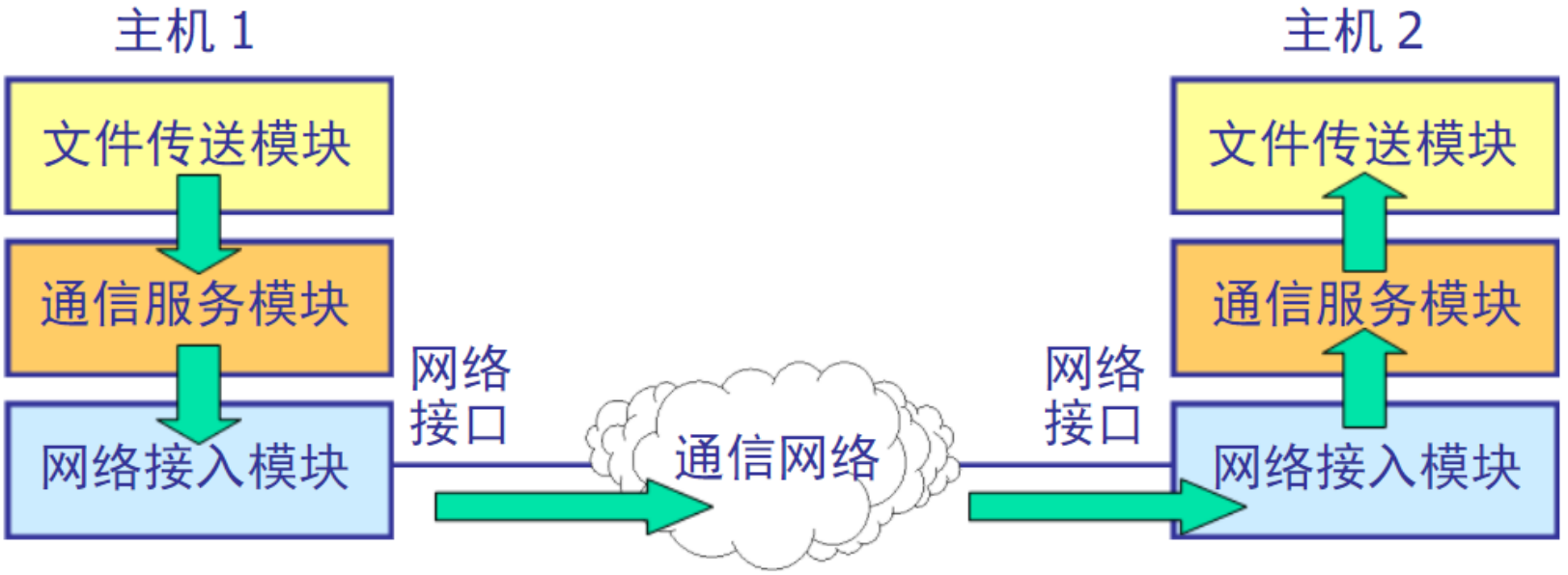
   协议 **://** 主机 **[:** 端口**] [/** 文件**] [#** 引用**]**
5. 客户**-**服务器（**Client-Server**）模式

9.1 网络编程基础



主机A → 本地 ISP → 第二层 ISP → NAP → 第一层 ISP → NAP → 第二层 ISP → 本地 ISP → 主机B

9.1 网络编程基础

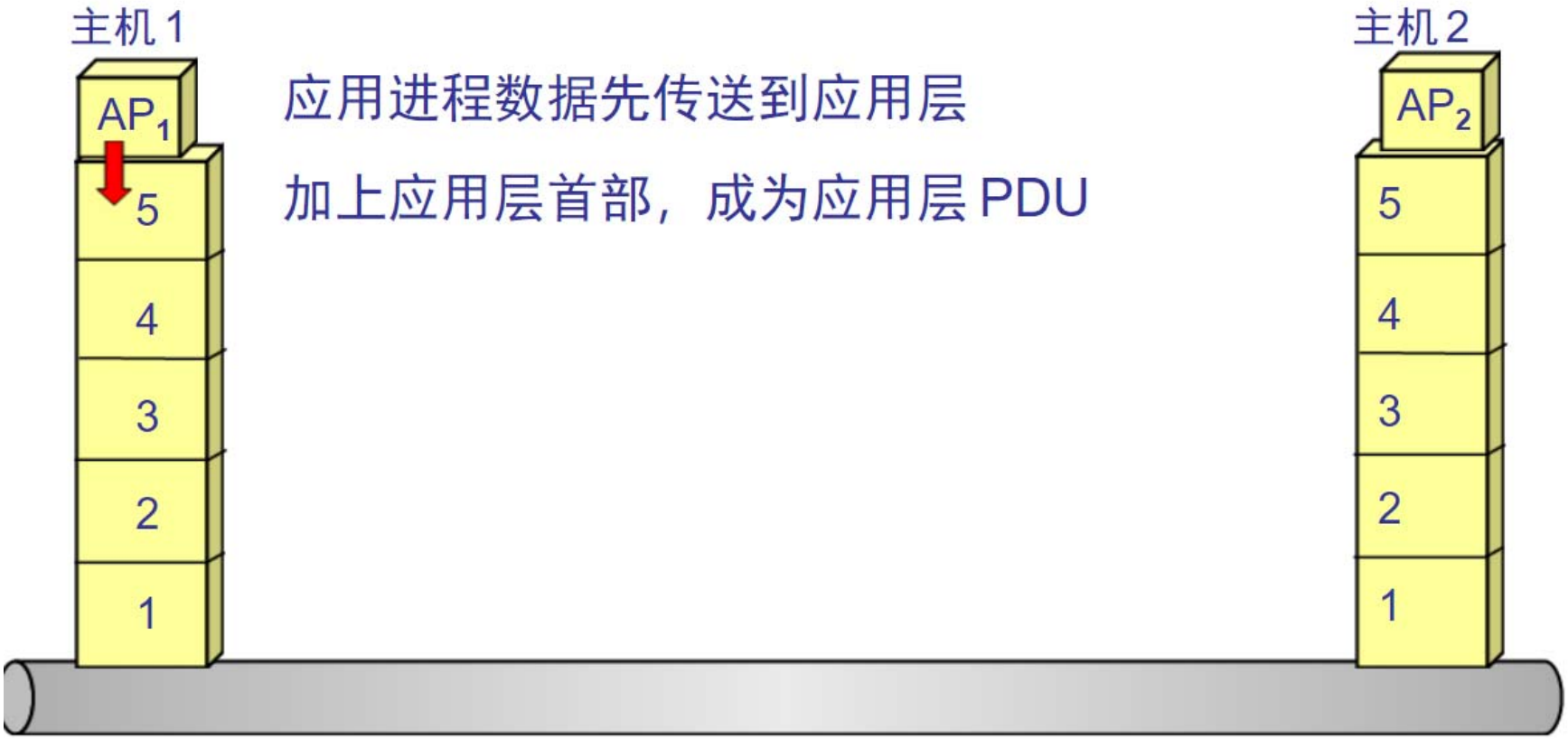


网络接入模块负责做与网络接口细节有关的工作
例如，规定传输的帧格式，帧的最大长度等。

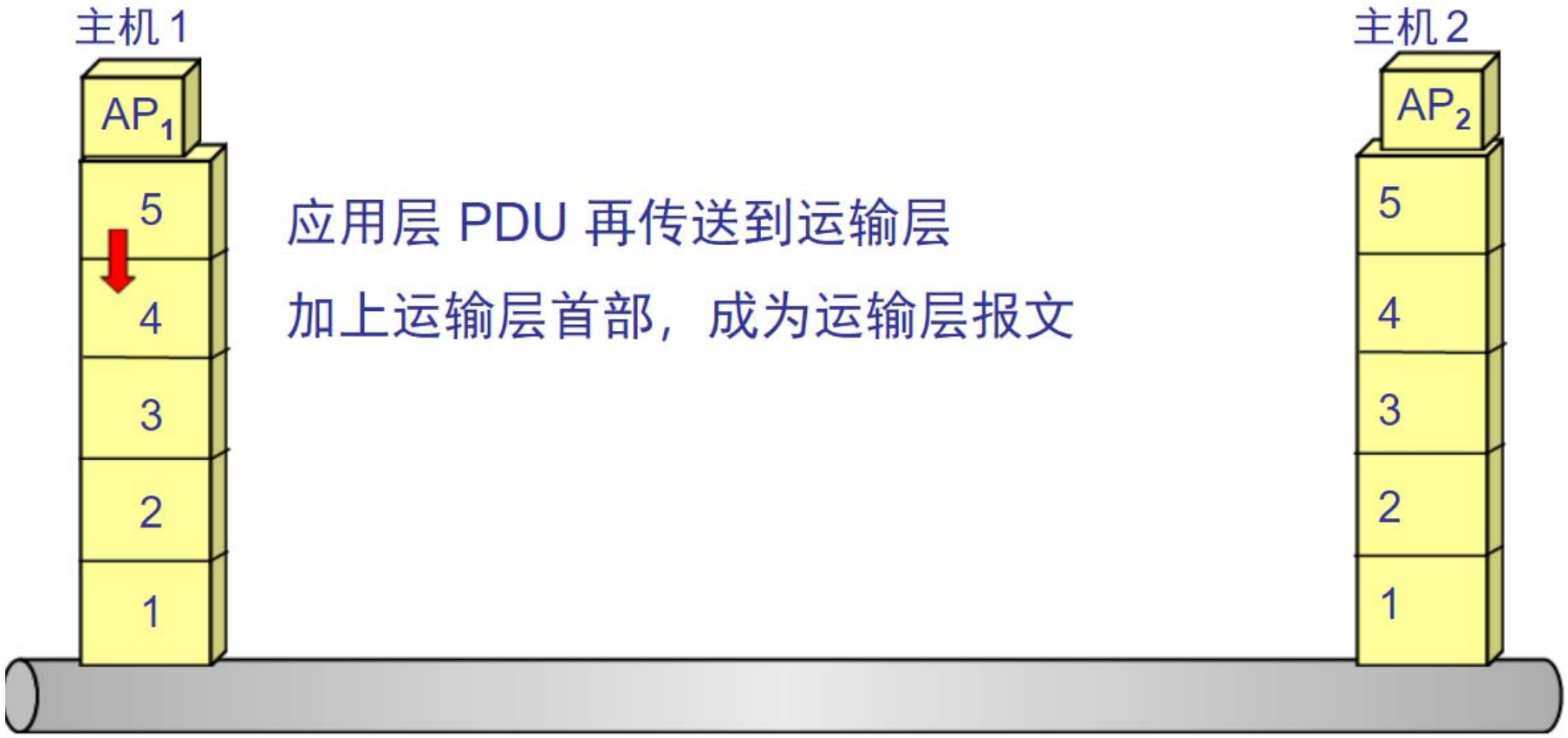
9.1 网络编程基础

- 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)

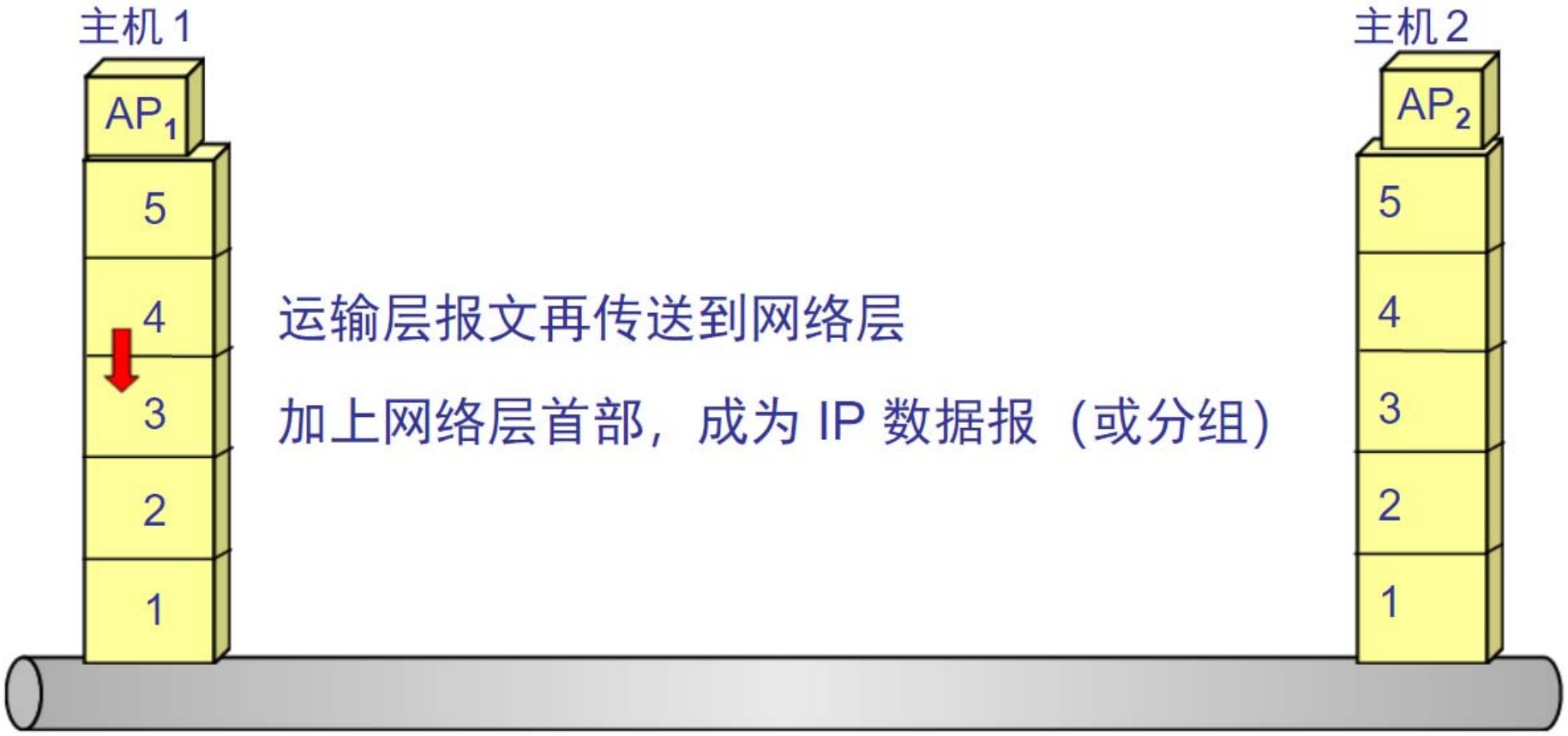
9.1 网络编程基础


主机1
AP1
5
4
3
2
1
应用进程数据先传送到应用层
加上应用层首部，成为应用层PDU
主机2
AP2
5
4
3
2
1

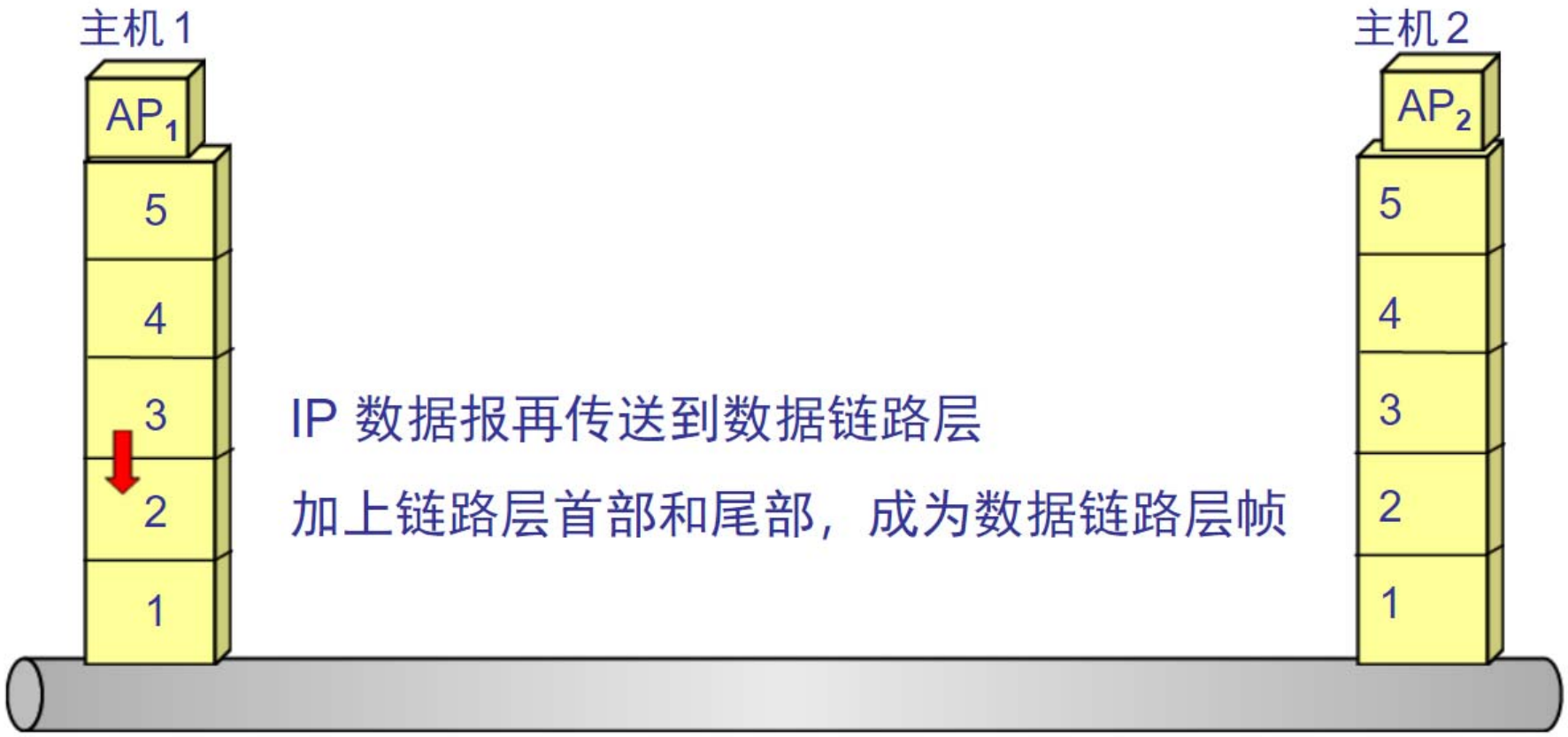
9.1 网络编程基础



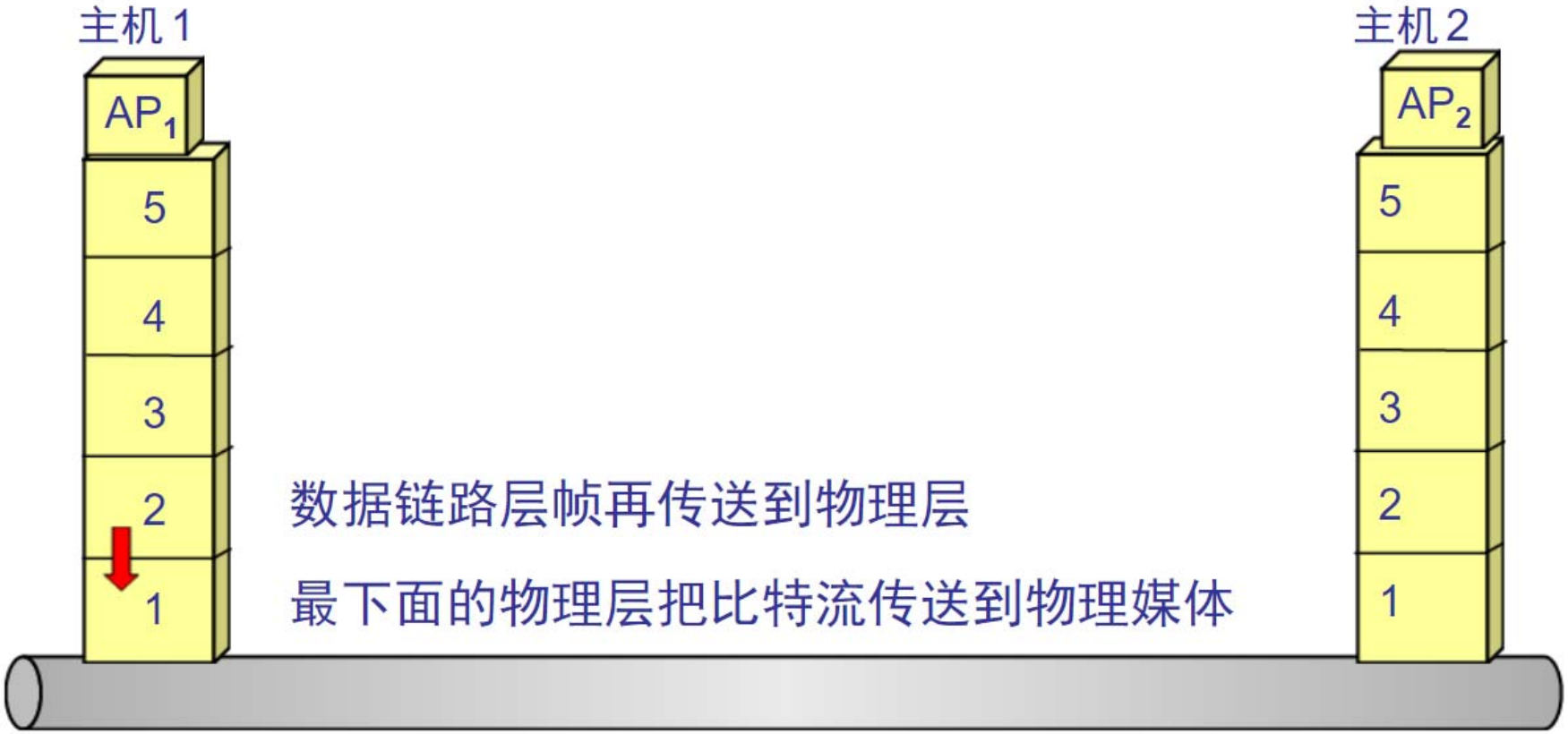
9.1 网络编程基础


主机 1
AP1
5
4
3
2
1
主机 2
AP2
5
4
3
2
1
运输层报文再传送到网络层
加上网络层首部，成为 IP 数据报（或分组）

9.1 网络编程基础



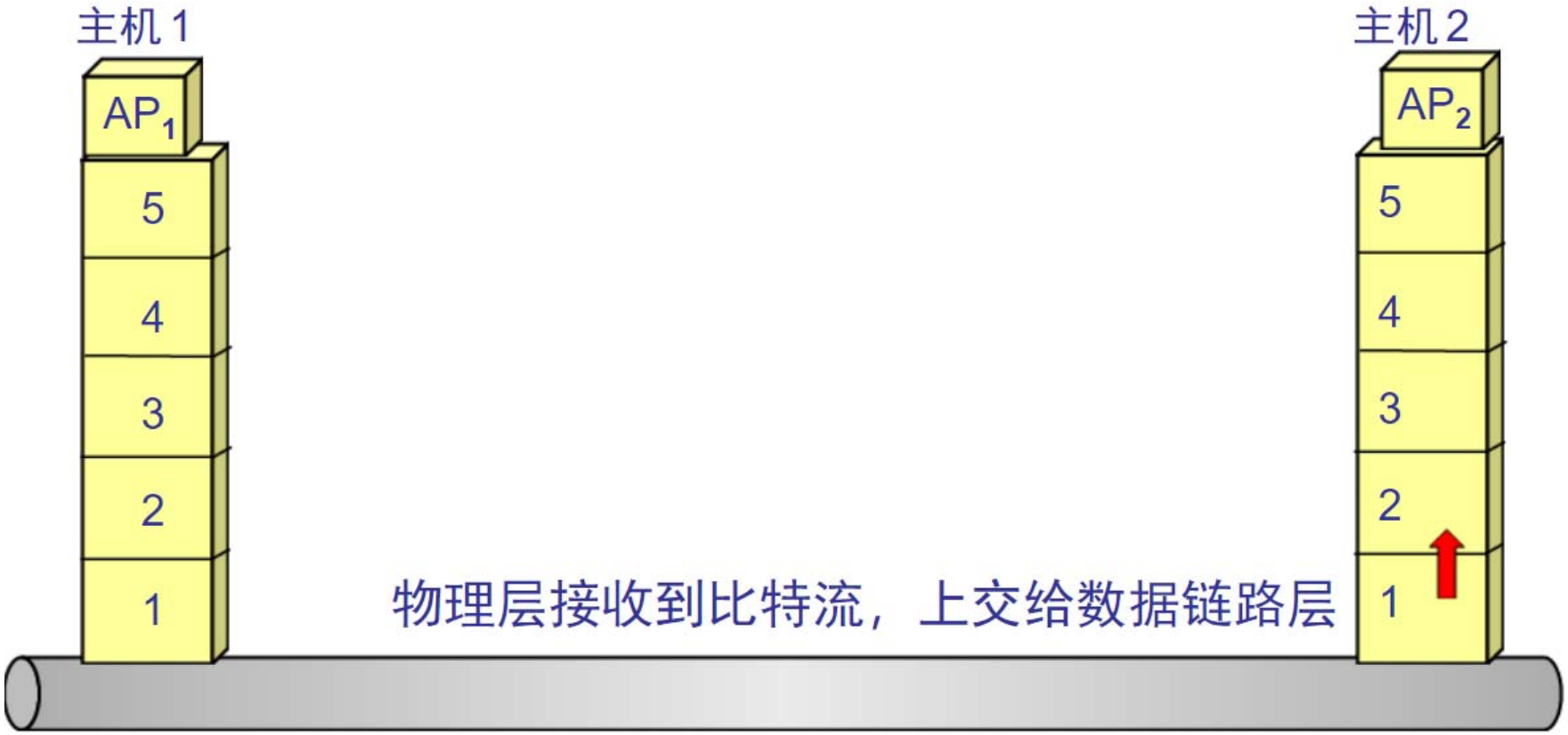
9.1 网络编程基础


主机1
AP1
5
4
3
2
1
主机2
AP2
5
4
3
2
1
数据链路层帧再传送到物理层
最下面的物理层把比特流传送到物理媒体

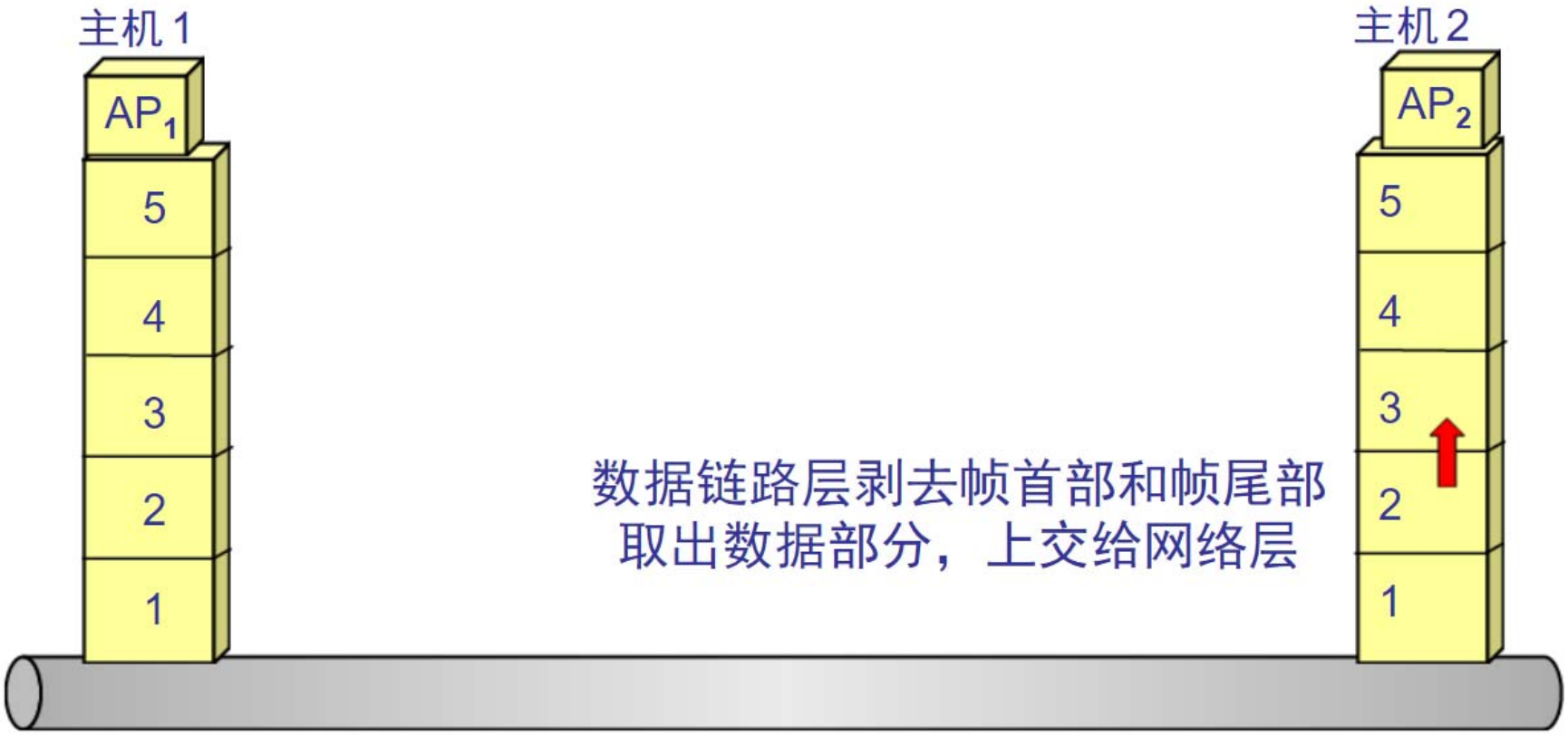
9.1 网络编程基础


主机1
主机2
AP1
AP2
5
4
3
2
1
5
4
3
2
1
电信号（或光信号）在物理媒体中传播
从发送端物理层传送到接收端物理层
物理传输媒体

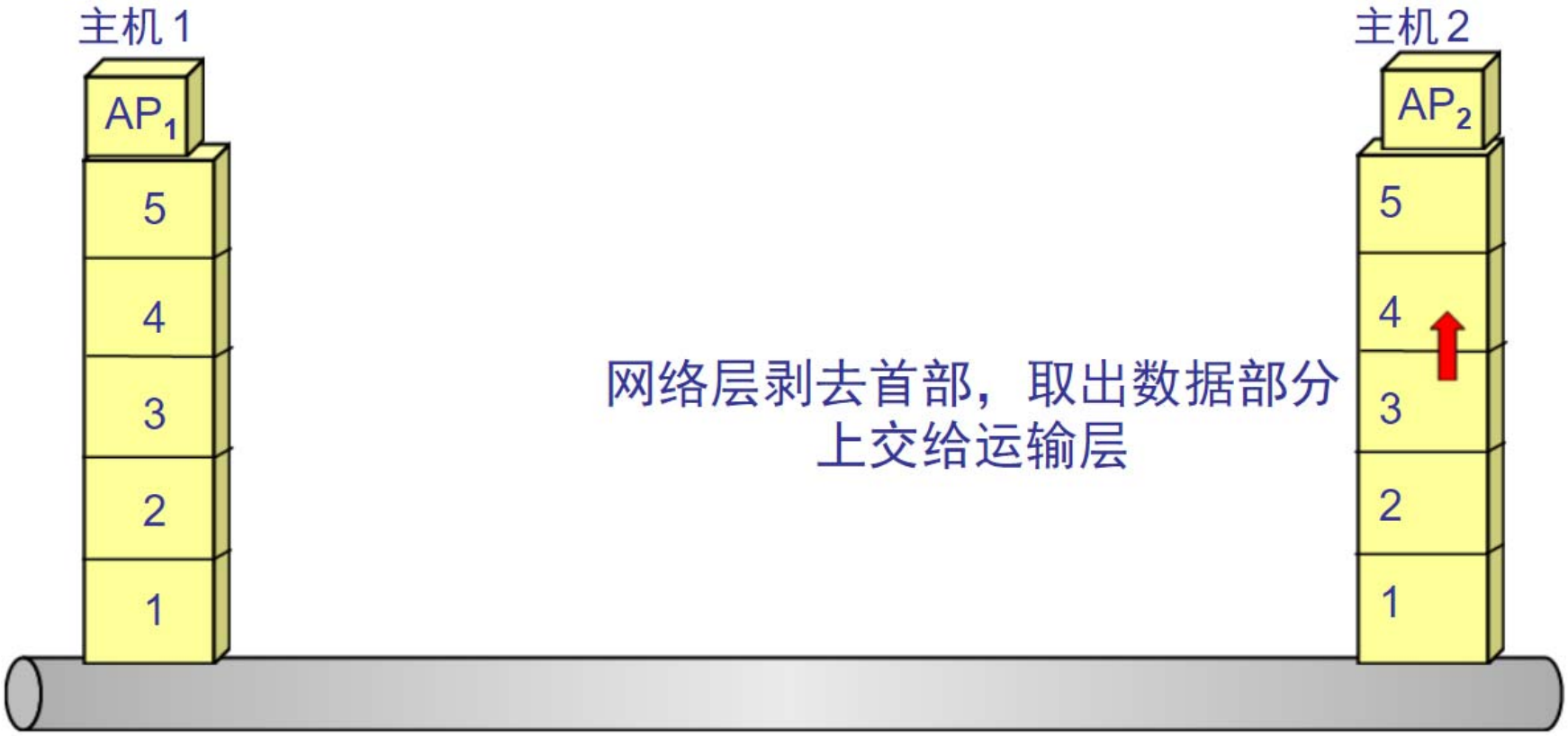
9.1 网络编程基础



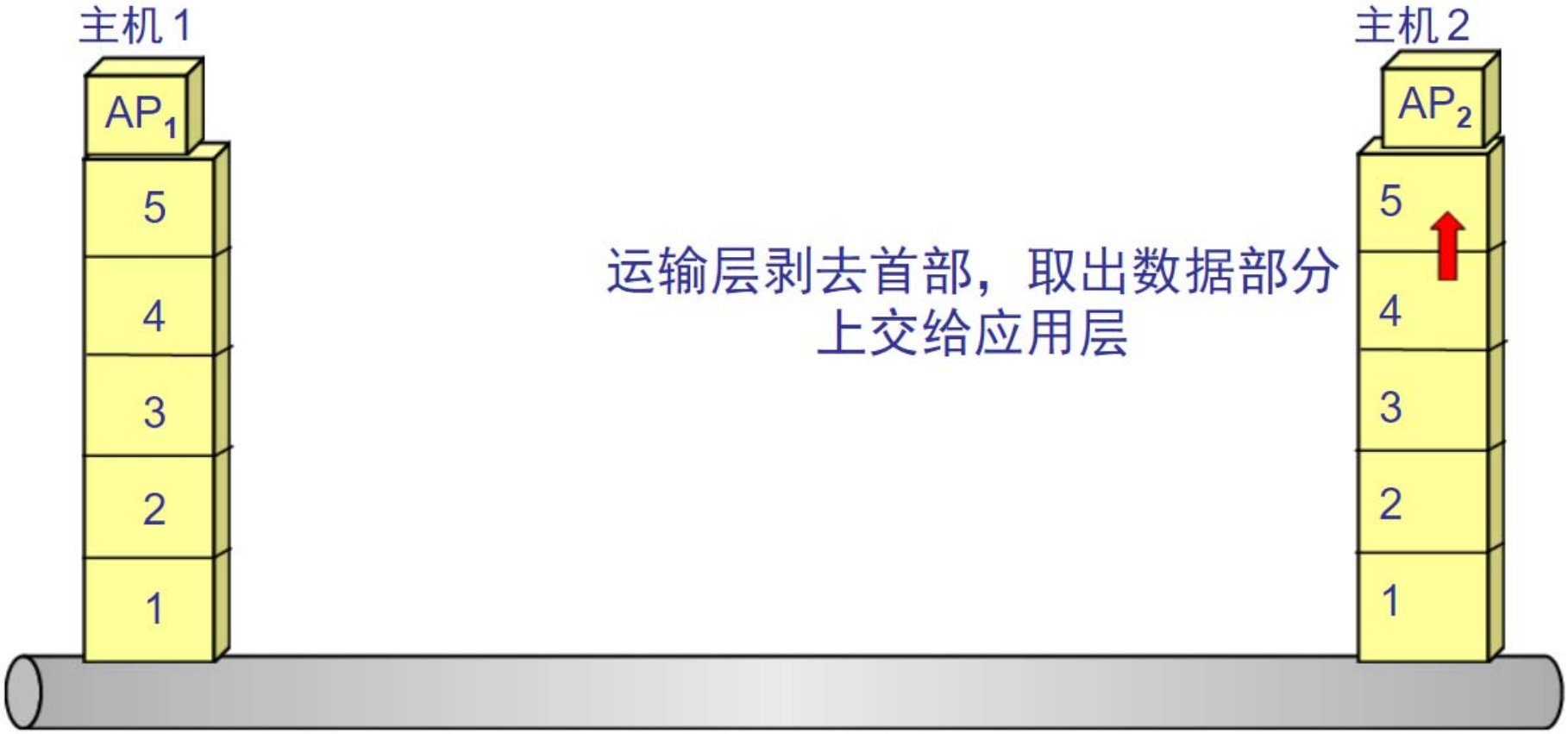
9.1 网络编程基础



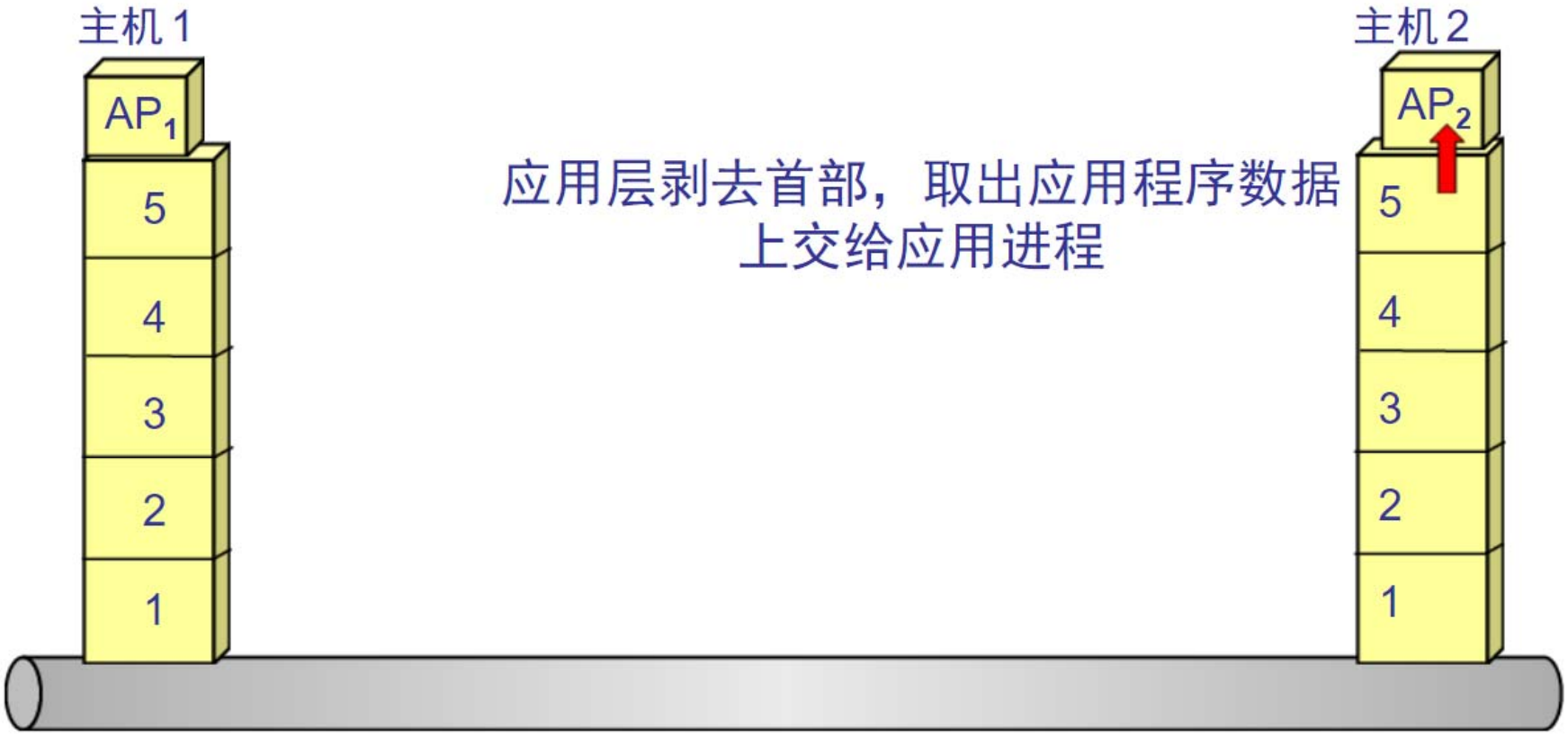
9.1 网络编程基础



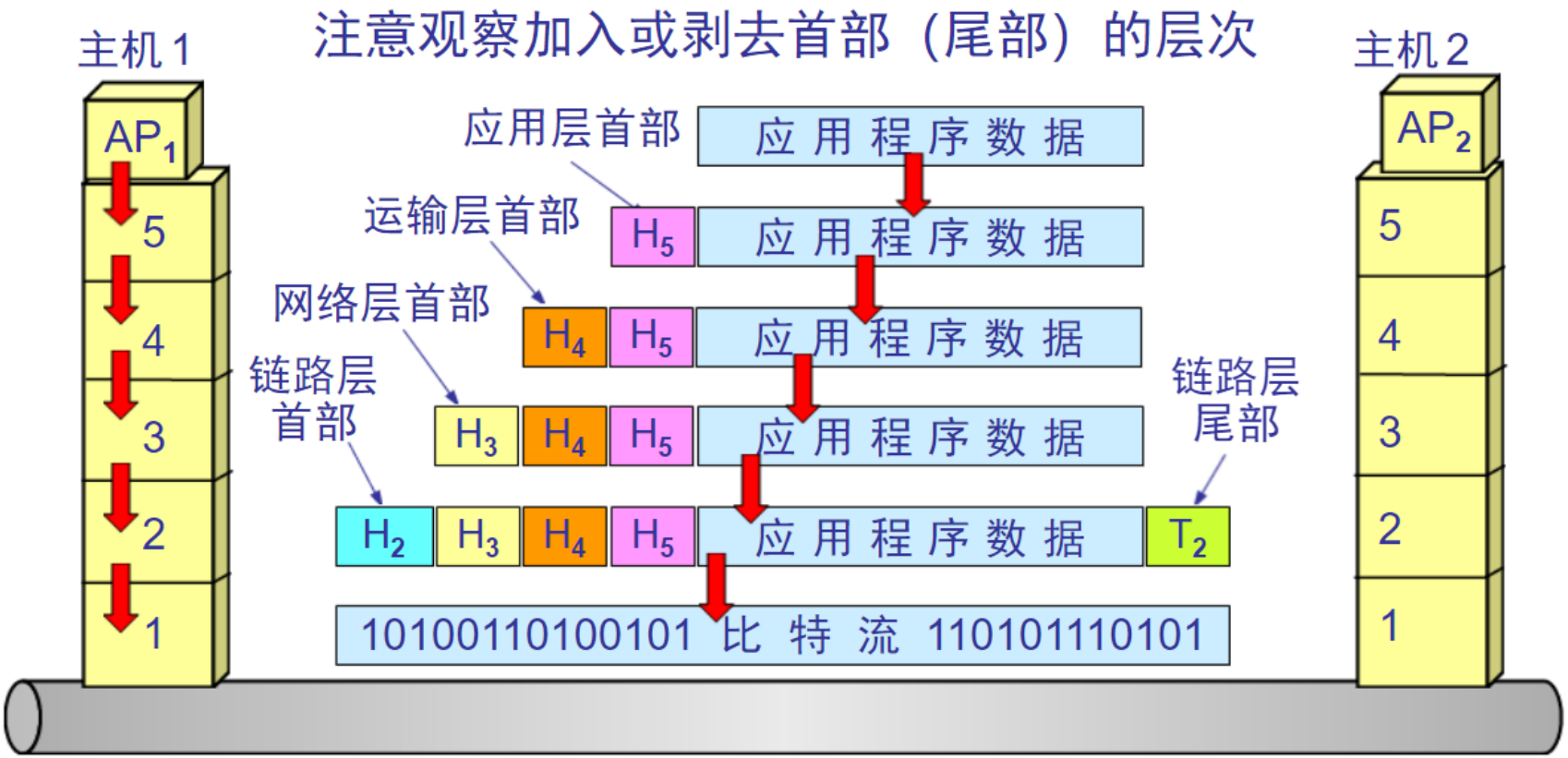
9.1 网络编程基础



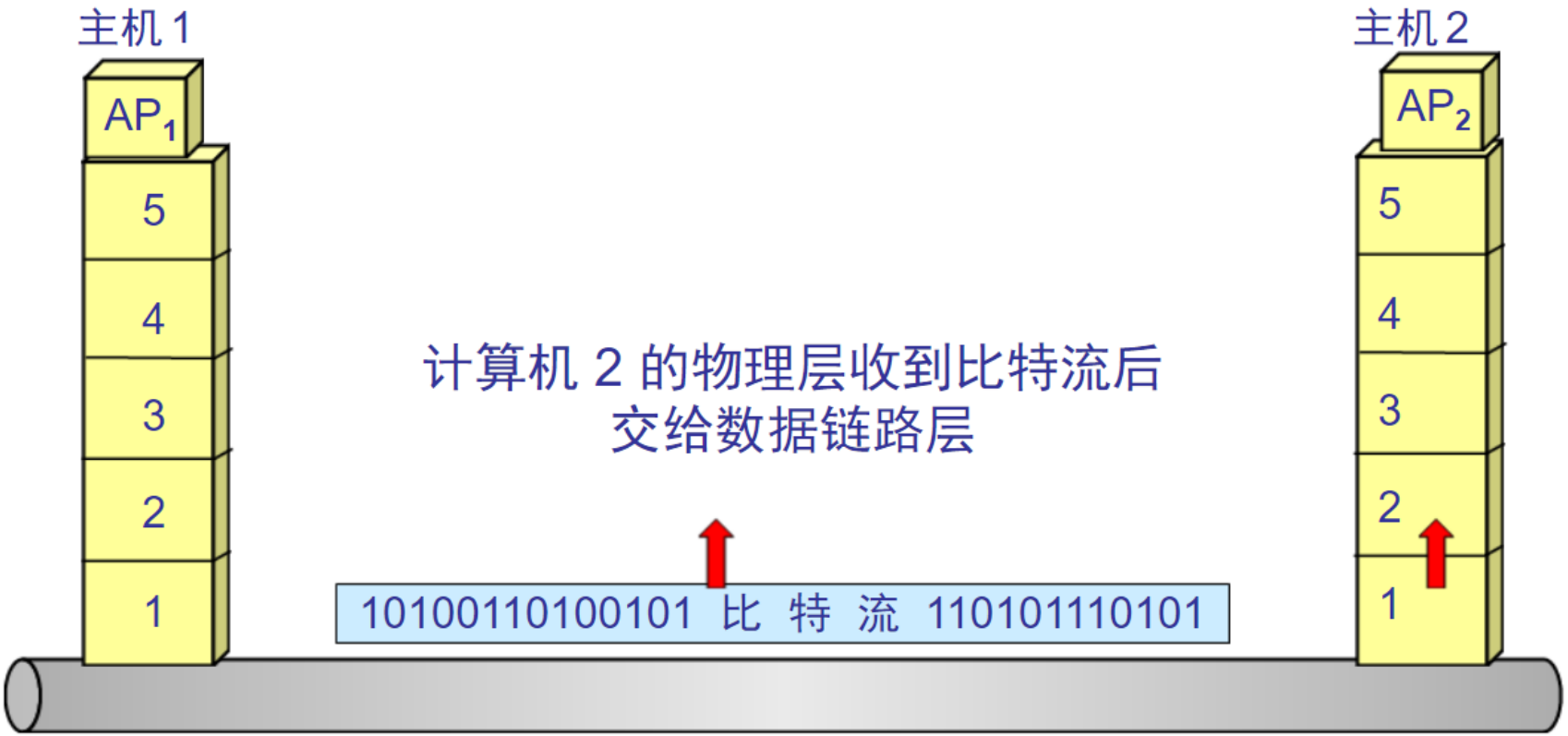
9.1 网络编程基础



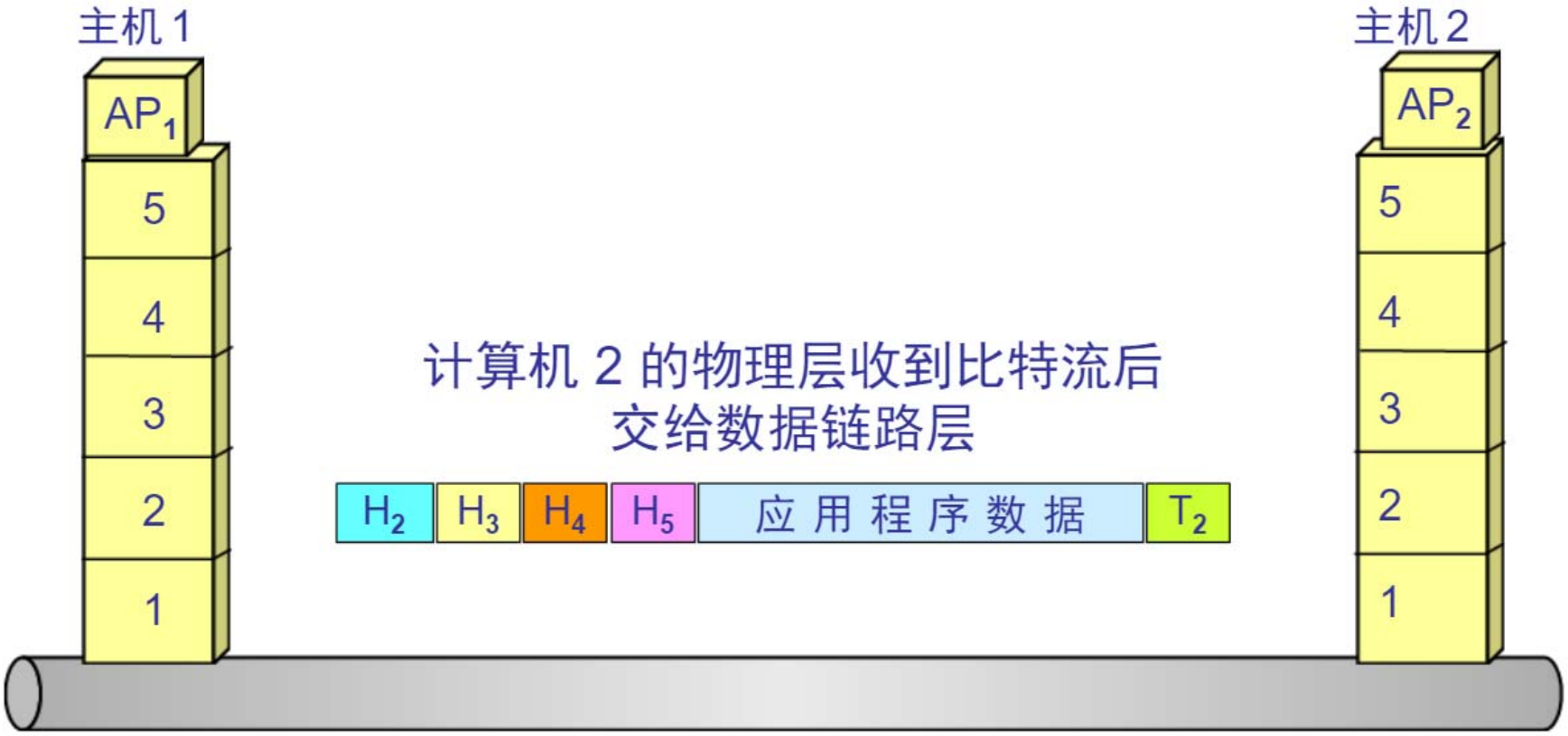
9.1 网络编程基础


注意观察加入或剥去首部（尾部）的层次
主机1
主机2
AP1
AP2
5
4
3
2
1
应用层首部
运输层首部
网络层首部
链路层首部
链路层尾部
H5
H4
H3
H2
T2
应 用 程 序 数 据
101001101001 01 比 特 流 110101110101

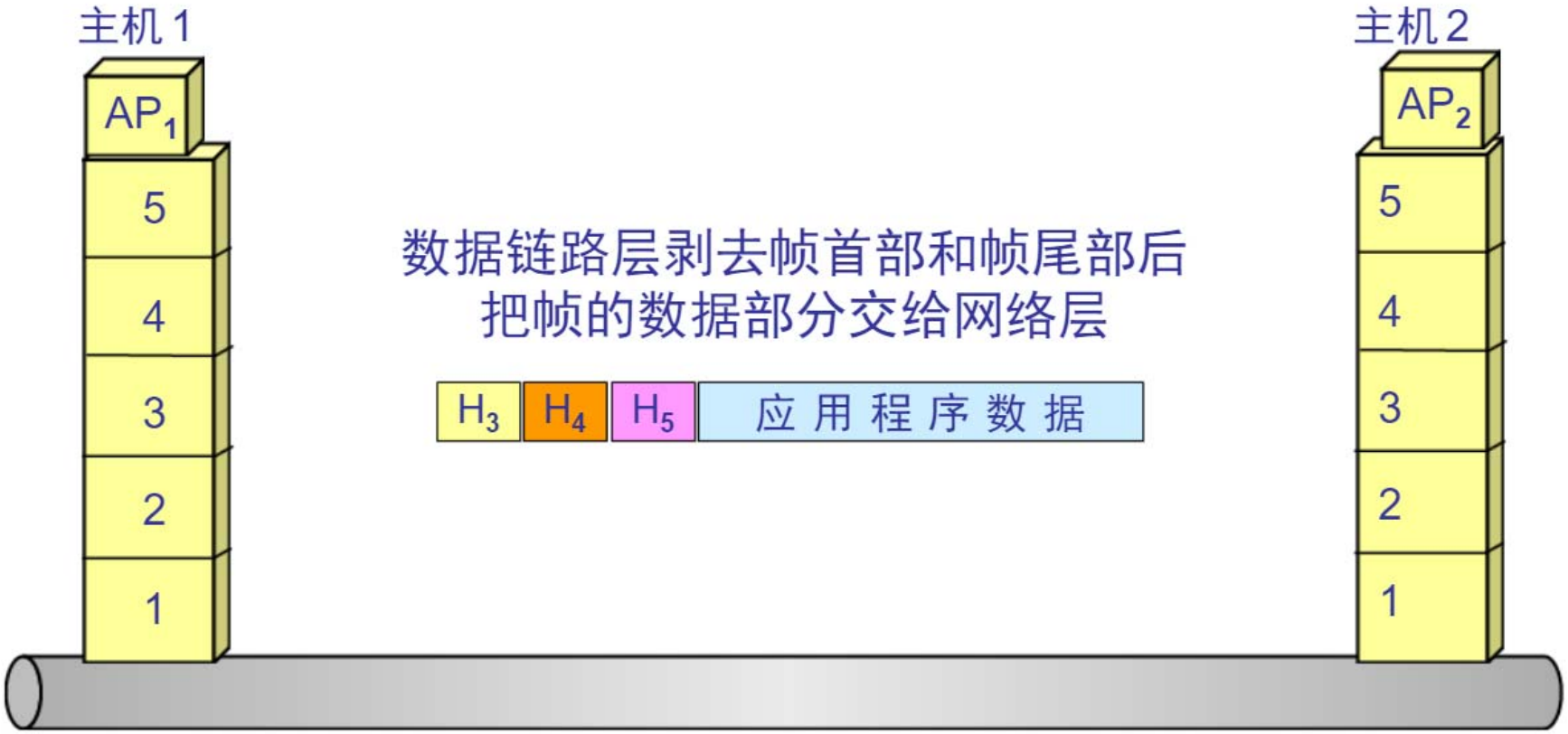
9.1 网络编程基础


主机 1
AP1
5
4
3
2
1
主机 2
AP2
5
4
3
2
1
计算机 2 的物理层收到比特流后
交给数据链路层
101001101001 01 比 特 流 110101110101

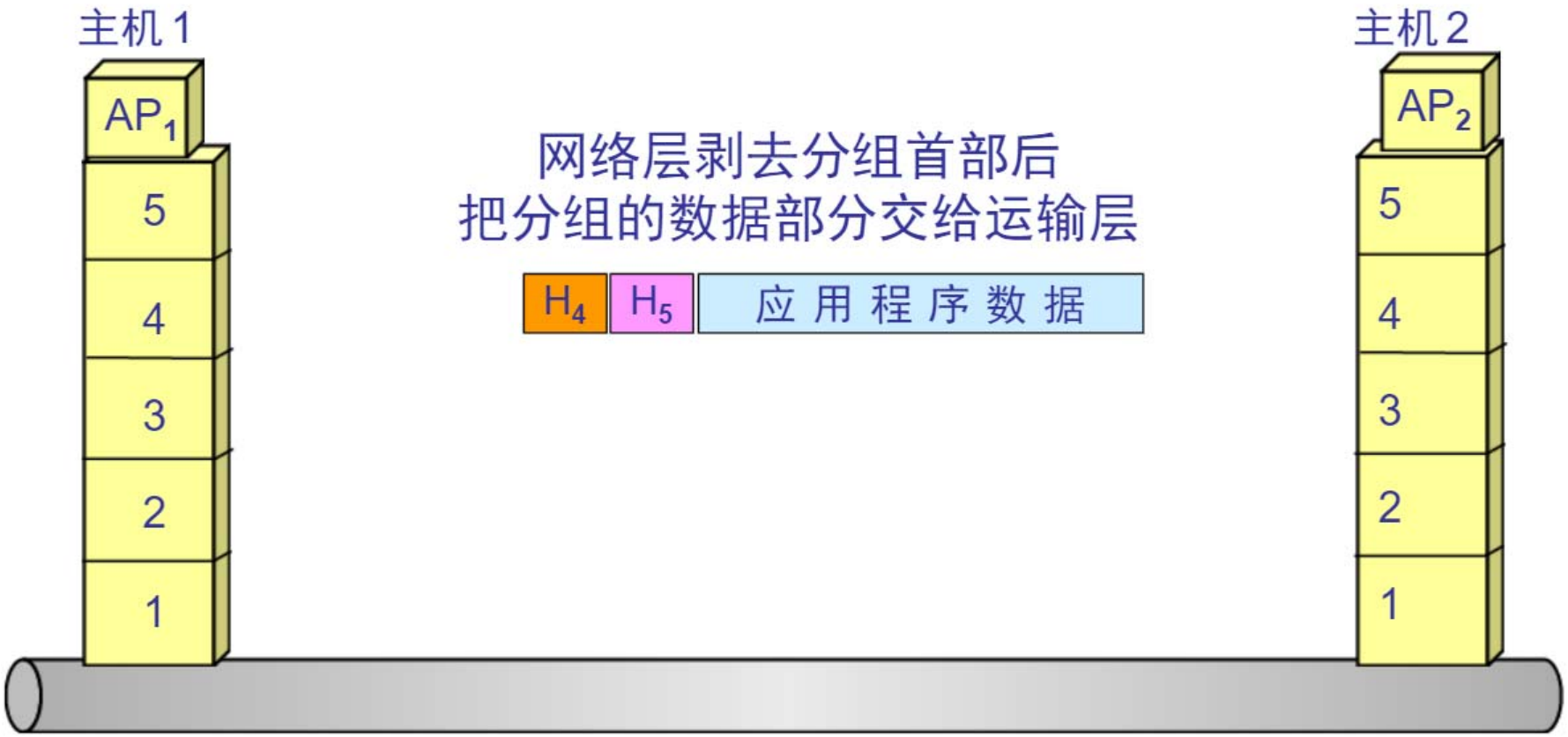
9.1 网络编程基础


主机 1
AP1
5
4
3
2
1
主机 2
AP2
5
4
3
2
1
计算机 2 的物理层收到比特流后
交给数据链路层
H2
H3
H4
H5
应 用 程 序 数 据
T2

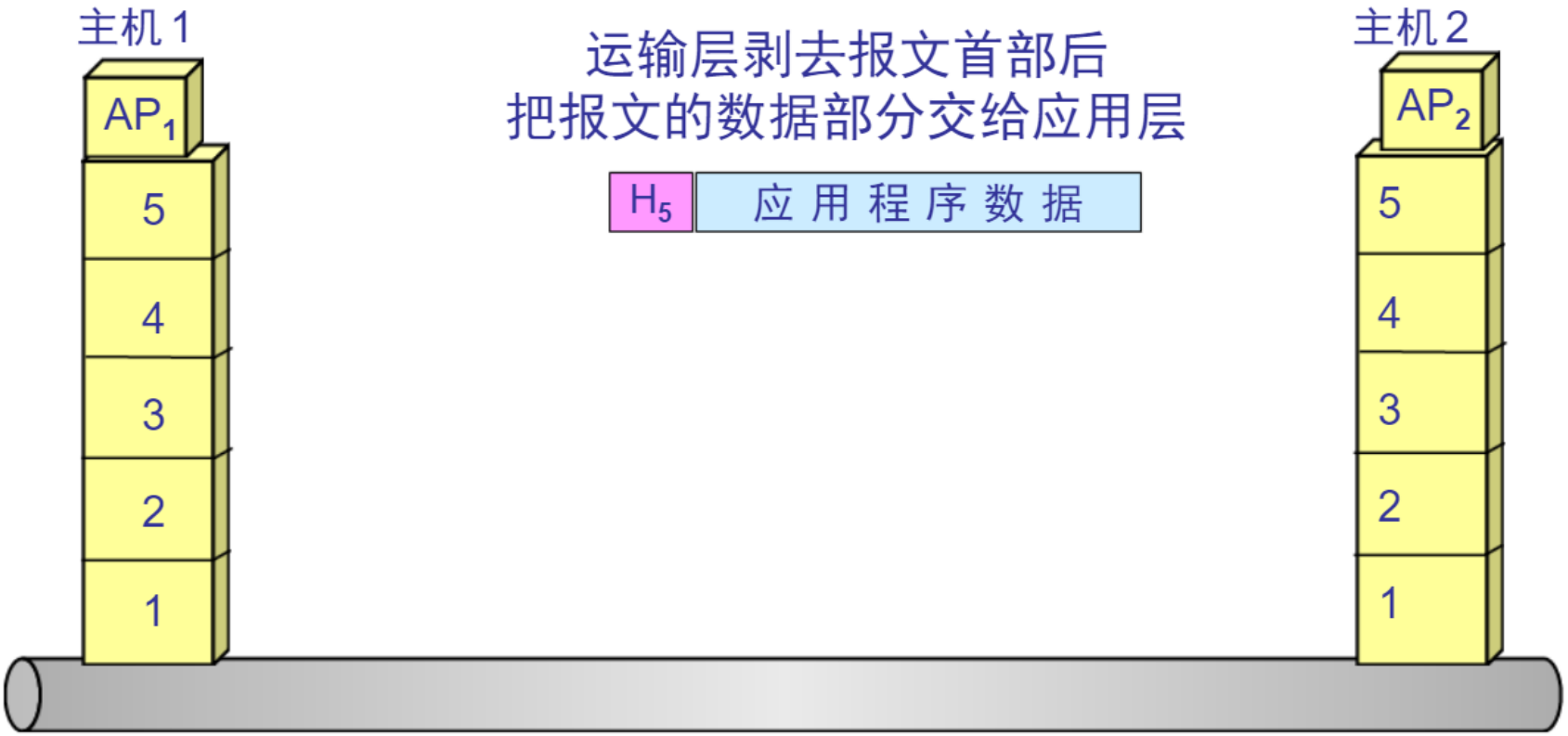
9.1 网络编程基础


主机1
AP1
5
4
3
2
1
主机2
AP2
5
4
3
2
1
数据链路层剥去帧首部和帧尾部后
把帧的数据部分交给网络层
H3
H4
H5
应用程序数据

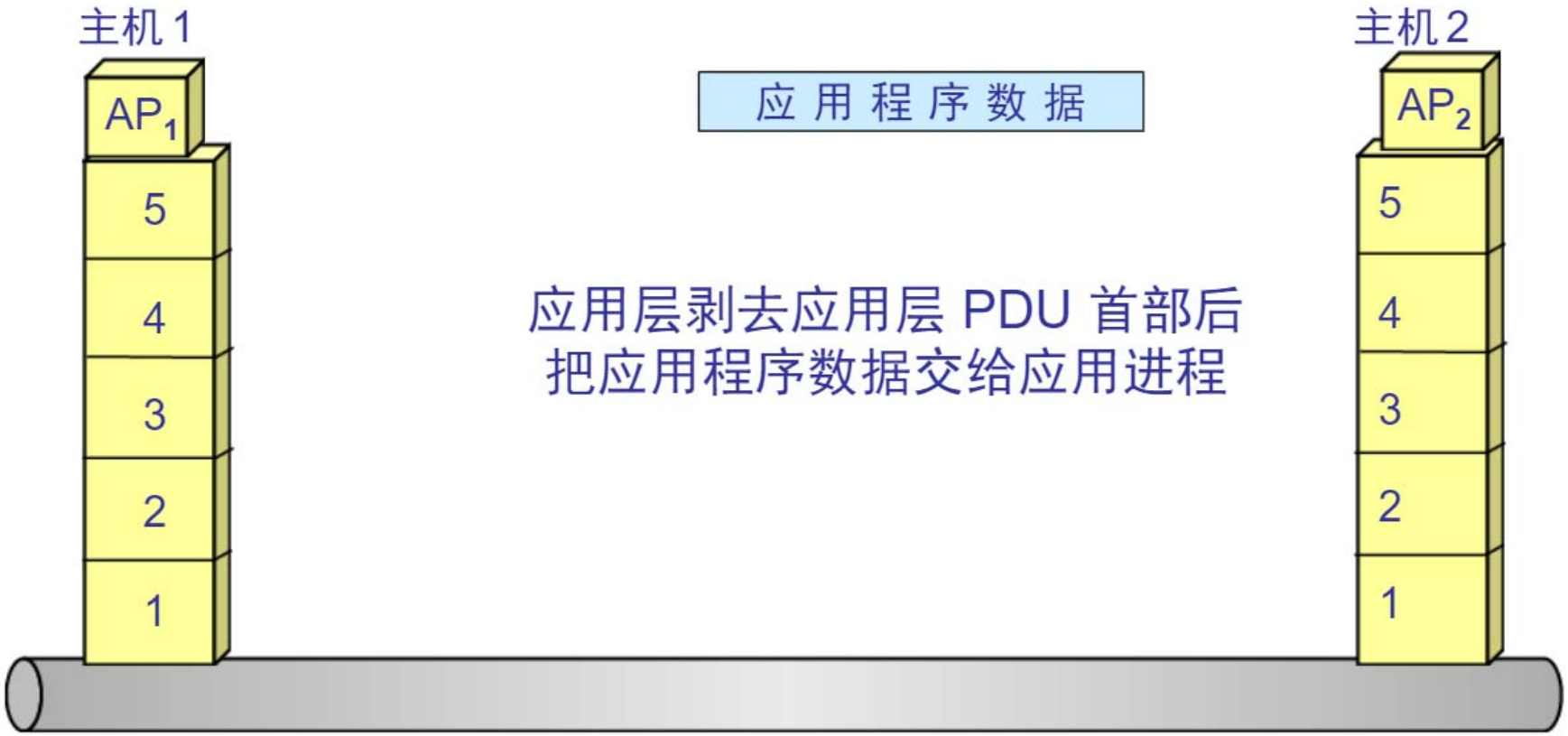
9.1 网络编程基础


主机 1
AP1
5
4
3
2
1
网络层剥去分组首部后
把分组的数据部分交给运输层
H4
H5
应 用 程 序 数 据
主机 2
AP2
5
4
3
2
1

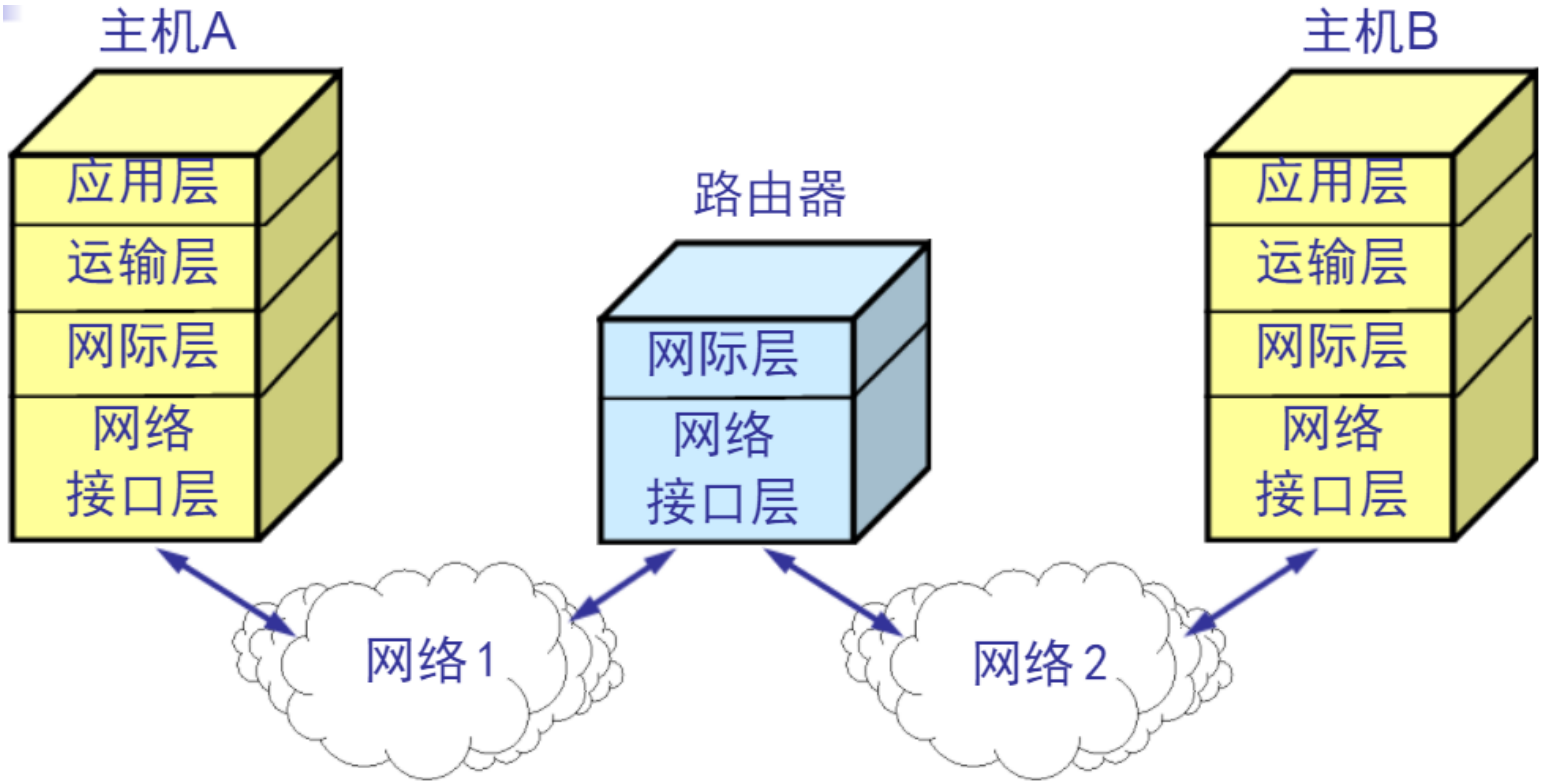
9.1 网络编程基础


主机 1
AP1
5
4
3
2
1
运输层剥去报文首部后
把报文的数据部分交给应用层
H5
应 用 程 序 数 据
主机 2
AP2
5
4
3
2
1

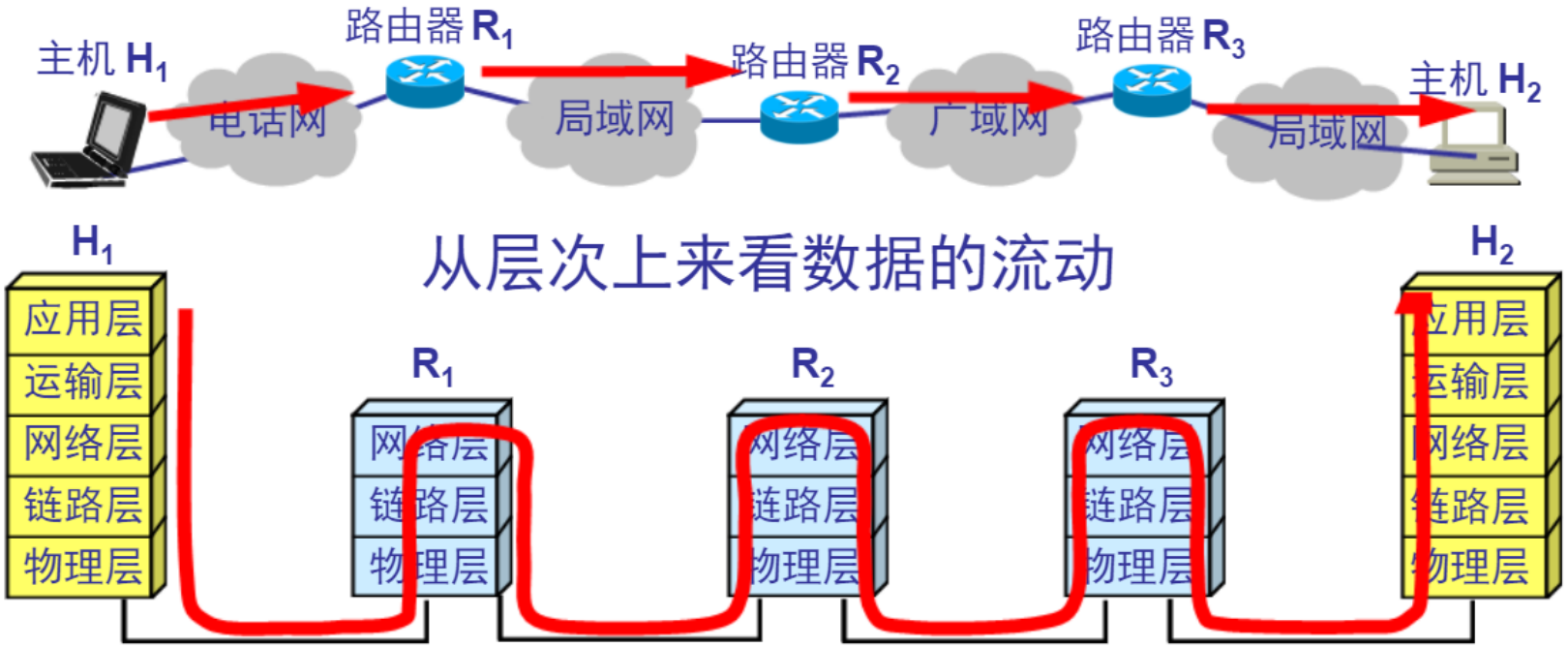
9.1 网络编程基础


主机 1
AP1
5
4
3
2
1
应 用 程 序 数 据
应用层剥去应用层 PDU 首部后
把应用程序数据交给应用进程
主机 2
AP2
5
4
3
2
1

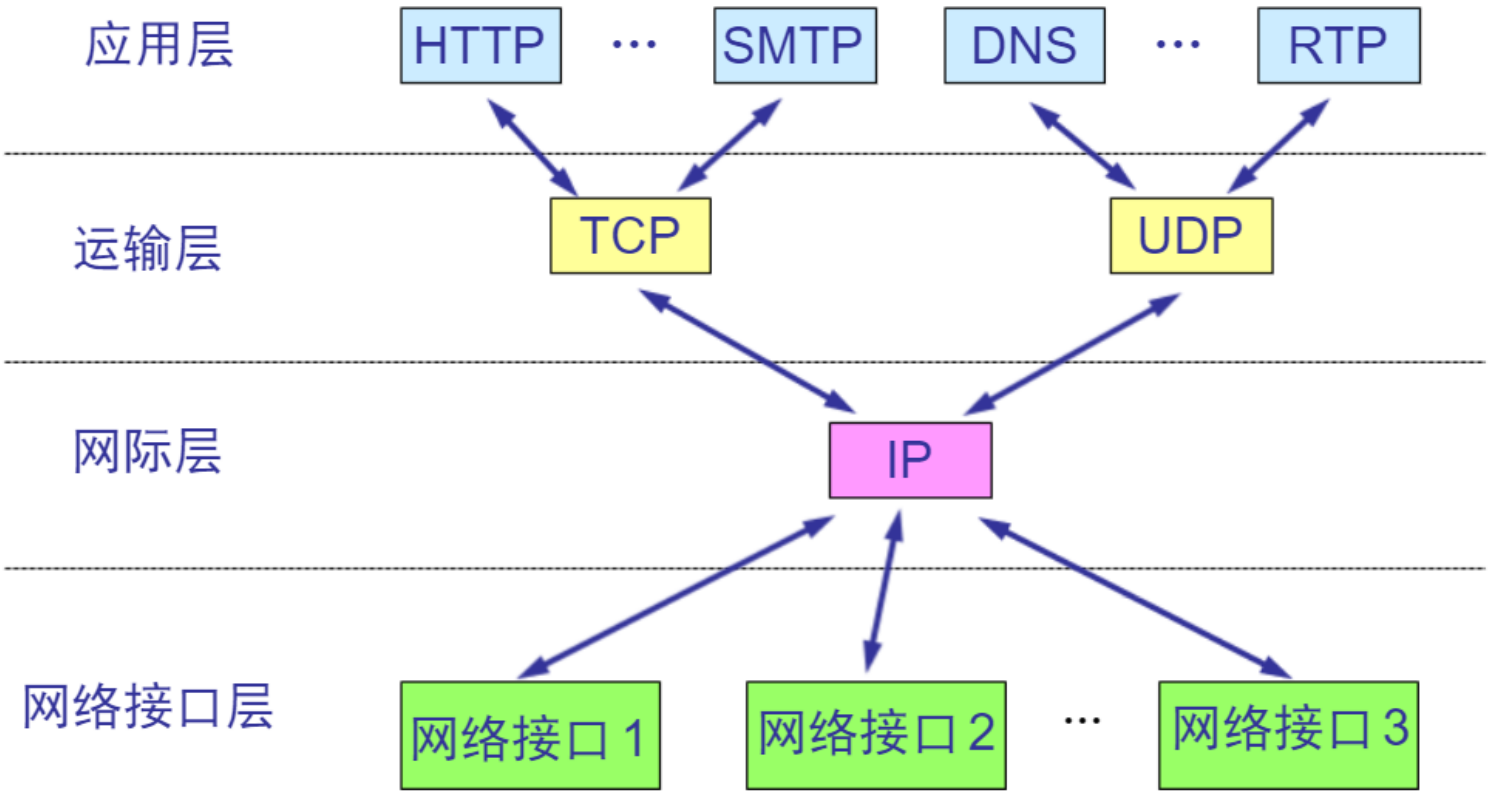
9.1 网络编程基础


主机A
应用层
运输层
网际层
网络
接口层
路由器
网际层
网络
接口层
主机B
应用层
运输层
网际层
网络
接口层
网络1
网络2

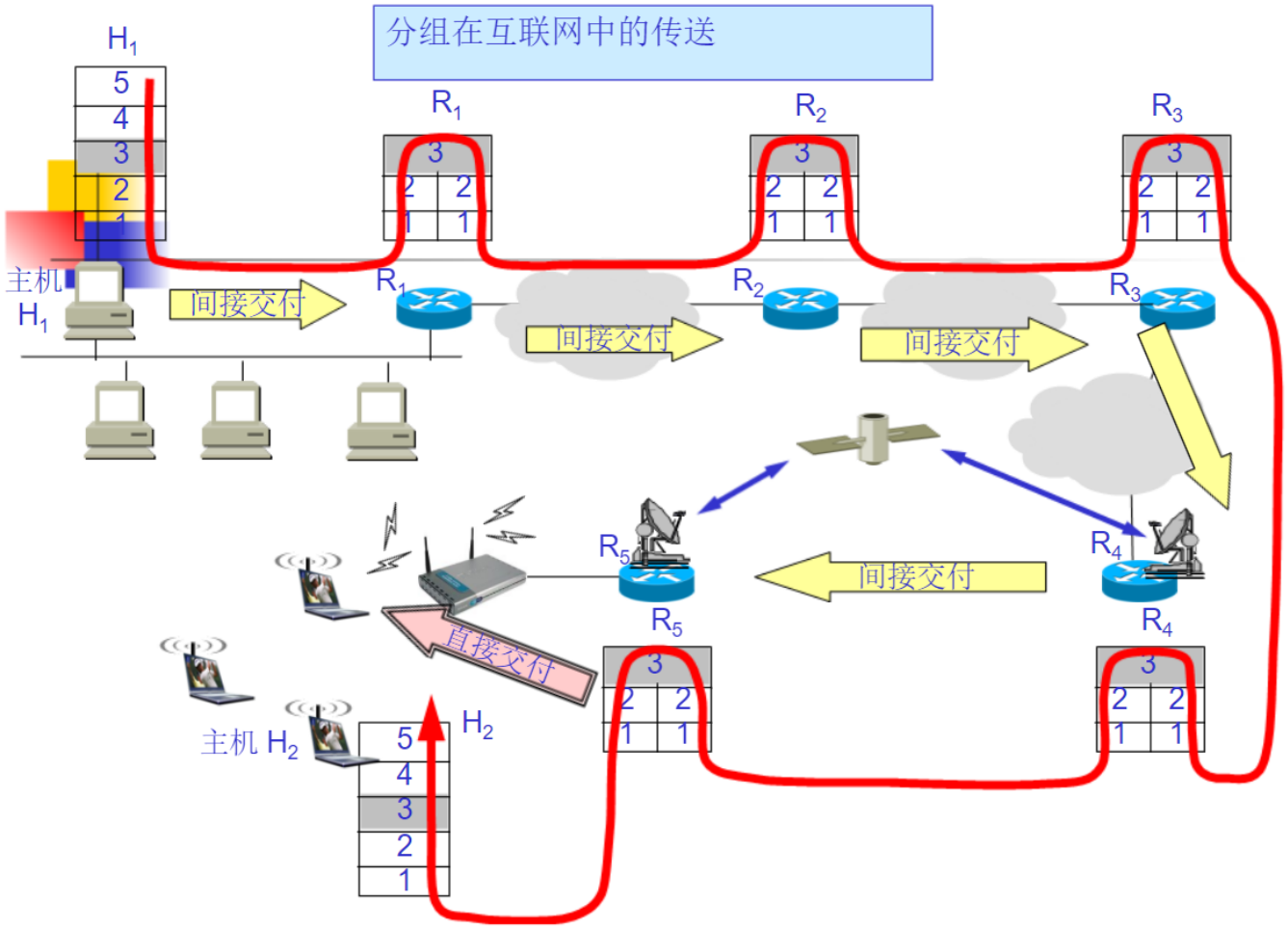
9.1 网络编程基础

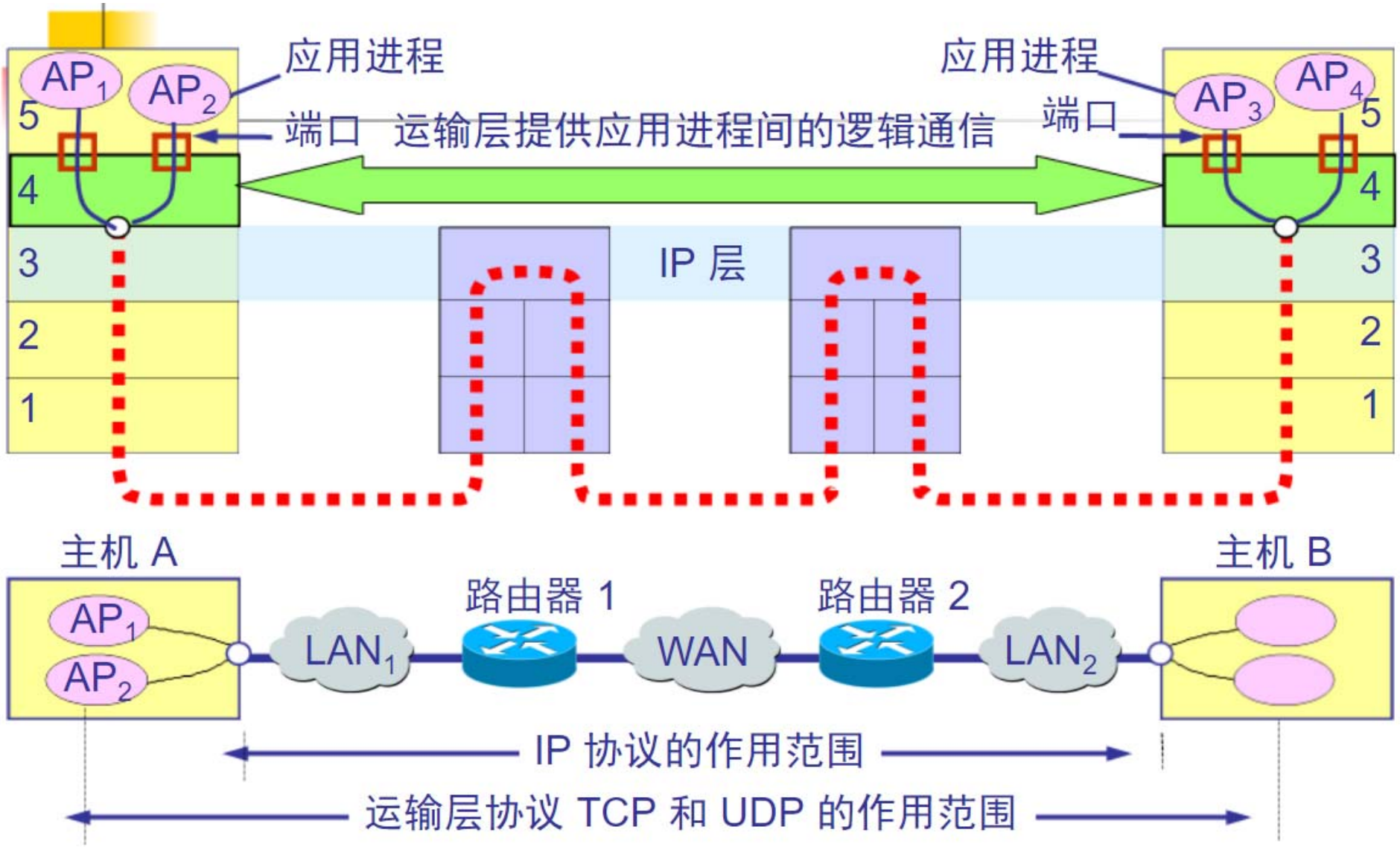

主机 H1
路由器 R1
路由器 R2
路由器 R3
主机 H2
电话网
局域网
广域网
局域网
从层次上来看数据的流动
H1
R1
R2
R3
H2
应用层
运输层
网络层
链路层
物理层

9.1 网络编程基础

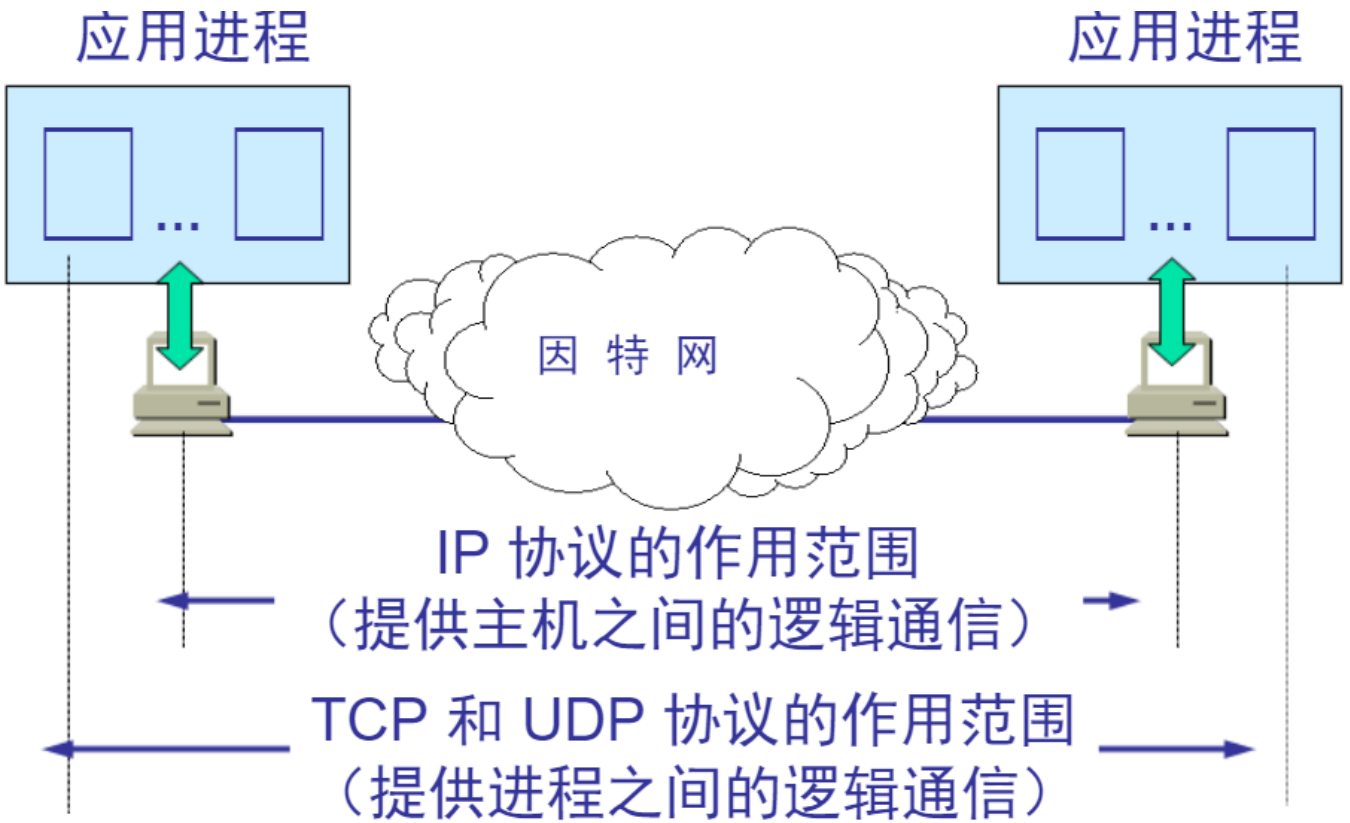

应用层
HTTP
…
SMTP
DNS
…
RTP
运输层
TCP
UDP
网际层
IP
网络接口层
网络接口 1
网络接口 2
…
网络接口 3

9.1 网络编程基础

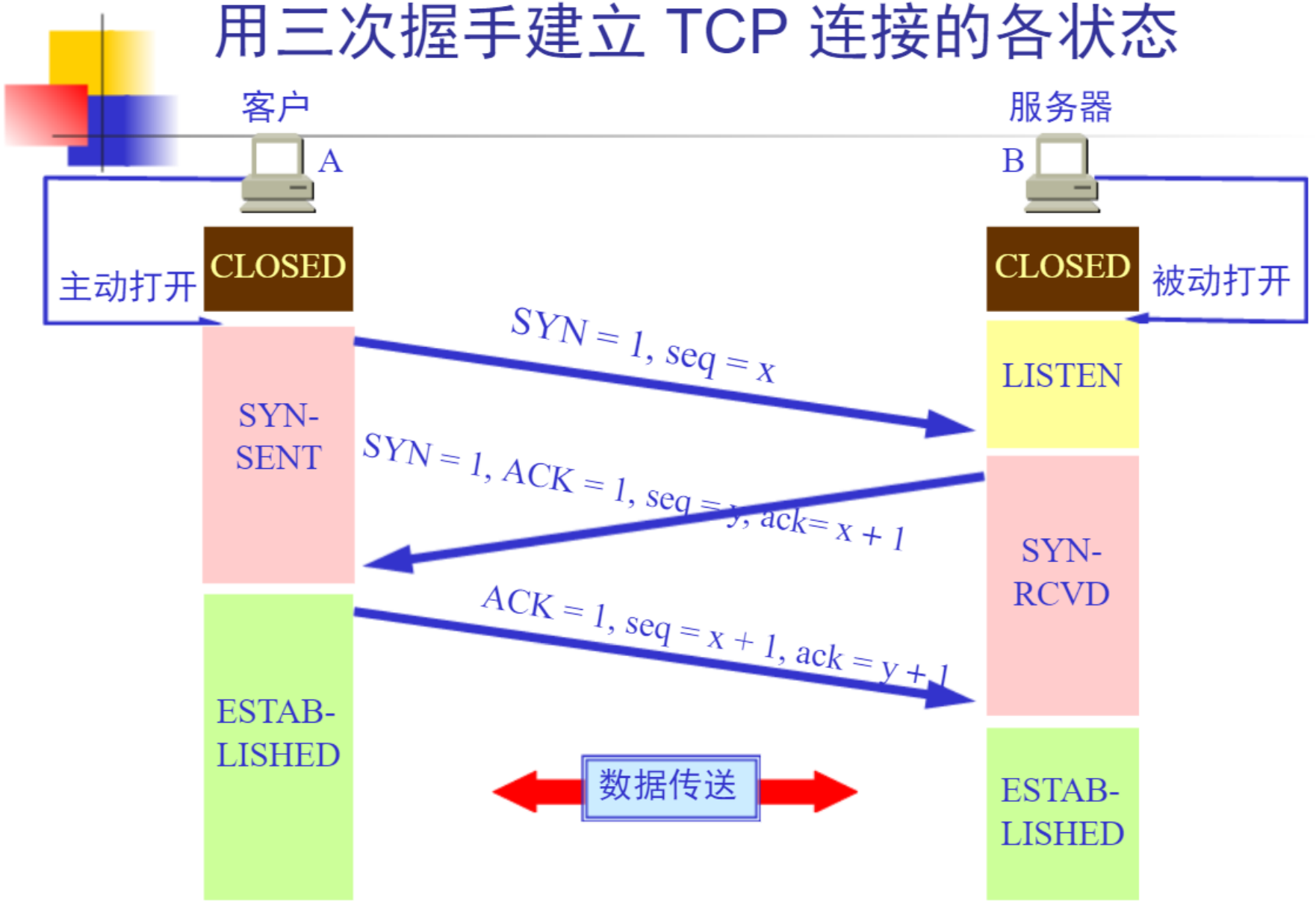

分组在互联网中的传送
H1
5
4
3
2
1
主机
H1
R1
R2
R3
间接交付
间接交付
间接交付
R5
R4
间接交付
直接交付
主机 H2
H2

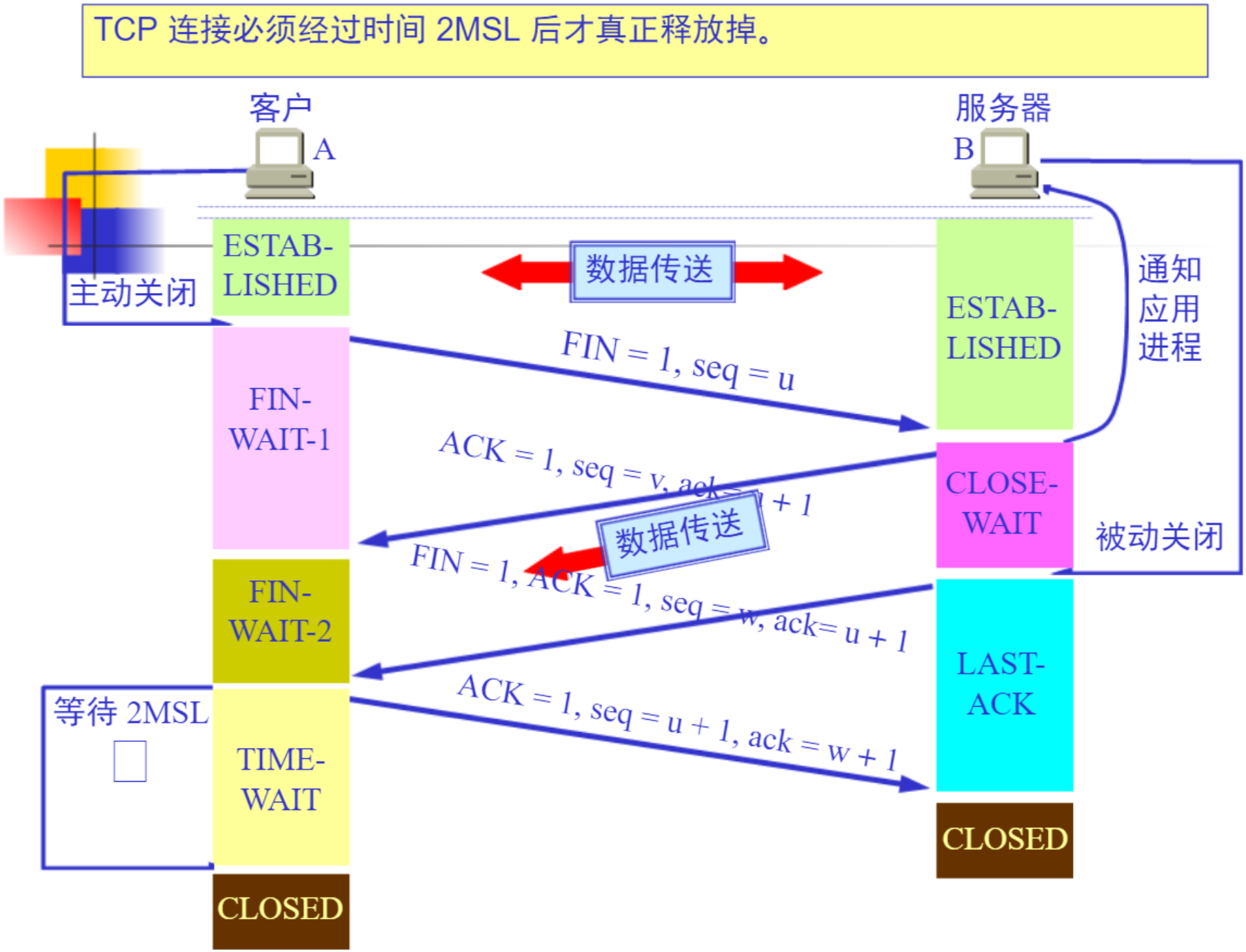

应用进程
应用进程
AP1
AP2
AP3
AP4
端口
运输层提供应用进程间的逻辑通信
端口
5
4
3
2
1
IP 层
主机 A
主机 B
路由器 1
路由器 2
LAN1
WAN
LAN2
IP 协议的作用范围
运输层协议 TCP 和 UDP 的作用范围

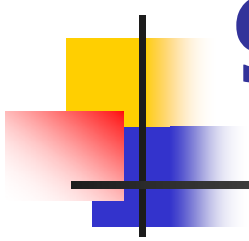
9.1 网络编程基础


应用进程
应用进程
因 特 网
IP 协议的作用范围
（提供主机之间的逻辑通信）
TCP 和 UDP 协议的作用范围
（提供进程之间的逻辑通信）

用三次握手建立 TCP 连接的各状态




TCP 连接必须经过时间 2MSL 后才真正释放掉。
客户
A
服务器
B
ESTAB-
LISHED
数据传送
ESTAB-
LISHED
主动关闭
通知
应用
进程
FIN = 1, seq = u
FIN-
WAIT-1
ACK = 1, seq = v, ack = u + 1
CLOSE-
WAIT
数据传送
被动关闭
FIN = 1, ACK = 1, seq = w, ack= u + 1
FIN-
WAIT-2
LAST-
ACK
等待 2MSL
ACK = 1, seq = u + 1, ack = w + 1
TIME-
WAIT
CLOSED
CLOSED



9.1 网络编程基础

课堂思考：在浏览器地址栏输入校园网地址后，浏览器是怎样把校园网主页呈现给我们的？

9.1 网络编程基础

（1）客户端浏览器向学校的DNS请求解析www.whut.edu.cn的IP地址；

（2）DNS解析出武汉理工大学服务器的IP地址；

（3）客户端浏览器与服务器Web进程通过三次握手机制建立TCP连接；

（4）客户端浏览器发送取文件命令（GET /index.html）；

（5）服务器Web进程给出响应，把首页资源文件index.html发送给浏览器；

（6）通过四次握手机制，TCP连接被释放；

（7）浏览器解析并显示index.html超文本文档文件中的所有内容。

9.2 使用URL访问网络资源

9.2.1 URL和IP地址

1. URL类

```
public final class URL implements java.io.Serializable
{
    public URL(String protocol, String host, int port, String file)
            throws MalformedURLException
     public String toString()                    //返回完整URL地址字符串
     public String getProtocol()                 //返回协议名
     public int getPort()                        //返回端口
     public int getDefaultPort()                 //返回默认端口
     public String getHost()                     //返回主机名
     public String getFile()                     //返回完整文件名
     public final InputStream openStream() throws
            java.io.IOException                  //使用流获得URL资源内容
}
URL url2 = new URL("http://www.edu.cn");
```

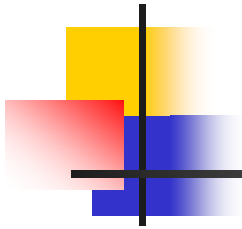
2. URLConnection类

1. URLConnection类声明

```
public abstract class URLConnection
{
    public URL getURL()                  //返回当前连接的URL对象
    public int getContentLength()   //返回资源文件的长度
    public String getContentType() //返回资源文件的类型
    public long getLastModified()   //返回资源文件的最后修改日期
}
```

2. 使用URL类的openConnection()方法创建一个URLConnection对象

```
Public URLConnection openConnection()  throws
      java.io.IOException
```



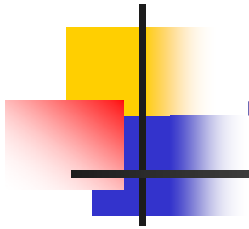
3. InetAddress类

```
public class InetAddress extends Object implements
      Serializable
{
   public static InetAddress getByName(String host)
     throws UnknownHostException
   public static InetAddress getByAddress(String host,
     byte[] addr) throws UnknownHostException
   public static InetAddress getLocalHost() throws
     UnknownHostException          //返回本地主机
   public String getHostAddress()   //返回IP地址字符串
   public String getHostName()      //返回主机名
}
```

9.2.2 使用选项卡窗格和编辑器窗格

1. 选项卡窗格

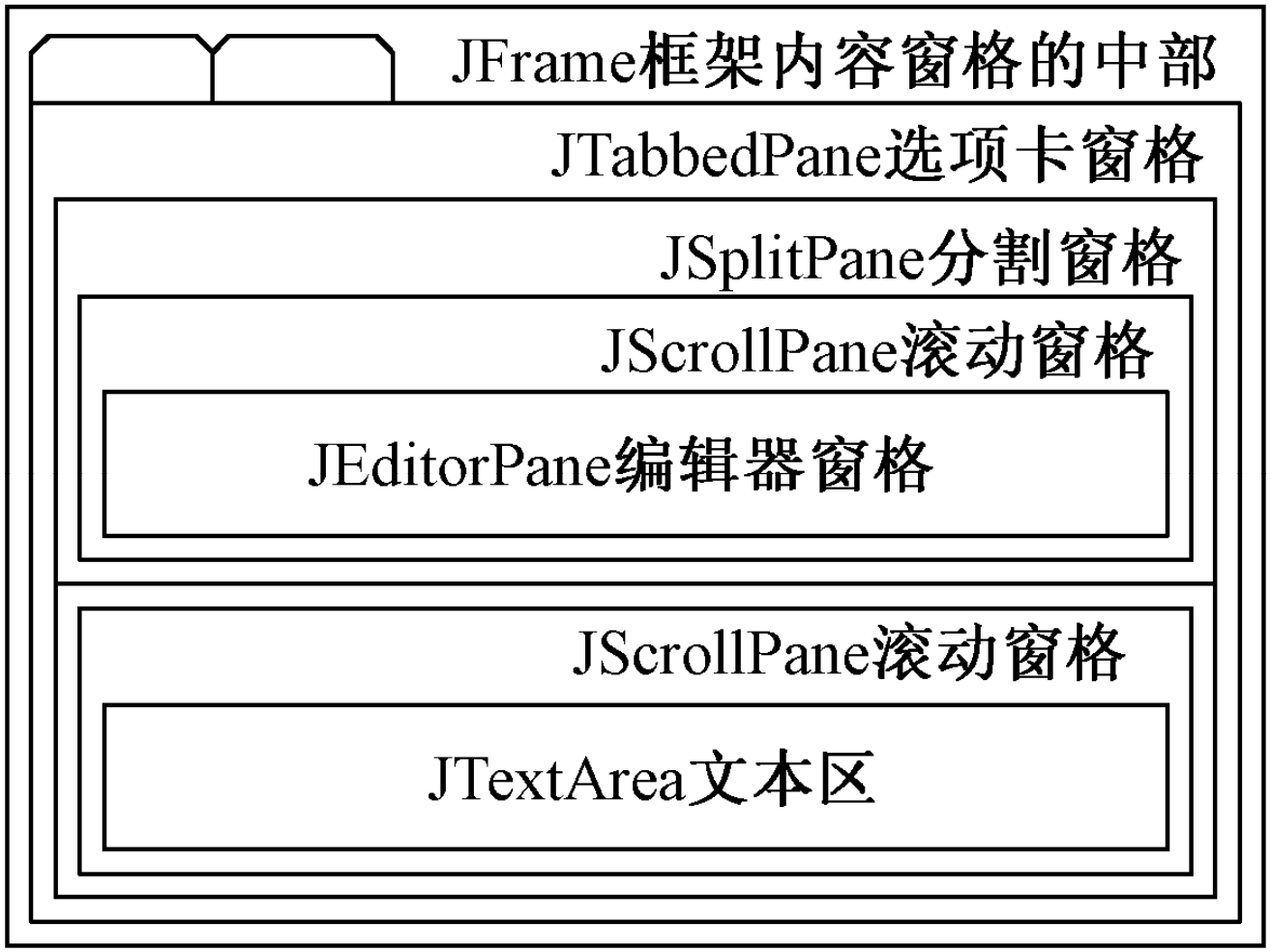
```
public class JTabbedPane extends JComponent implements
      Serializable, Accessible, SwingConstants
{
   public JTabbedPane()                                    //构造方法
   public void addTab(String title, Component comp) //添加页
   public int getTabCount()                                //返回页数
   public int getSelectedIndex()                          //当前选中页序号
   public void setSelectedIndex(int index)       //选中第index页
   public void addChangeListener(ChangeListener l)
                                                  //注册选择事件监听器
}
```

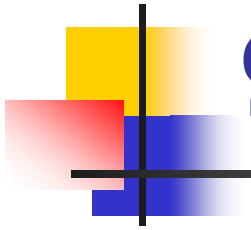


2. 编辑器窗格

```
public class JEditorPane extends
    JTextComponent
{
   public JEditorPane()                  //构造方法
   public JEditorPane(URL url) throws
          IOException     //指定初始页的URL
   public JEditorPane(String url) throws
          IOException
}
```

【例9.1】查看指定URL的Web页编辑器及HTML文档。





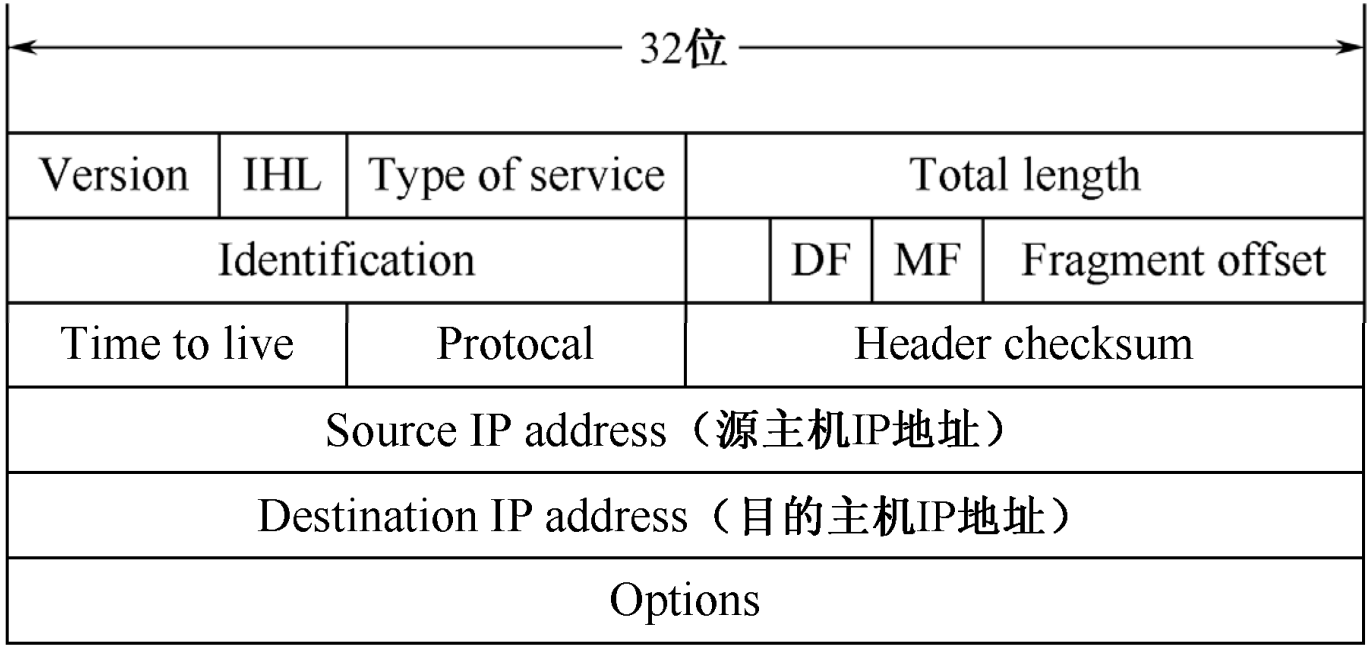
9.3 TCP Socket通信

1. 9.3.1 TCP Socket通信原理
2. 9.3.2 Java的Socket通信

9.3.1 TCP Socket通信原理

1. IP协议

- **IPv4数据报头格式**

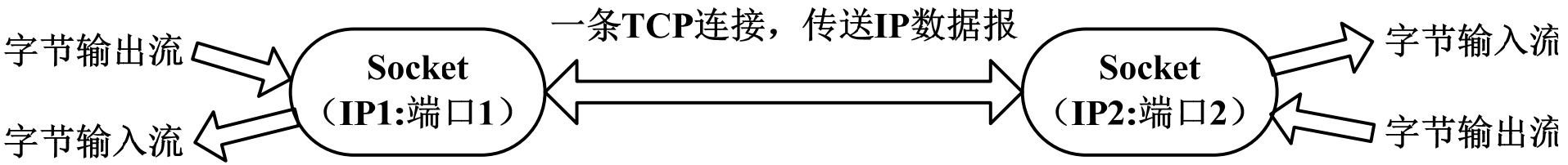


2. 传输层协议

- 用户数据报协议（**UDP**）
- 传输控制协议（**TCP**）
  - 通过端口指定服务

3. 基于**TCP**连接的**Socket**通信

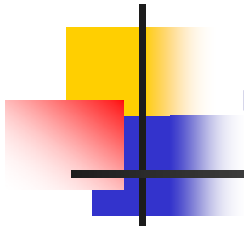
Socket：**IP**地址和端口，套接字



9.3.2 Java的TCP Socket通信

1. ServerSocket类和Socket类

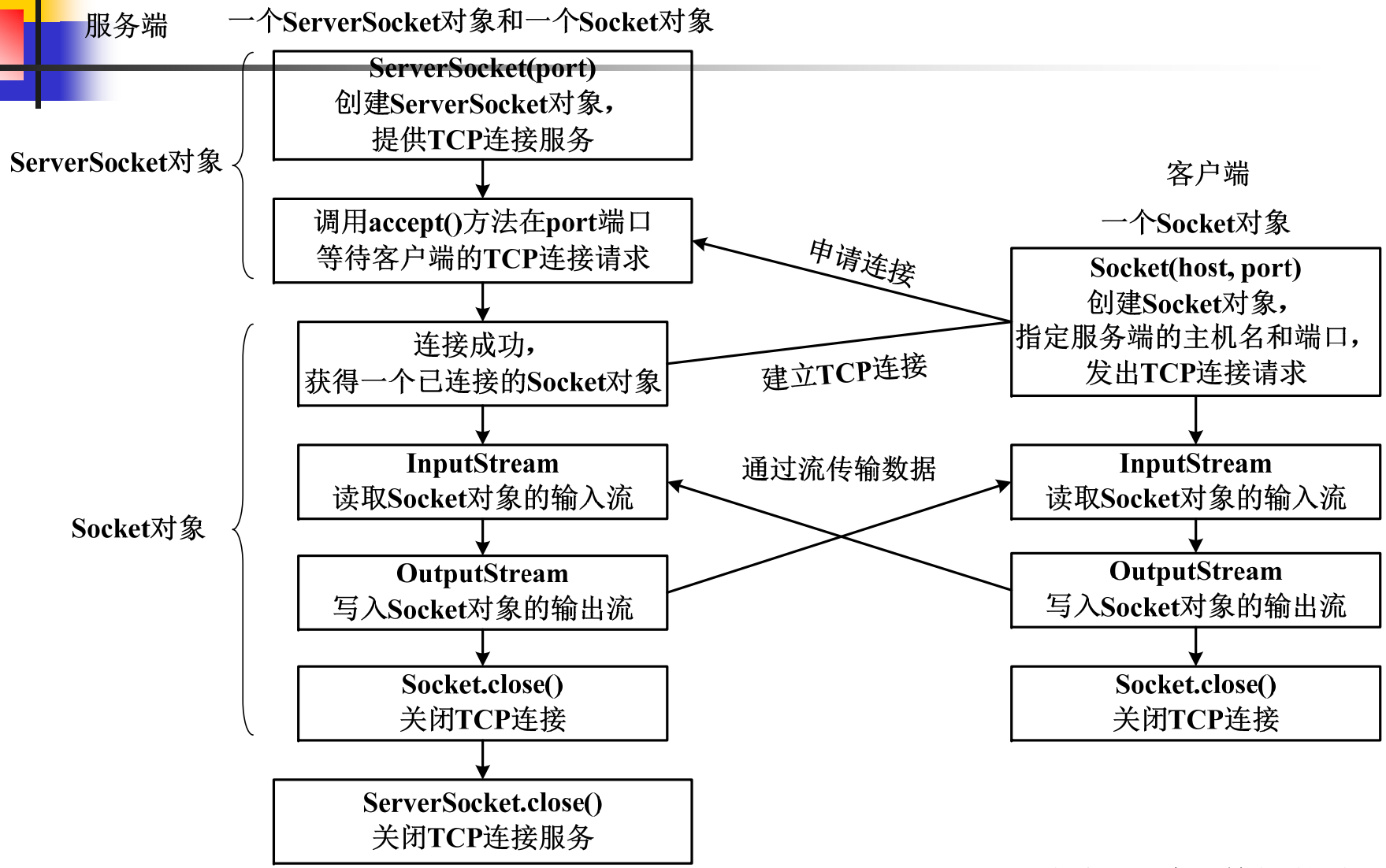
```
public class ServerSocket
{
    public ServerSocket(int port) throws
        IOException     //构造方法，指定端口号
    public Socket accept() throws IOException
        //等待接收客户端的连接请求，连接成功后返回一个已连接的Socket对象
    public void close() throws IOException
        //停止等候客户端的连接请求
}
```



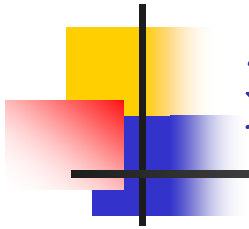
Socket类

```
public class Socket
{
    public Socket(String host, int port) throws
      UnknownHostException, IOException
                                    //构造方法，指定主机名和端口号
    public InputStream getInputStream() throws
      IOException          //返回TCP连接提供的字节输入流
    public OutputStream getOutputStream() throws
      IOException          //返回TCP连接提供有字节输出流
    public synchronized void close() throws
      IOException          //关闭TCP连接
}
```

2. Java的Socket通信流程



《Java程序设计实用教程(第4版)》



【例9.2】 采用TCP Socket通信实现的点对点聊天。

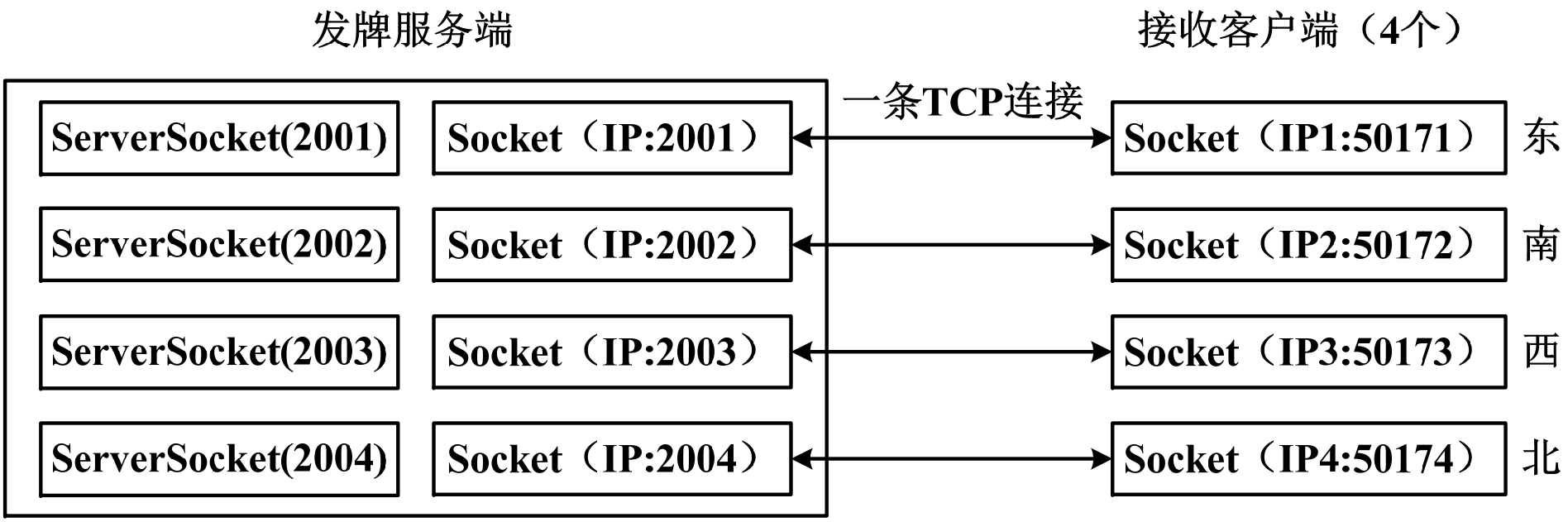
1. 服务端程序
2. 客户端程序以及聊天室的图形用户界面

3. 提供多客户的Socket通信服务

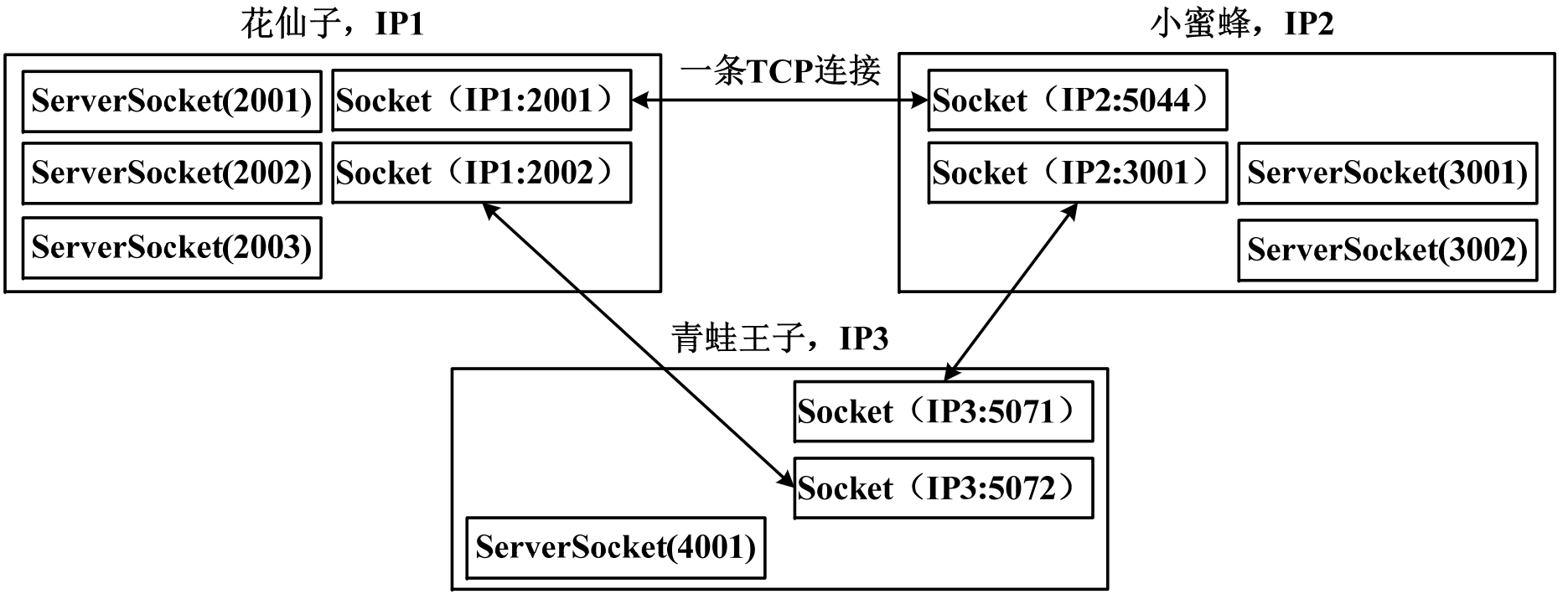
【例9.3】 网络发牌程序。

① 发牌服务端

② 接收客户端



【例9.4】多客户的TCP Socket通信。



9.4 UDP数据报通信

UDP（用户数据报协议）是一个无连接的协议，以数据报为单位进行数据传输。

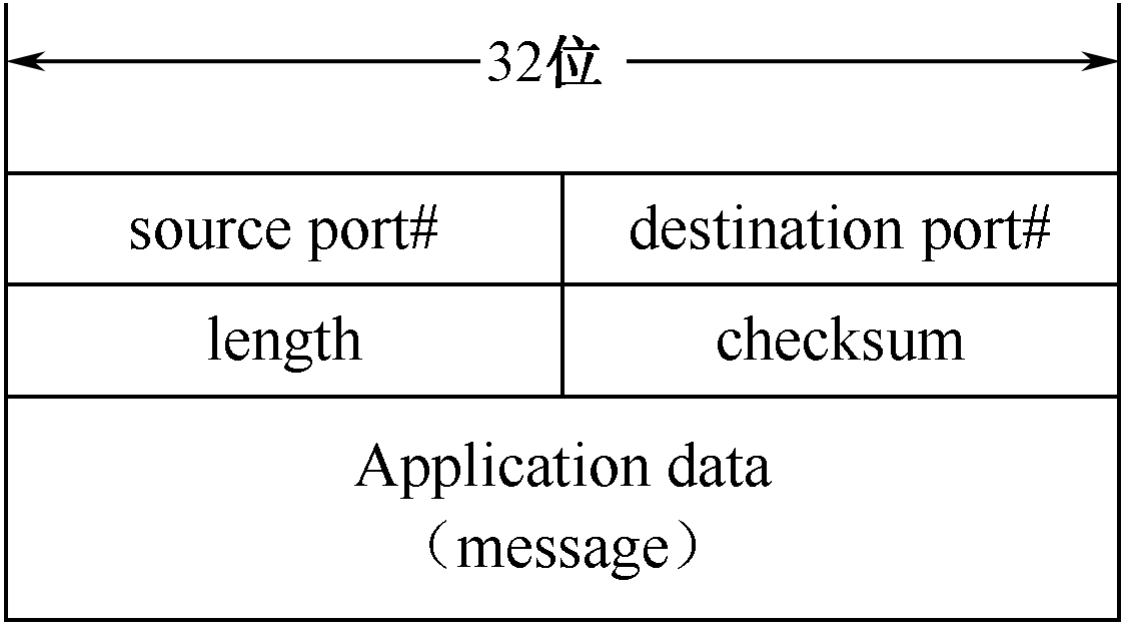


图9.10 UDP段结构

9.4.1 UDP数据报
1.数据报包

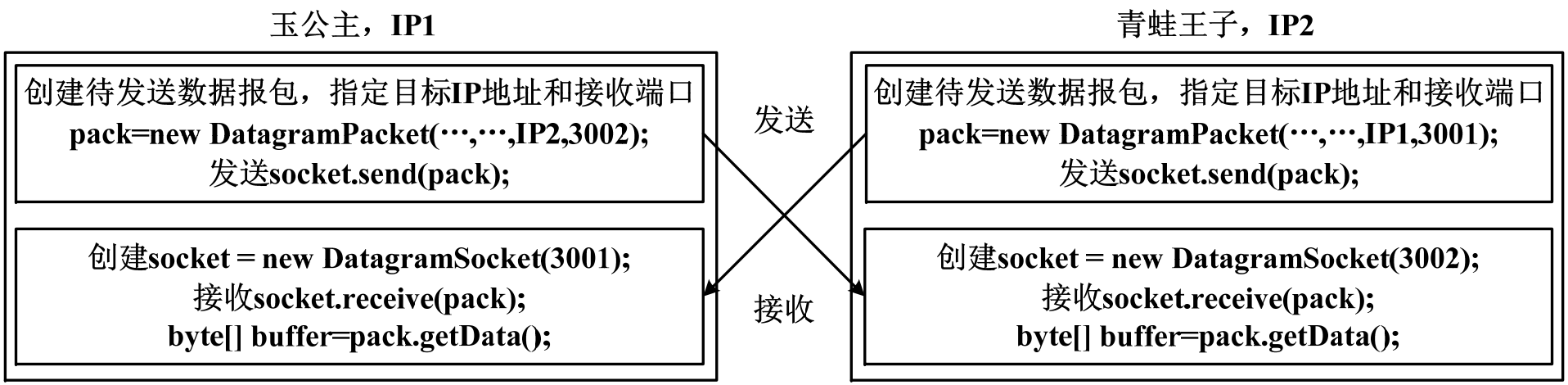
```
public final class DatagramPacket extends Object
{
   public DatagramPacket(byte[] buf, int length, InetAddress
                         address, int port)      //创建发送数据报
   public DatagramPacket(byte[] buf, int length) //创建接收数据报

   public byte[] getData()                       //从缓冲区中返回数据
   public int getLength()                        //返回数据报的长度
   public InetAddress getAddress()               //返回远程主机IP地址
   public int getPort()                          //返回远程主机的端口号
   public void setAddress(InetAddress iaddr) //发往的主机的IP地址
   public void setPort(int iport)                //发往的远程主机上的端口
}
```

2. 数据报Socket

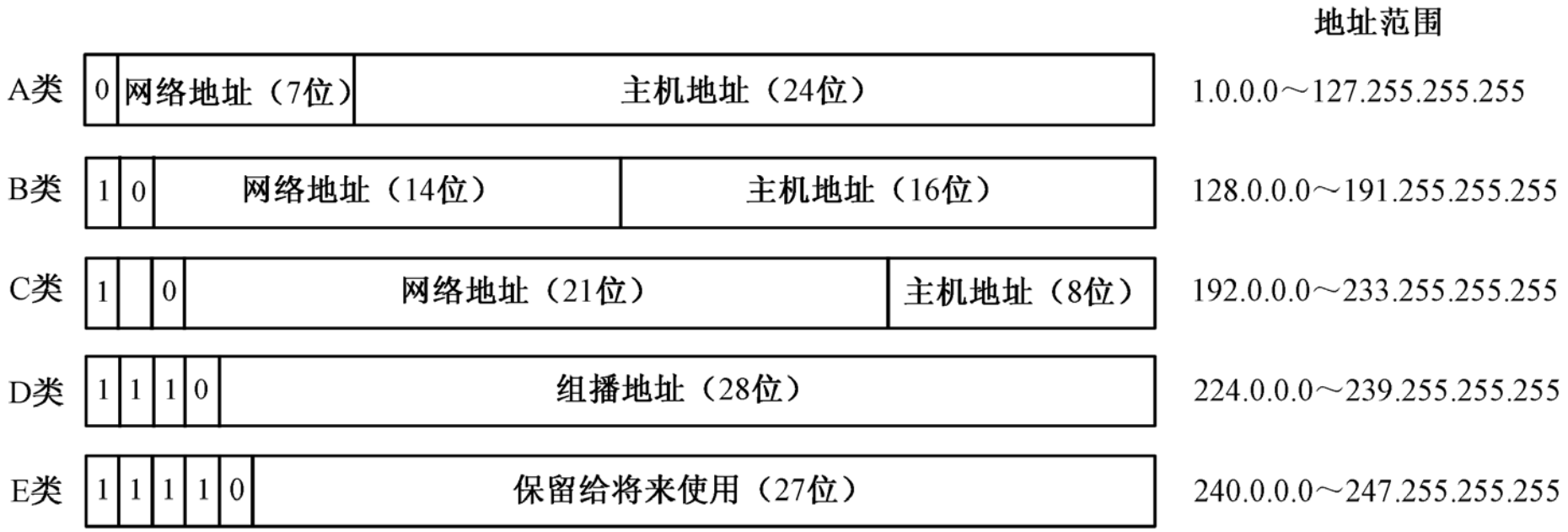
```
public class DatagramSocket extends Object
{
   public DatagramSocket() throws SocketException
                          //创建Socket，绑定一可用端口
   public DatagramSocket(int port) throws
                          SocketException    //port指定端口
   public void send(DatagramPacket pack) throws
                          IOException  //发送pack数据报包
   public void receive(DatagramPacket pack) throws
                       IOException//接收数据报包存于pack中
   public void close()                    //关闭Socket
}
```

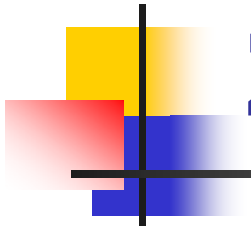
【例9.5】采用UDP数据报通信实现的点对点聊天。



9.4.2 UDP组播数据报

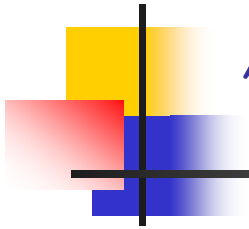
1. 组播地址





2. 组播Socket

```
public class MulticastSocket extends DatagramSocket
{
   public MulticastSocket(int port) throws IOException
                              //创建组播套接字并将其绑定到port端口
   public void joinGroup(InetAddress dip) throws
                    IOException//加入广播组，dip指定组播地址
   public void setTimeToLive(int ttl) throws
                                  IOException//设置广播范围
   public void leaveGroup(InetAddress dip) throws
                                  IOException  //离开广播组
}
```



例9.6和例9.7

【例9.6】 用于控制网络考试时间的广播。

① 提供时间组播的服务端程序

② 接收时间组播的客户端程序

【例9.7】组播聊天。

实验9 网络通信

- 目的：通过**URL**访问网络资源， **Socket**通信。
- 要求：
  ① 熟悉通过**URL**获得指定网络资源内容和文件属性；
  ② 掌握**TCP Socket**通信；
  ③ 熟悉**UDP**数据报通信和组播通信。
- 重点：**TCP Socket**， **UDP Socket**。
- 难点： **TCP Socket**， **UDP Socket**。