



## 第2章 协议体系结构

南京大学计算机系 黄皓教授

2007年9月4日 星期二



# 参考文献

1. 谢希仁，陈鸣，张兴元，计算机网络，电子工业出版社，2003年。
2. Behrouz, A, et al, 数据通信与网络，机械工业出版社，2007年7月。



# 内容

1. 协议体系结构的必要性
2. 协议体系结构
3. 开放系统互联（OSI）
4. TCP/IP 体系结构



## 2.1 协议体系结构的必要性

### ■ 一个例子:

领导决定了要起草的一个文件的主题，秘书起草了这个文件，与领导交流、讨论、定稿。

外文秘书翻译文件。

秘书将文件交给收发室。

收发室交邮局。

邮局根据地址决定运输单位。

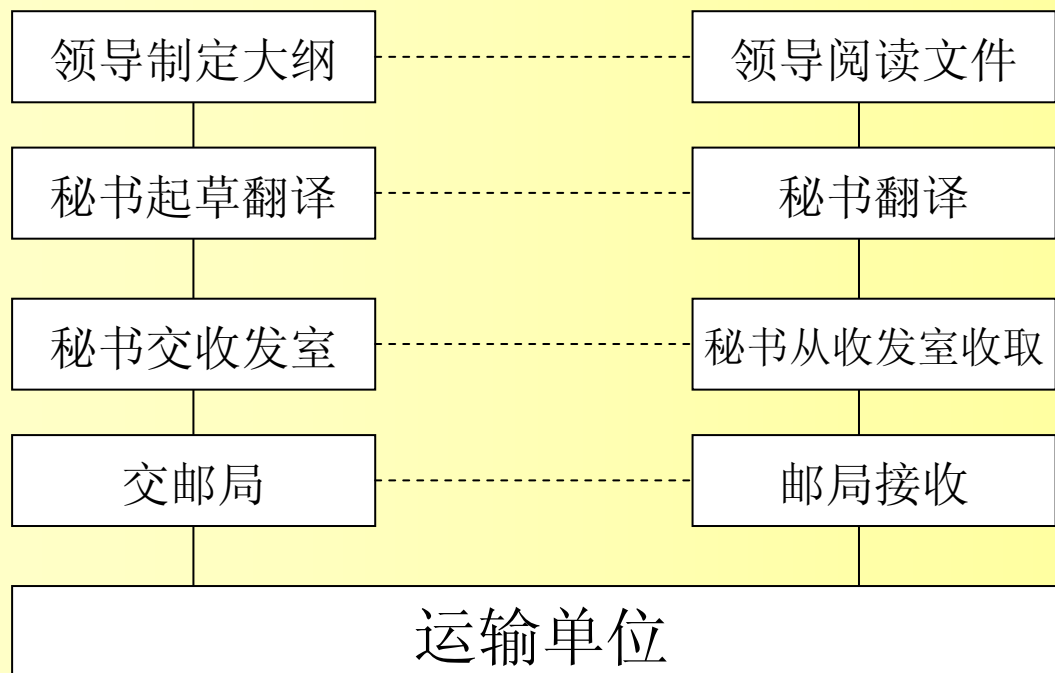
目的地邮局将文件交给接收单位收发室。

接收单位收发室将邮件交收件人或收件人外文秘书秘书。

收件人外文秘书将文件交收件人秘书或收件人。



## 例子的层次化





## 2.1 协议体系结构的必要性

### ■ 层的概念

- ☐ 发送者与接收者对应
- ☐ 秘书与秘书对应
- ☐ 发送邮局与接收邮局对应
- ☐ 中转站与中转站对应
- ☐ .....



## 2.1 协议体系结构的必要性

- 两台计算机上的进程之间通信时需要完成的工作。
  - 如果文件格式不一样，必须有一个系统执行格式转换。
  - 源点系统的应用程序必须确认终点系统的应用程序已经准备好接收文件。
  - 源点系统必须确认终点系统已经准备好接收数据。
  - 激活直接链接的数据通信通道，告诉通信网络它所期望的终点系统标识。



## 2.1 协议体系结构的必要性

- 为进行网络中的数据交换而建立的规则、标准成为**网络协议**。
- **网络协议的要素**
  - **语法**: 数据与控制信息的结构和格式
  - **语义**: 需要发出何种控制信息、完成何种动作、作出何种反应
  - **同步**: 包括速率匹配和排序

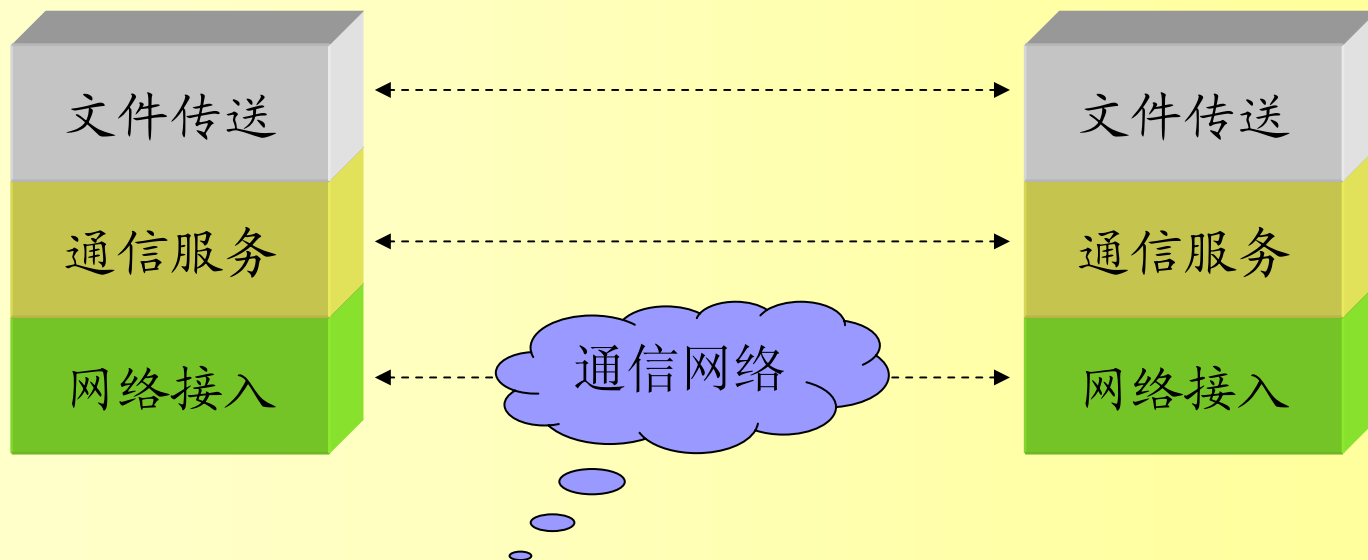




## 2.2 协议体系结构

### ■ 简单协议体系结构

- 文件传送
- 通信服务
- 网路接入





## 2.2 协议体系结构

### ■ 网络协议的层次结构的好处

- ☐ 各层独立
- ☐ 灵活性好
- ☐ 结构上可以分割
- ☐ 易于实现与维护
- ☐ 促进标准化工作

### ■ 计算机网络的各层及其协议的集合成为网络的体系结构



## 2.2 协议体系结构

- 第一个网络体系结构

- IBM 公司1974年提出的System Network Architecture (SNA)。

- 其他的体系结构

- Digital公司: Digital Network Architecture (DNA)
  - 宝来机器公司: Burroughs Network Architecture (BNA)
  - Honeywell公司: Distributed System Architecture (DSA)



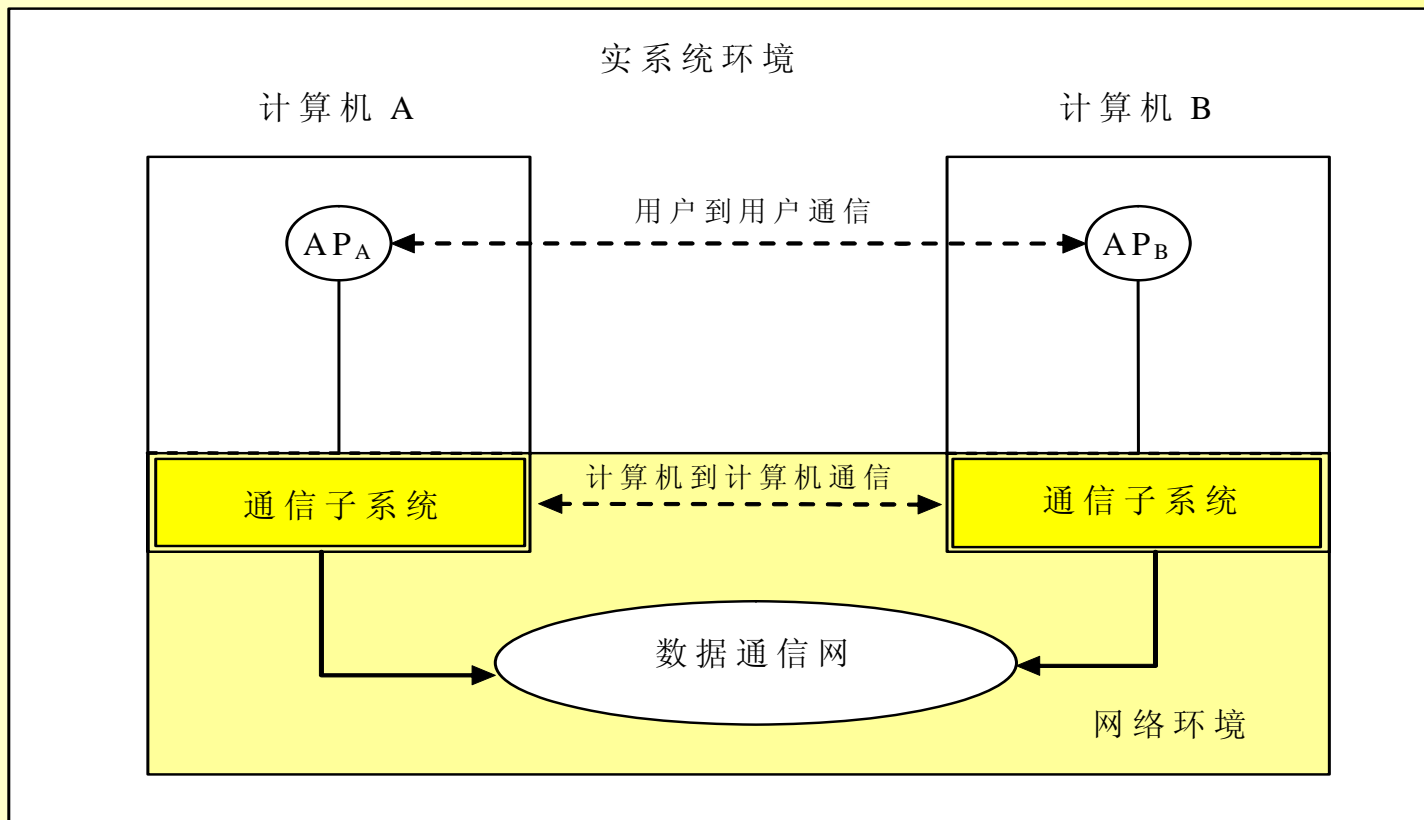
## 2.3 OSI : Open System Interconnection

- 1977年 ISO 建立了一个组织机构负责开发计算机之间的通信。成果：**开放系统互联**。
- **开放**：一个系统只要遵循OSI 标准，就可以和世界上任何地方、也遵循这个同意标准的任何系统进行通信。
- **系统**：按一定关系或规则工作在一起的一组物体或部件。它可以是一台或多台计算机以及和这些计算机相关的软件、外部设备、终端等。
- 开放互联系统是使世界范围内的应用进程能开放式地进行信息交换



## 2.3 开放系统互联 OSI

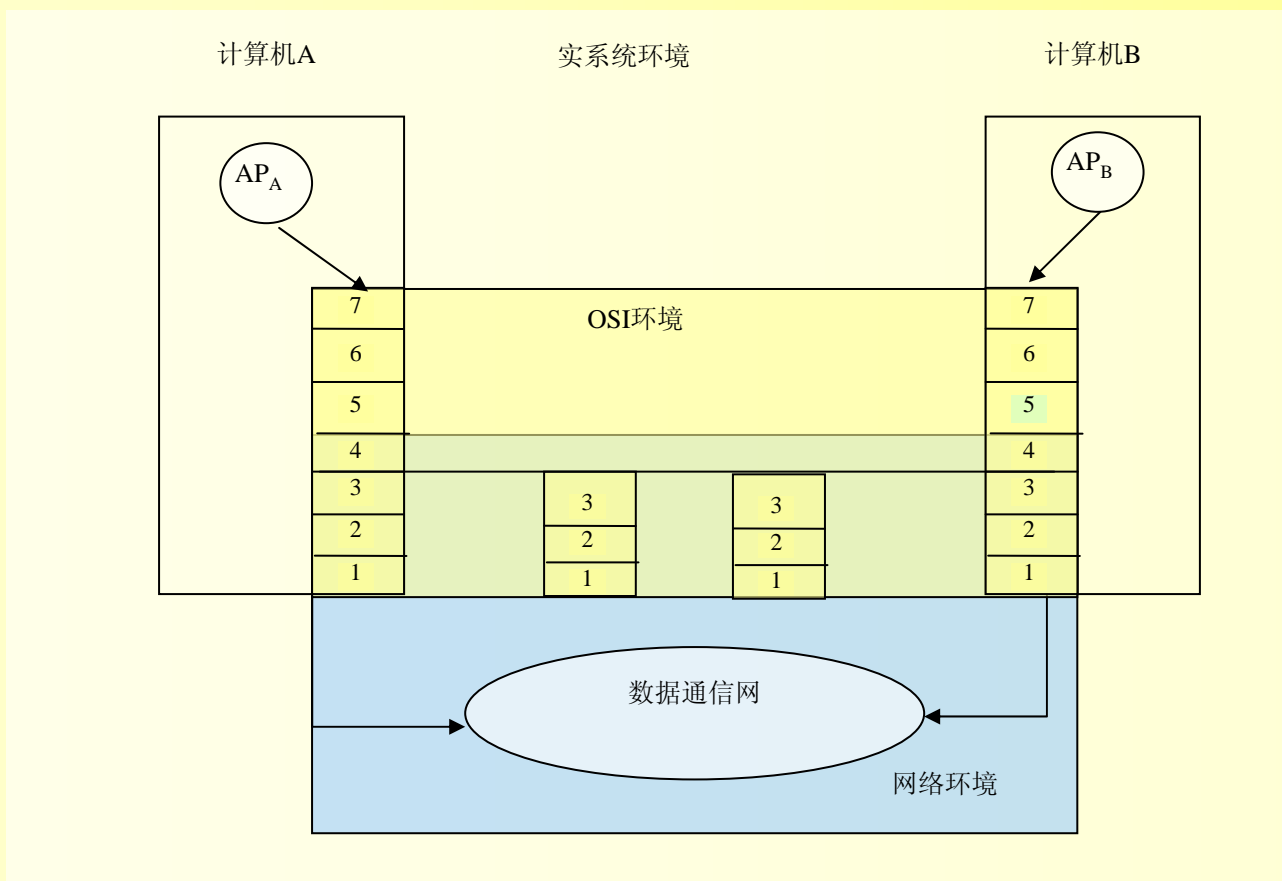
### ■ 网络环境





## 2.3 开放系统互联 OSI

### ■ 开放互联环境





## 2.3 开放系统互联 OSI

- **开放系统互连环境**。在一个计算机中只有开放系统互连参考模型中的7个层次才处在开放系统互连环境之内。包括应用进程在内的其他部分，都不属于开放系统互连环境，它们都处在本地系统之中。
- **中继开放系统**。数据通信网中的交换结点只包含有下三层。表示结点的这三个层次又称为中继开放系统。
- **信息传递路线**。
- **网络环境**。包括7层中的第1至4层、中继开放系统和整个网络的物理媒体。
- **用户视图**。每个用户能够看见它们自己的本地系统管理模块、对方用户的映象。看不见数据格式的匹配、流量控制、差错控制、传输速率的匹配、路由选择以及传输媒体的选择等等。



## 2.3 开放系统互联 OSI

- (N)层：第N层。
- 实体(entity)：表示任何可以发送或接收信息的硬件或软件进程。
  - 每一层都可以看成是由若干个实体所组成。
  - 位于不同子系统的同一层内相互交互的实体，就构成了对等实体。
- 对等层(peer layer)之间的通信
  - 复杂过程每一层实体来说都被屏蔽掉了。
  - 各层协议，实际上就是在各个对等层之间传递数据时的各项规定。

	开放系统A		开放系统B		开放系统C	
最高层						
(N+1)层						(N+1)子系统
(N)层	(N)实体		(N)实体		(N)实体	(N)子系统
(N-1)层						(N-1)子系统
最低层						





# 协议与服务

## ■ 协议

- 控制两个对等(N)实体进行通信的规则集合称为(N)协议。
- 协议的语法方面的规则定义了所交换的信息的格式
- 协议的语义方面的规则定义了发送者或接收者所要完成的操作

## ■ 服务

- 两个(N)实体间的通信(在(N)协议的控制下),使(N)层能够向上一层提供服务,这种服务称为(N)服务。
- 接受(N)服务的是上一层的实体,即(N+1)实体。它们也称为(N)用户或更严格些是(N)服务用户。
- (N)实体要实现(N)协议还要使用(N-1)实体提供的(N-1)服务。

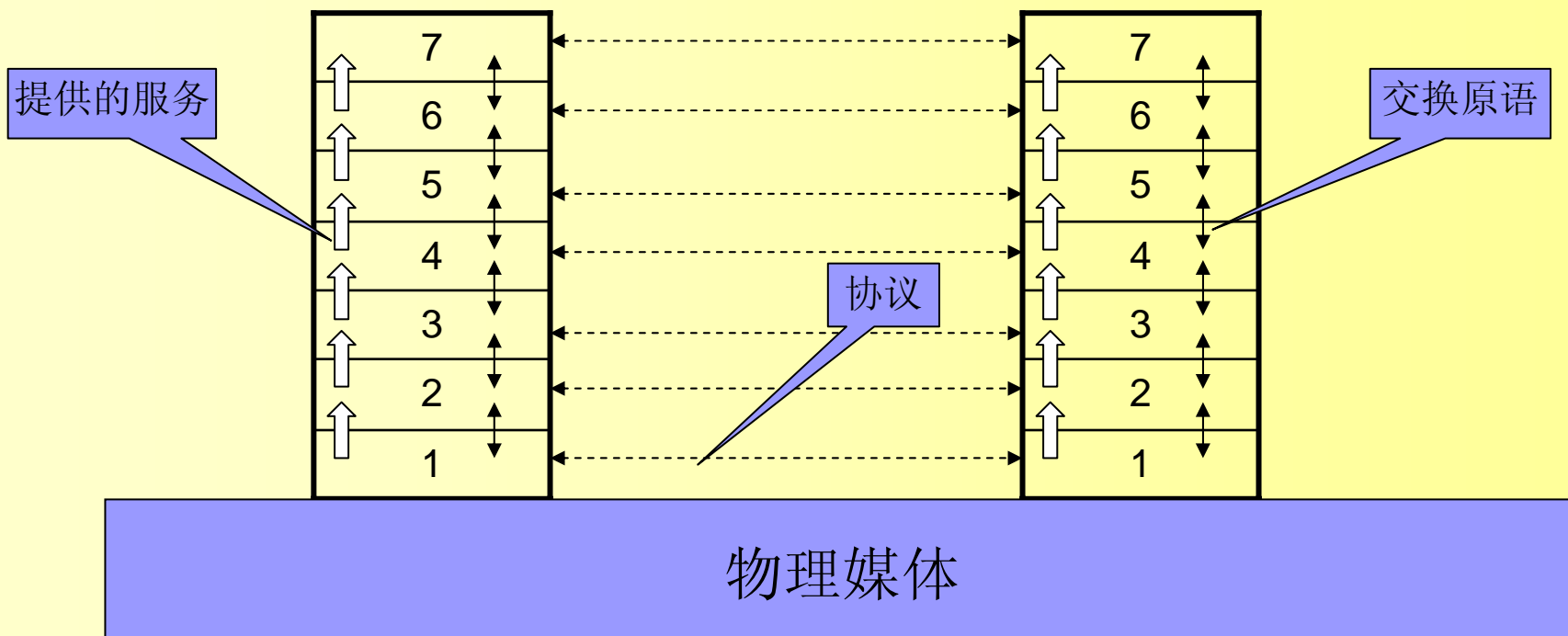


# 协议与服务

- (N)协议的实现保证了(N)服务得以向上一层提供，但(N)服务用户只能看见(N)服务而无法看见(N)协议。(N)协议对(N)服务用户是透明的。
- 协议是“水平的”，即协议是控制对等实体之间的通信的规则。但服务是“垂直的”，即服务是由下层向上层通过层间接口提供的。
- 上层通过与下层的服务原语的交换来使用下层所提供的服务。
- 并非在(N)层内完成的全部功能都称为(N)服务，只有那些能够被高层看得见的功能才能称之为“服务”。



# 协议与服务





# 协议与服务

## ■ 服务访问点SAP(Service Access Point)

- 服务访问点SAP (N)实体和上一层的(N+1)实体的逻辑接口，有时也可称之为端口(port)或插口(socket)。
- 显然，这里所说的“接口”和通常所说的两个设备之间的并行接口或串行接口是很不一样的。

## ■ 一个(N)实体向上一层所提供的服务由以下三部分组成

- (N)实体自己提供的某些功能;
- 从(N-1)层及其以下各层以及本地系统环境得到的服务;
- 通过与处在另一开放系统中的对等(N)实体的通信而得到的服务。

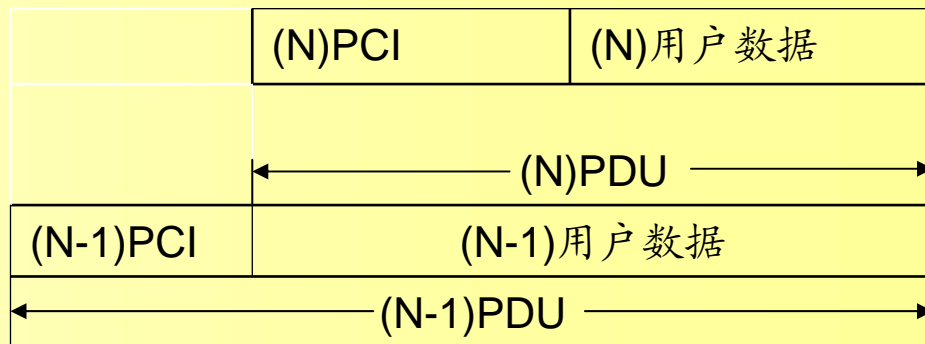


# 信息传送单元

## ■ 协议数据单元PDU

- 协议数据单元就是在不同站点的各层对等实体之间，为实现该层协议所交换的信息单元。
- 考虑到协议的要求，如时延、效率等因素，协议数据单元的大小一般都有一定的要求。
- 通常将第N层的协议数据单元记为(N)PDU。

- 本层的用户数据，记为(N)用户数据；
- 本层的协议控制信息，记为(N)PCI。(Protocol Control Information)。





# 信息传送单元

## ■ 接口数据单元IDU

- OSI规定：在同一系统的相邻两层实体的一次交互中，经过层间接口的信息单元的大小，称为接口信息单元IDU。
- 接口信息单元的大小和相应的协议数据单元的大小并无直接的关系。例如，协议数据单元可以是1000个字节，但接口可能要求每次只能通过1个字节。
- 协议数据单元在通过层间接口时，还需要加上一些控制信息，例如说明一共通过多少个字节等。这些信息称为接口控制信息ICI(Interface Control Information)
- $IDU = ICI + PDU$
- 当IDU通过层间接口后，就将加上的ICI去掉了。



# 信息传送单元

## ■ 服务数据单元SDU

- (N)实体完成 (N) 服务用户的请求所需要的数据单元, (N) SDU。
- $(N) \text{ PDU} = (N) \text{ PCI} + (N) \text{ SDU}$ ,
- $(N-1) \text{ IDU} = (N) \text{ PDU} + (N-1) \text{ ICI}$
- $(N-1) \text{ PDU} = (N-1) \text{ PCI} + (N-1) \text{ SDU}$

- 当  $(N) \text{ SDU} > (N) \text{ 用户数据时}$ , 就需要分片。



# 服务原语

- 当(N+1)实体向(N)实体请求服务时，服务用户与服务提供者之间进行交互时所要交换的一些必要信息就称为服务原语
  - 表明需要本地的或远端的对等实体做哪些事情。
- OSI规定了每一层均可使用的四种服务原语类型。

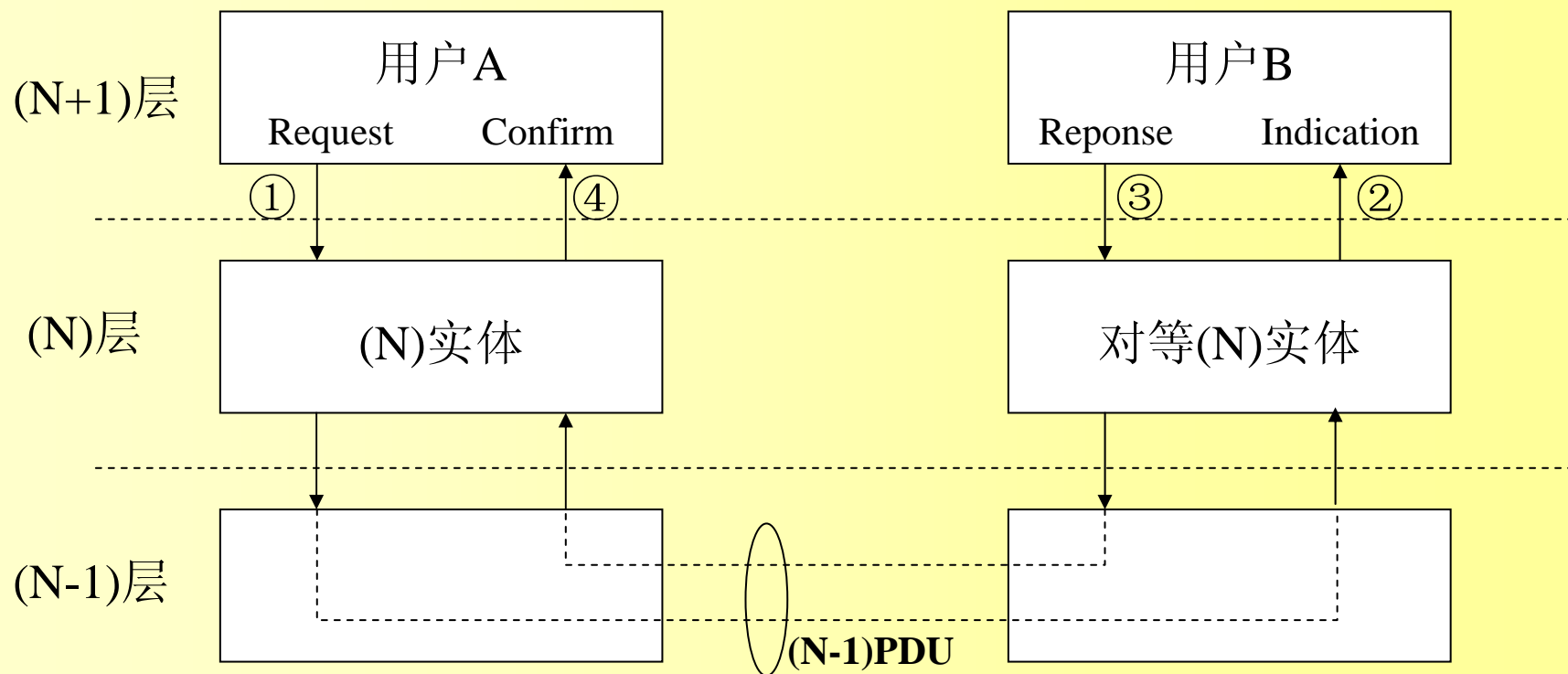
服务原语类型	译名	意 义
Request	请求	一个实体希望得到某种服务
Indicator	指示	把关于某一事件的信息告诉某一实体
Response	响应	一个实体愿意响应某一事件
Confirm	证实	把一个实体的服务请求加以确认并告诉它





# 服务原语

## ■ 空间表示法

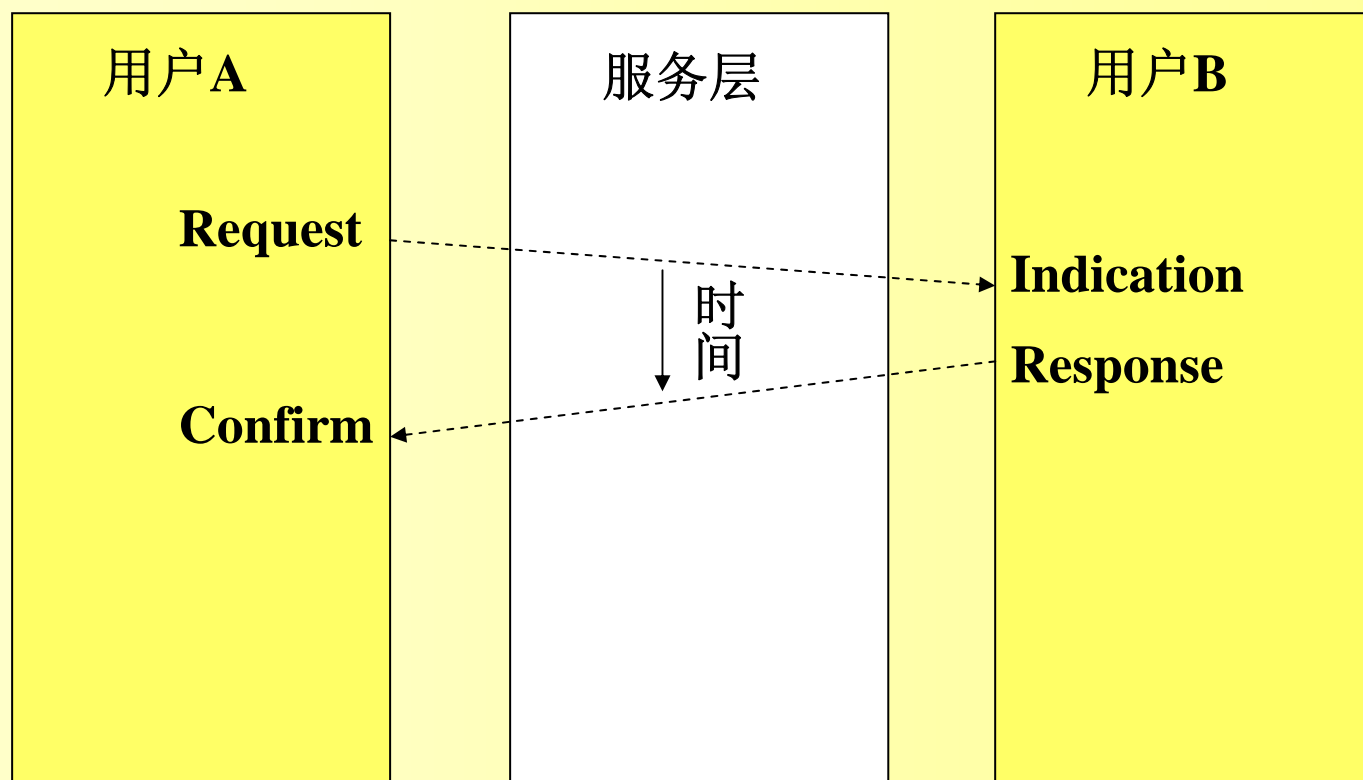




# 服务原语

## ■ 时间表示法

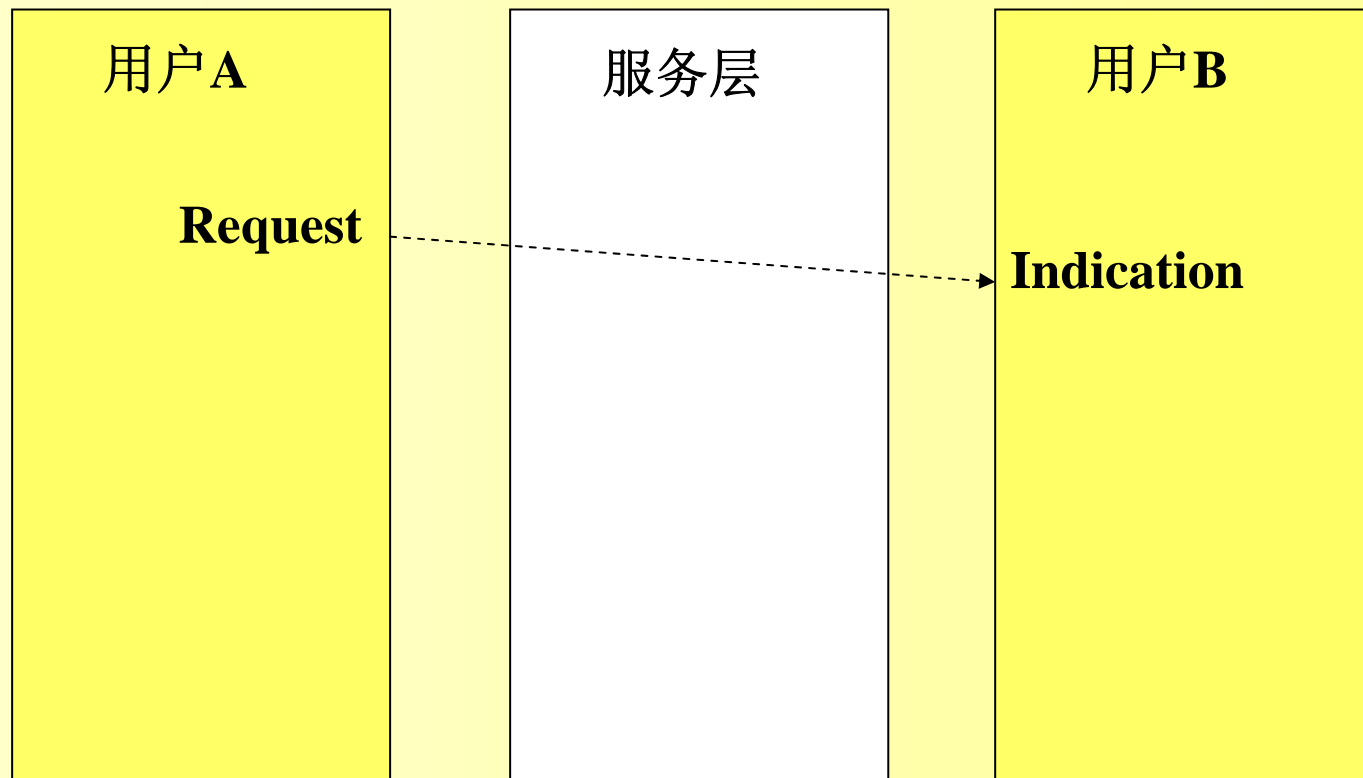
### □ 需要证实的服务





# 服务原语

- 时间表示法
  - 不需要证实的服务





# 服务原语

### ■ 设张三想给李四打电话

- 张三就要向电信局发出要求电话服务的请求，张三先摘机，然后拨李四的电话号码。这就相当于发出**Request**服务原语，被叫李四的电话号码就相当于原语所携带的参数。
- 李四的电话振铃了，相当于服务提供者(电信局的电话交换机)向李四发出。**Indication**服务原语。
- 李四摘机，相当于服务用户发出**Response**服务原语。
- 张三在耳机中听见回铃声断了，知道李四摘机了，相当于服务提供者发出**Confirm**服务原语，使服务用户知道刚才所请求的服务已经实现了。



# 面向连接服务

- 面向连接服务是在数据交换之前，必须先建立连接。当数据交换结束后，则应终止这个连接。
- 在建立连接阶段，在有关的服务原语以及协议数据单元中，必须给出源用户和目的用户的全地址。
- 在数据传送阶段，就可以使用一个连接标识符来表示上述这种连接关系。
- 连接标识符通常比一个全地址的长度要短得多。在连接建立阶段，还可以协商服务质量及其他任选项目。
- 当被叫用户拒绝连接时，连接即告失败。



# 面向连接服务

- 通常的面向连接服务是可靠的报文序列服务。
  - 每个用户都可以发送可变长度(在某一最大长度限度内)的报文，这些报文按顺序发送给远端的用户。
  - 报文的接收也是按顺序的。
  - 有时用户可以发送一个很短(1至2字节长)的报文，但希望这个报文可以不按序号而优先发送。这就是“加速数据”(expedited data)，它常用来传送中断控制命令。
- 面向连接服务在网络层中又称为虚电路服务。
  - “虚”表示：虽然在两个服务用户的通信过程中没有自始至终都占用一条端到端的完整物理电路，但却好象占用了一条这样的电路。
  - 若两个用户需要经常进行频繁的通信，则可建立永久虚电路。



# 无连接服务

- 在无连接服务的情况下，两个实体之间的通信不需要先建立好一个连接，因此其下层的有关资源不需要事先进行预定保留。
- 无连接服务的优点是灵活方便和比较迅速。但无连接服务不能防止报文的丢失、重复或失序。
- 当采用无连接服务时由于每个报文都必须提供完整的目的站地址，因此其开销也较大。
- 无连接服务比较适合于传送少量零星的报文。



# 无连接服务的三种类型

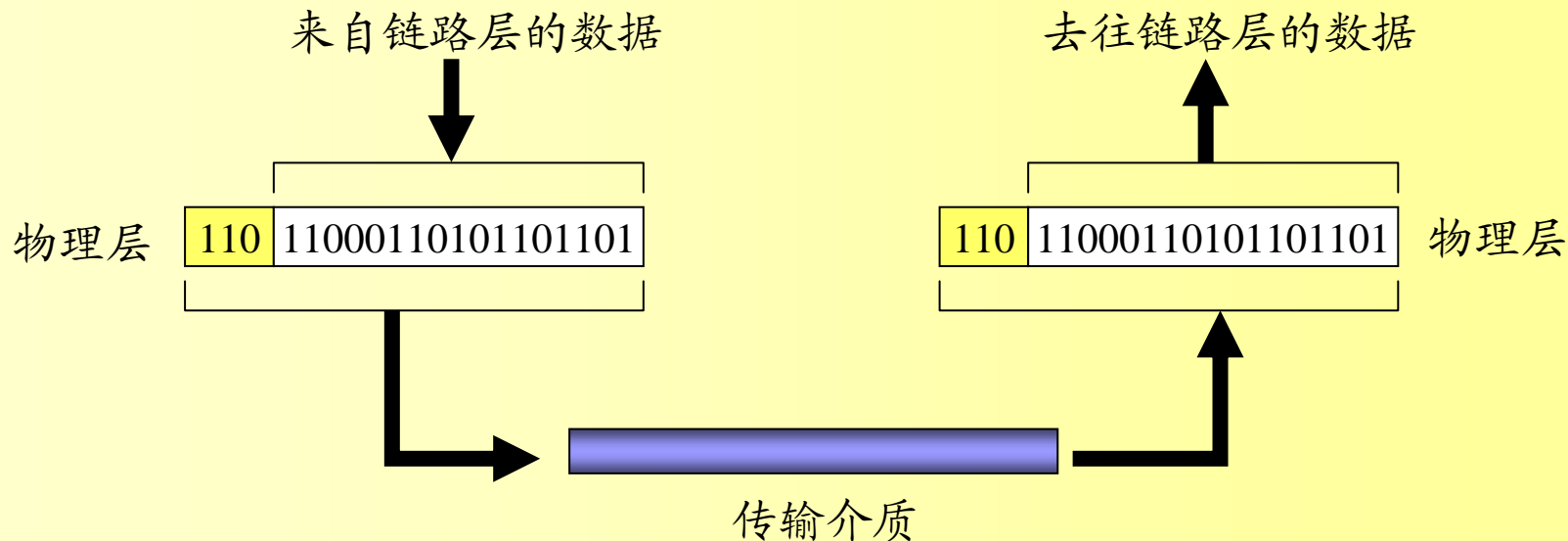
- 数据报(datagram)。
  - 它的特点是发完了就算，而不需要接收端做任何响应。
- 证实交付(confirmed delivery)
  - 它又称为可靠的数据报。这种服务对每一个报文产生一个证实给发方用户，
  - 不过这个证实不是来自接收端的用户而是来自提供服务的层。这种证实只能保证报文已经发给远端的目的站了，但并不能保证目的站的用户已经收到了这个报文。挂号的电子邮件就属于这种类型。
- 请求回答(request reply)
  - 这种类型的数据报是**收端用户**每收到一个报文，就向发端用户发送一个应答报文。
  - 双方发送的报文都有可能丢失。





# 物理层

- 物理层(physical layer)包含那些在物理介质上传输位流所必需的功能。它定义了接口与传输介质的机械和电气特性，也定义了物理设备和接口为了传输而必须执行的过程和功能。





# 物理层

### ■ 接口与介质的物理特性

- 物理层定义了设备与传输介质之间的接口特性，也定了传输介质的类型。

### ■ 位的表示。

- 物理层的数据是没有任何解释的位(bit)流(由0和1所组成的序列)。
- 物理层定义编码(encoding)的类型，如何将0和1转换成信号 — 电信号或光信号。

### ■ 数据速率

- 传输速率 (transmission rate)，即每秒发送的位数，也在物理层定义。换言之，物理层也定义一个位持续多长时间。

### ■ 位同步

- 发送方与接收方不仅使用相同的比特率，还必须位同步。换言之，发送方的时钟与接收方的时钟必须同步。



# 物理层

### ■ 线路配置

- 物理层涉及设备与介质的连接。在**点到点配置**中，两个设备通过一条专用链路连接。在**多点配置**中，许多设备共享一条链路。

### ■ 物理拓扑结构

- 物理拓扑结构定义如何将设备连接成网络。设备的连接方式可以是**网状**拓扑结构(每台设备均与其他设备连接)、**星型**拓扑结构(通过中心设备与其他设备连接)、**环状**拓扑结构(每台设备与下一台设备连接以组成一个环)和**总线**拓扑结构(每台设备都在一条公共链路上)。

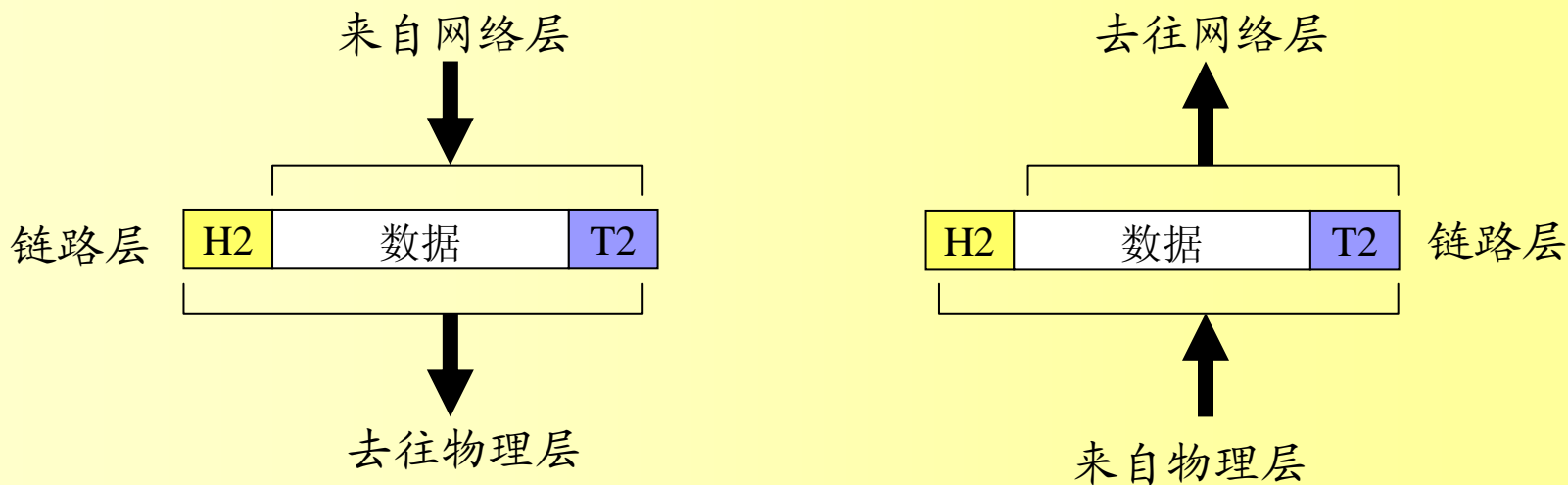
### ■ 传输方式

- 物理层也定义两台设备之间的传输方向：**单工**、**半双工**和**全双工**。在单工方式中，只有一个设备能发送，另一个设备只能接收。单工方式是单向通信。在半双工方式中，两台设备都能发送和接收，但不能在同一时刻。在全双工(或双工)方式中，两个设备能在同一时刻发送与接收。



# 数据链路层

- 数据链路层(data link layer)将物理层——对数据不做任何改动的传输通道变成可靠的链路，这样可以将物理层的数据无差错地传递给上层。



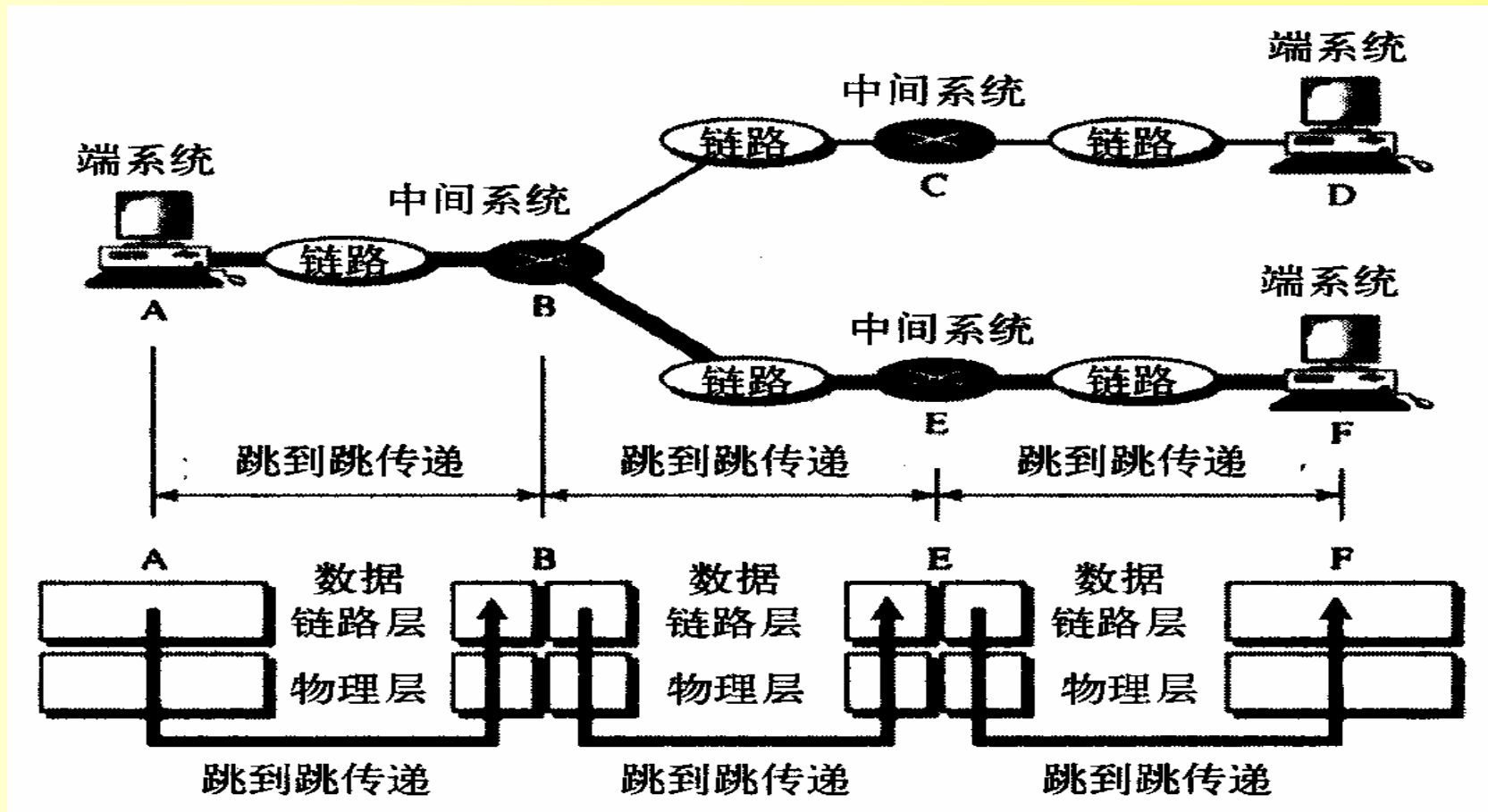


# 数据链路层

- 成帧。
  - 数据链路层将接收到的来自网络层的位流划分成称为帧(frame)的易处理数据单元。
- 物理寻址。
  - 如果帧是发送给网络中不同系统, 则数据链路层在帧的头部添加发送方的物理地址与接收方的物理地址。如果帧要发往发送方网络之外的系统, 那么接收方的地址就是连接一个网络到下一个网络的设备地址。
- 流量控制
  - 如果接收方接收数据的速率小于发送方发送的速率, 那么数据链路层就会采用流量控制机制以防止接收方过载。
- 差错控制
  - 数据链路层增加一些机制用来检测与重发损坏帧或丢失帧, 从而增加了物理层的可靠性。还用了一个机制防止重复帧。差错控制通常在一个帧的结束处增加一个尾部来处理。
- 访问控制
  - 当两台或多台的设备连接到同一条链路上时, 数据链路协议必须能决定在任意时刻由哪一台设备来获取对链路的控制权。



## 2.3 开放系统互连 OSI



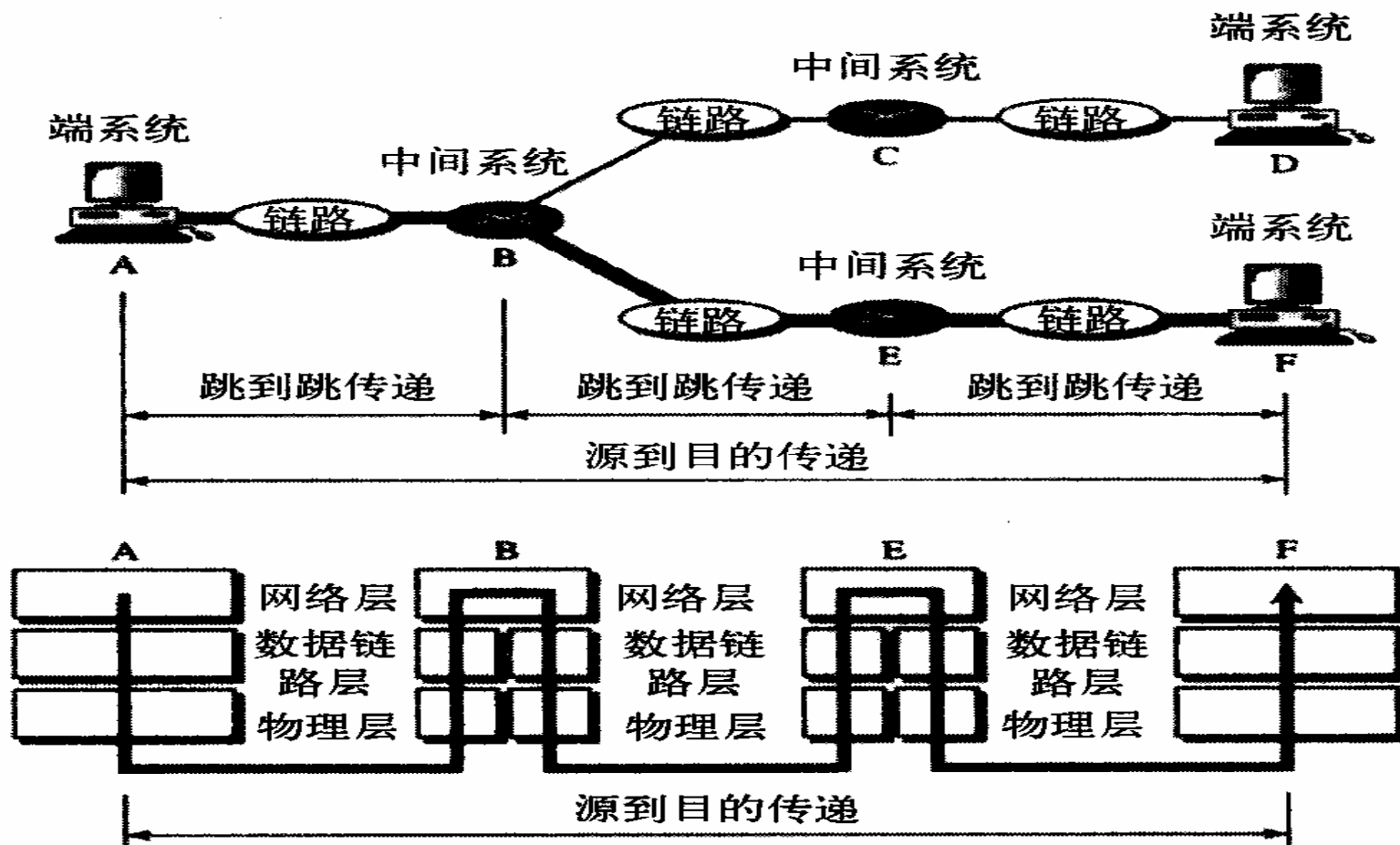


# 网络层

- 网络层(network layer)负责将分组从源地址传递到目的地地址，可能会通过多个网络(链路)。
  - 尽管数据链路层会监视同一个网络(链路)上两个系统之间分组的传递，但是还需要网络层确保每个分组能够从出发点到达目的地。
  - 如果两个系统连接在同一条链路上，则通常不需要网络层。然而，如果两个系统在不同的网络(链路)上并通过网络(链路)之间的设备连接，通常就需要网络层以完成源端到目的端的传递。



# 网络层







# 网络层

## ■ 逻辑寻址

- 物理寻址负责处理本地网络寻址问题，它由数据链路层完成。如果一个分组通过网络边界，就需要另一个寻址系统来帮助区分源与目的系统。网络层给来自上一层的分组增加一个头部，它包括发送方与接收方的逻辑地址。

## ■ 路由选择

- 当一些独立网络或链路连接在一起构成一个互联网(一个由网络连成的网络) 或一个更大的网络时，连接设备(称为路由器或网关)把分组送到它们的最终目的地， 网络层的功能之一是提供这种机制。

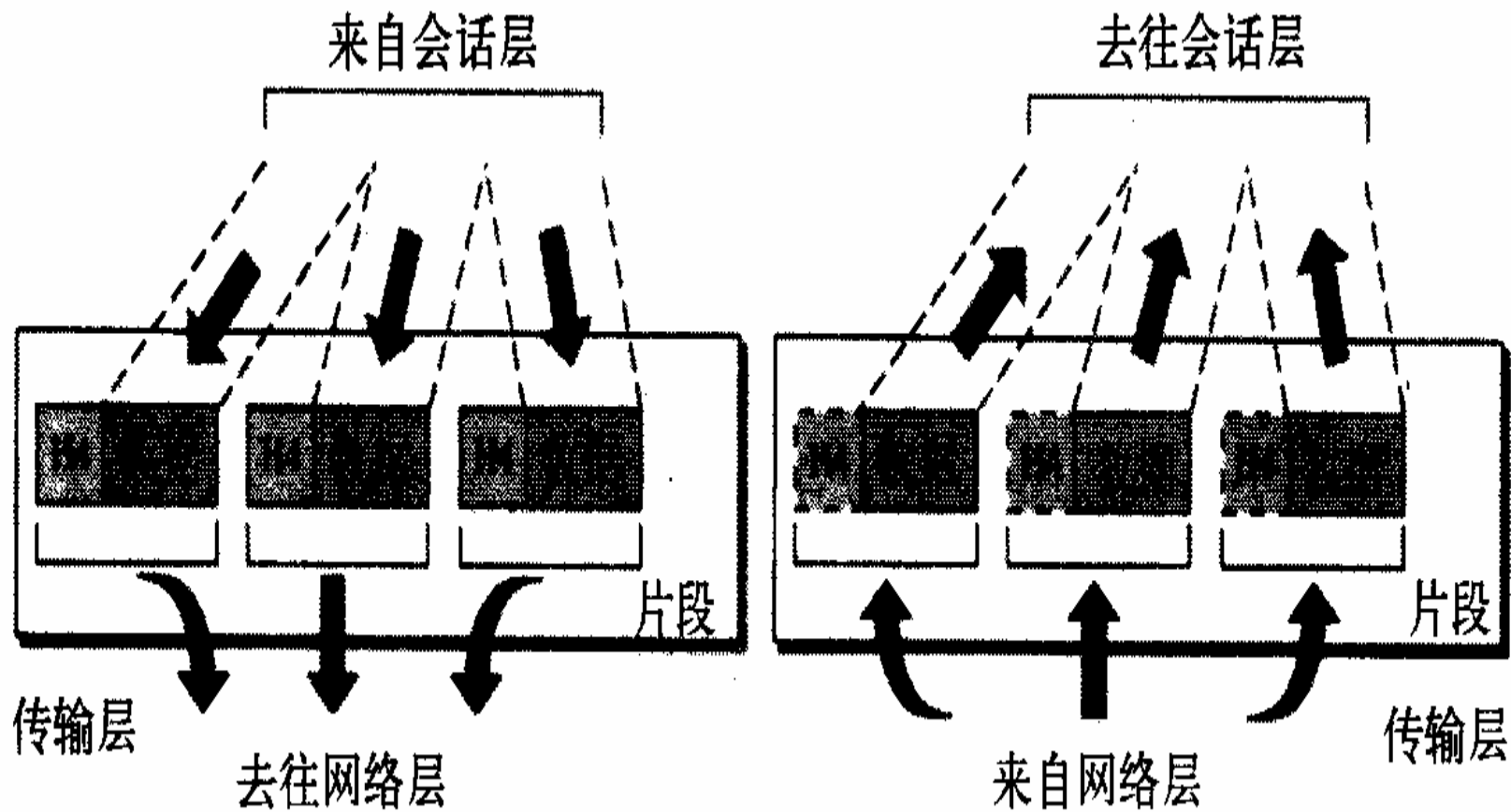


# 传输层(transport layer)

- 传输层负责整个报文的进程到进程传递(process to process delivery)。进程是主机上运行的应用程序。
- 尽管网络层监管各个分组的源端到目的端传递(source to destination delivery)，但是它并不了解这些分组之间的相互关系。网络层独立处理每个分组，就像每个分组都属于一个独立报文一样，不管实际上是否这样。
- 传输层正好相反，它确保整个报文无差错并按顺序地到达目的地，并在源到目的层次进行差错控制和流量控制。



# 传输层(transport layer)





# 传输层(transport layer)

## ■ 服务点寻址

- 网络层将每个分组传送到指定的计算机上，而传输层则将整个报文传送给该计算机上的指定进程。

## ■ 分段和组装

- 将报文分解成可传输的片段，并且给这些片段编上序号。这些序号不仅使传输层可以在接收端将报文正确地组装，而且可以用来标识和替换传输中丢失的分组。

## ■ 连接控制

- 传输层可以是无连接的或面向连接的。无连接的传输层将每个片段作为一个独立的分组，并将它传递到目的机器的传输层。而面向连接的传输层，在传递分组之前，需要与目的机器的传输层建立一条连接。所有数据被传输后，该连接释放。



### ■ 流量控制

- 如同数据链路层一样，传输层担负流量控制(Flow control)的责任。然而，这一层的流量控制是在端到端上，而不是在单条链路上。

### ■ 差错控制

- 如同数据链路层一样，传输层担负差错控制(error control)的责任。然而，这一层的差错控制是在进程到进程上，而不是在单条链路上。发送方的传输层确保整个报文无差错(损坏、丢失或重复)地到达接收方的传输层。通常通过重发来纠正差错。



# 会话层

## ■ 会话层负责对话控制和同步。

### □ 对话控制

- 会话层允许两个系统进入会话。它允许两个进程之间以半双工(一次一个方向)或全双工(一次两个方向)方式进行通信。

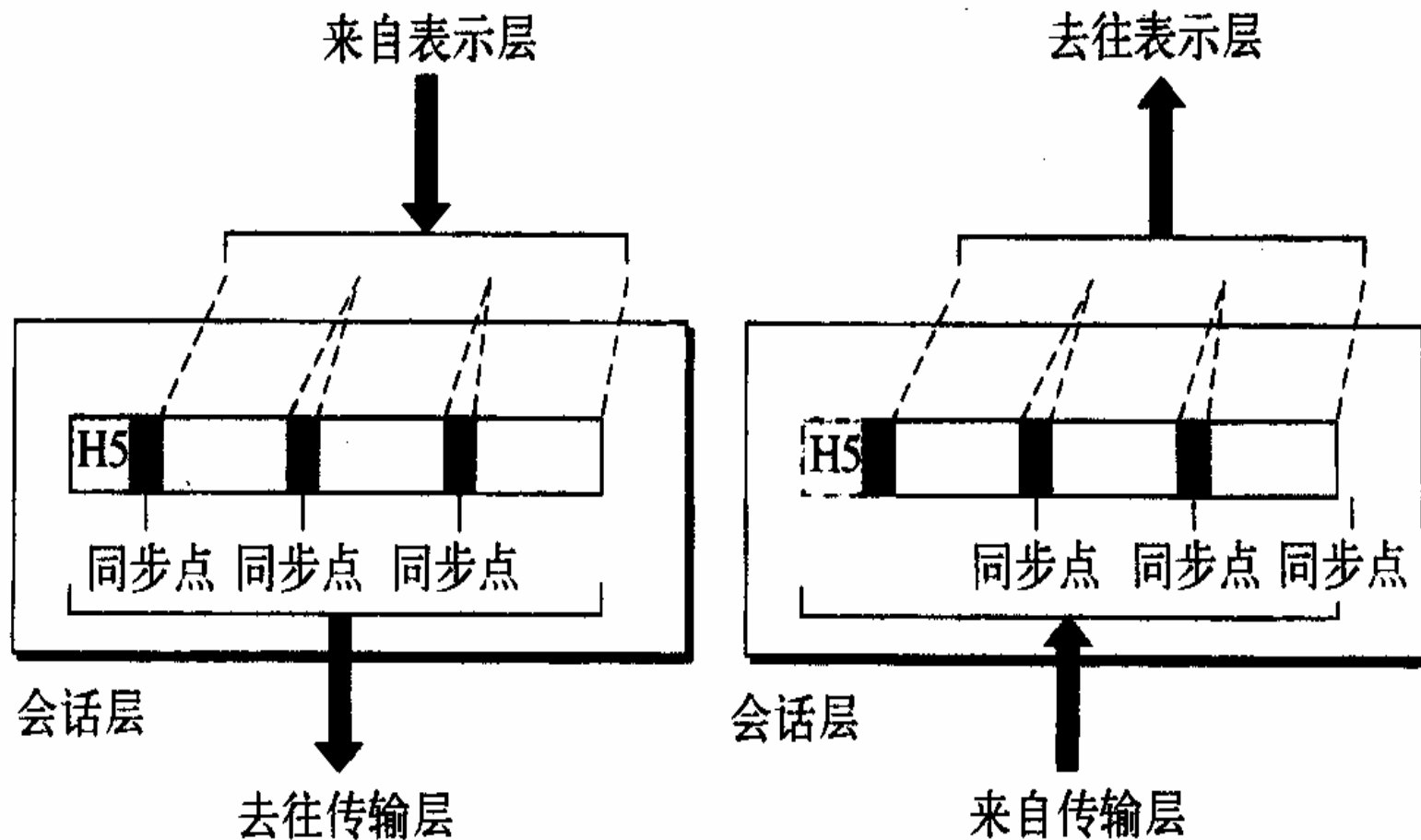
### □ 同步

- 会话层允许一个进程在数据流中增加检查点或同步点(synchronization point)。

- 例如，假定一个系统要传输2000页的一个文件，为了确保每收到100页就独立地进行确认，每100页后应适当插入一个检查点。在这种情况下，如果传输到第532页发生崩溃，则在第501页处开始重传，而第1页到第500页不必重传。



## 会话层





# 表示层(presentation layer)

表示层负责翻译、加密和压缩数据。

### ■ 翻译

- 两个系统中的进程(运行程序)通常以字符串、数字等格式来交换信息。在传输前，信息必须变换成位流。因为不同计算机采用不同的编码系统，因此表示层负责不同编码方法之间的互操作。发送方的表示层将信息从发送方使用的格式转换成一个公共的格式。接收方机器的表示层将公共格式转换成接收方使用的格式。

### ■ 加密

- 为了传输敏感信息，一个系统必须保证秘密。加密是指发送方将原来的信息转换成另一种形式，并在网络上发送转换后的形式。解密是将信息转换回它的原始形式。

### ■ 压缩

- 数据压缩减少需要传输数据的位数。数据压缩对于多媒体(如文本、音频和视频)的传输显得特别重要。





# 应用层(application layer)

- 使得用户(不管是人还是软件)可以访问网络。它提供了用户接口和服务支持。
- 比如电子邮件、Web服务, 视频点播, 远程文件访问与传输, 共享数据库管理以及其他分布式信息服务。



## 2.4 TCP/IP参考模型

- 与OSI参考模型不同，TCP/IP模型更侧重于互联设备间的数据传送，而不是严格的功能层次划分。
- OSI参考模型在解释互联网络通信机制上比较适合，但TCP/IP成为了互联网络协议的市场标准。



# Internet简史 — ARPA的要求

- 在60年代初，美国国防部领导的远景研究规划局ARPA ( Advanced Research Project Agency ) 提出要研制一种崭新的、能够适应现代战争的、生存性很强的网络。
  - 和传统的电信网不同，这种新型的网络不是为了打电话，而是用于计算机之间的数据传送。
  - 新型的网络能够连接不同类型的计算机，即不局限于单一类型的计算机。
  - 所有的网络结点都同等重要。
  - 计算机在进行通信时，必须有冗余的路由。
  - 网络的结构应当尽可能地简单，但能够非常可靠地传送数据。

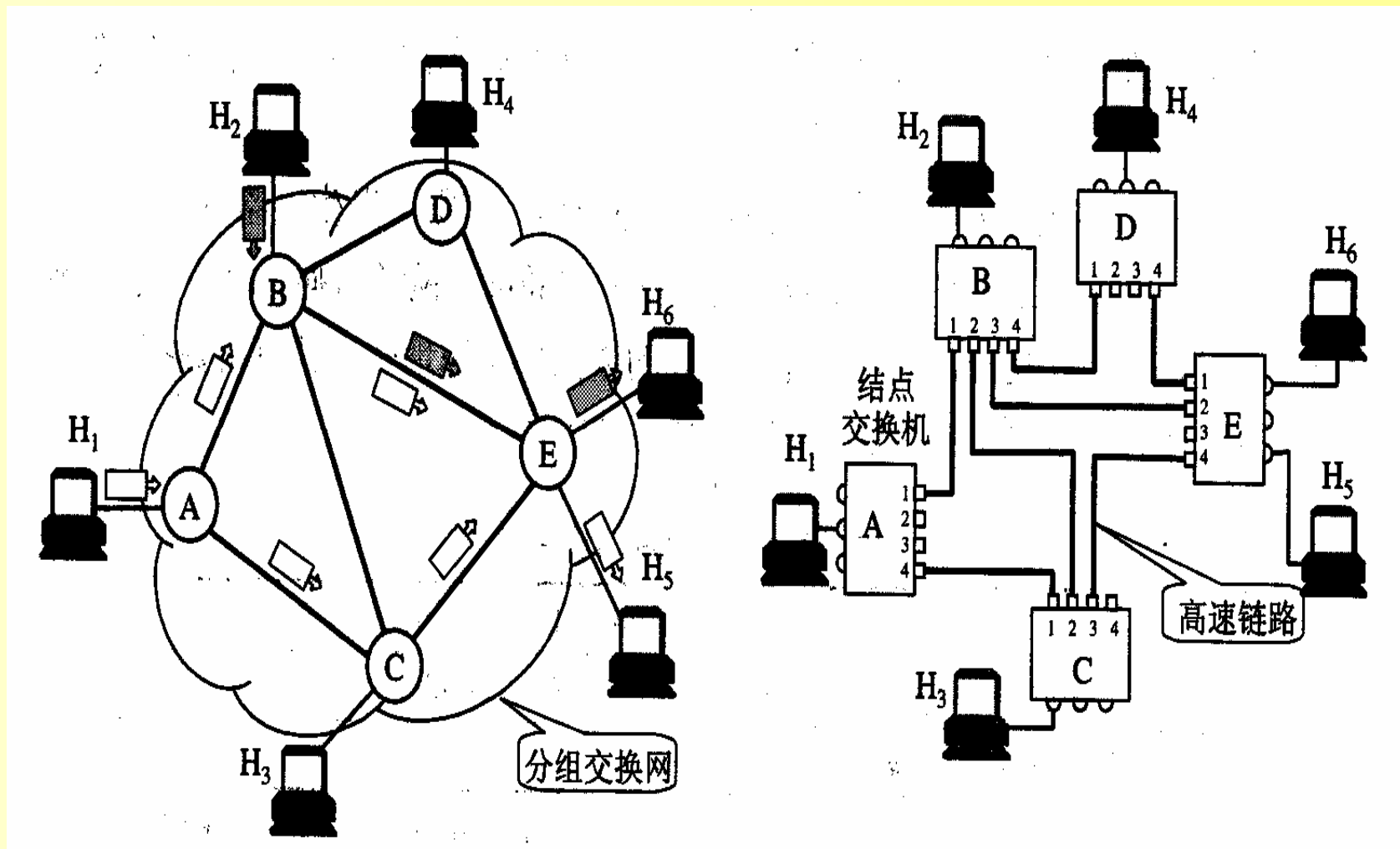


# Internet简史

- Leonard Kleinrock 和一个麻省理工学院的博士，抽象化了包交换的底层技术，并在1961年出版了关于该项目的论文。使用排队论证明分组交换网络在数据通信方面的优越性。
- Kleinrock的工作向他们提供了关键思想
- 来自麻省被称为BBN的小型声像公司签定了合同。1969产生了ARPANET。



# Internet简史—— 分组交换网





# Internet简史

- 最初的A R P A N E T包括四台主机，它们分别位于U C L A、斯坦福研究所、加州大学圣巴巴拉分校和犹他大学，使用网络控制协议(Network Control Protocol, N C P)，向用户提供的服务能力包括登录到远程主机、在一个远程打印机上打印、传输文件等。
- R a y Tomlinson, 一个B B N的工程师，在1 9 7 1年编写了第一个邮件程序。



# Internet简史

- 1974年，Vinton Cerf和Robert Kahn发明了传输控制协议(Transmission Control Protocol, TCP)。在80年代初代替了受限的NCP。
- NSF认识到ARPANET所进行工作的重要性并决定发展自己的网络。NSFNet连接了大学和政府职能部门的许多超级计算机。
- 在1980年后和1990年初，NSFNet代替了旧有速度慢的ARPANET，成为Internet正式骨干网。



# Internet简史 — 第一阶段

- 第一阶段是从单个网络**ARPANET** 向互连网发展的过程。
  - 1969 年美国国防部创建的第一个分组交换网ARPANET最初只是一个单个的分组交换网。
  - 所有要连接在ARPANET 上的主机都直接与就近的**结点交换机**相连。
  - 在ARPANET问世后，其规模一直增长很快。1984 年ARPANET上的主机已超过1000 台。
  - 到了70 年代中期，人们已认识到不可能仅使用一个单独的网络来满足所有的通信问题。于是ARPA 开始研究多种网络互联技术。
  - 1983年， ARPANET分解成两个网络。一个仍称为ARPANET，是进行实验研究用的科研网。另一个是军用的计算机网络MILNET。
  - 1990年ARPANET正式宣布关闭。

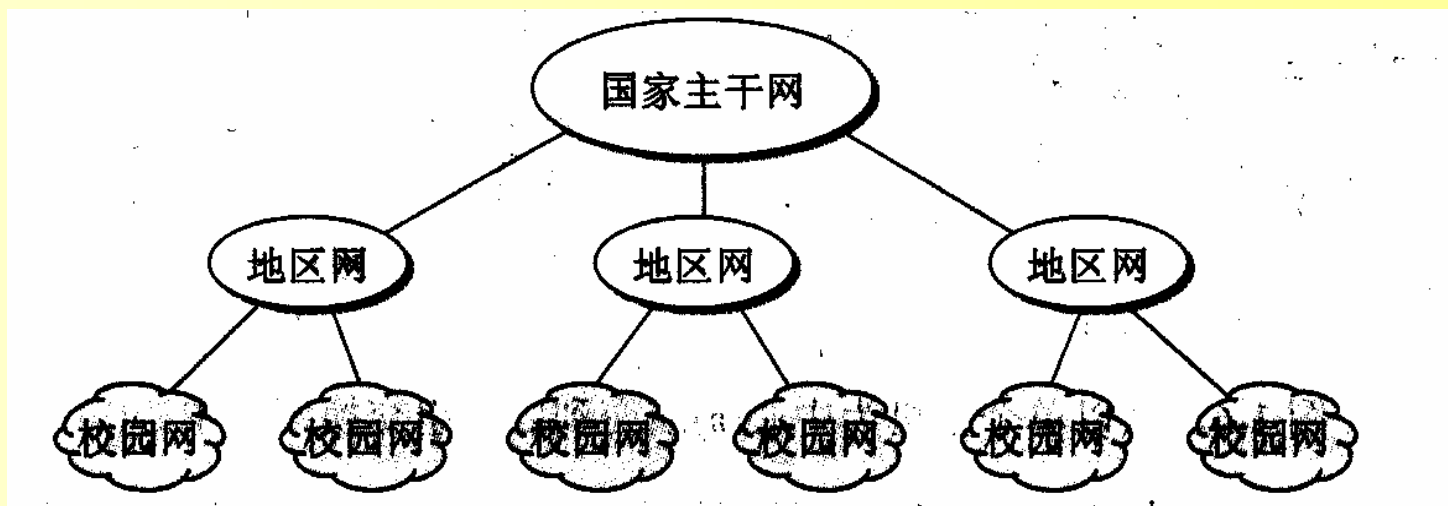




# Internet简史 — 第二阶段

## ■ 第二阶段的特点是建成了三级结构的因特网。

- ARPANET的发展使美国国家科学基金会NSF ( National Science Foundatlon ) 认识到计算机网络对科学研究的重要性, 因此从1985年起, 美国国家科学基金会就围绕六个大型计算机中心建设计算机网络。1986年, NSF建立了国家科学基金网NSFNET。它是一个三级计算机网络, 分为主干网、地区网和校园网。这种三级计算机网络覆盖了全美国主要的大学和研究所以。





# Internet简史 — 第三阶段

- 1991 年，NSF 和美国的其他政府机构开始认识到，因特网必将扩大其使用范围，不会仅限于大学和研究机构。
- 美国政府决定将因特网的主干网转交给私人公司来经营，并开始对接入因特网的单位收费。





# 计算机网络在我国的发展

- 最早着手建设专用计算机广域网的是铁道部，铁道部在1980年即开始进行计算机联网实验。
- 1989年11月我国第一个公用分组交换网CN PAC 建成运行。
- 1993年9月建成新的中国公用分组交换网，并改称为CHINAPAC。



## 全国范围的公用计算机网络

- 中国公用计算机互联网CH INAPAC
- 中国教育和科研计算机网CERNET
- 中国科学技术网CSTNET
- 中国联通互联网UNINET
- 中国网通公用互联网CNCNET
- 中国国际经济贸易互联网CIETNET
- .....



# 因特网的标准化工作

- 国际标准的制定又是一个非常复杂的问题，这里既有很多技术问题，也有很多属于非技术问题，如不同厂商之间经济利益的争夺问题等。
- 标准制定的时机
  - 标准制定得过早，由于技术还没有发展到成熟水平，会使技术比较陈旧的标准限制了产品的技术水平。其结果是以后不得不再次修订标准，造成浪费。
  - 反之，若标准制定得太迟。也会使技术的发展无章可循，造成产品的互不兼容，因而也会影响技术的发展。
- 因特网标准的制定
  - 因特网所有的技术文档都可从因特网上免费下载
  - 任何人都可以用电子邮件随时发表对某个文档的意见或建议。
  - RFC文档可以在 [www.ietf.org](http://www.ietf.org) 的网站上免费获得。



# 因特网的标准化工作

- 1992 年成立了一个国际性组织叫做因特网协会( Internet society , 简称为ISOC) , 以便对因特网进行全面管理以及在世界范围内促进其发展和使用。
- ISOC 下面有一个技术组织叫做因特网体系结构委员会 IAB ( Internet Architecture Board)
  - IETF : 短期和中期的工程问题, 主要是针对协议的开发和标准化。
  - IRTF: 进行理论方面研究和开发一些需要长期考虑的问题。

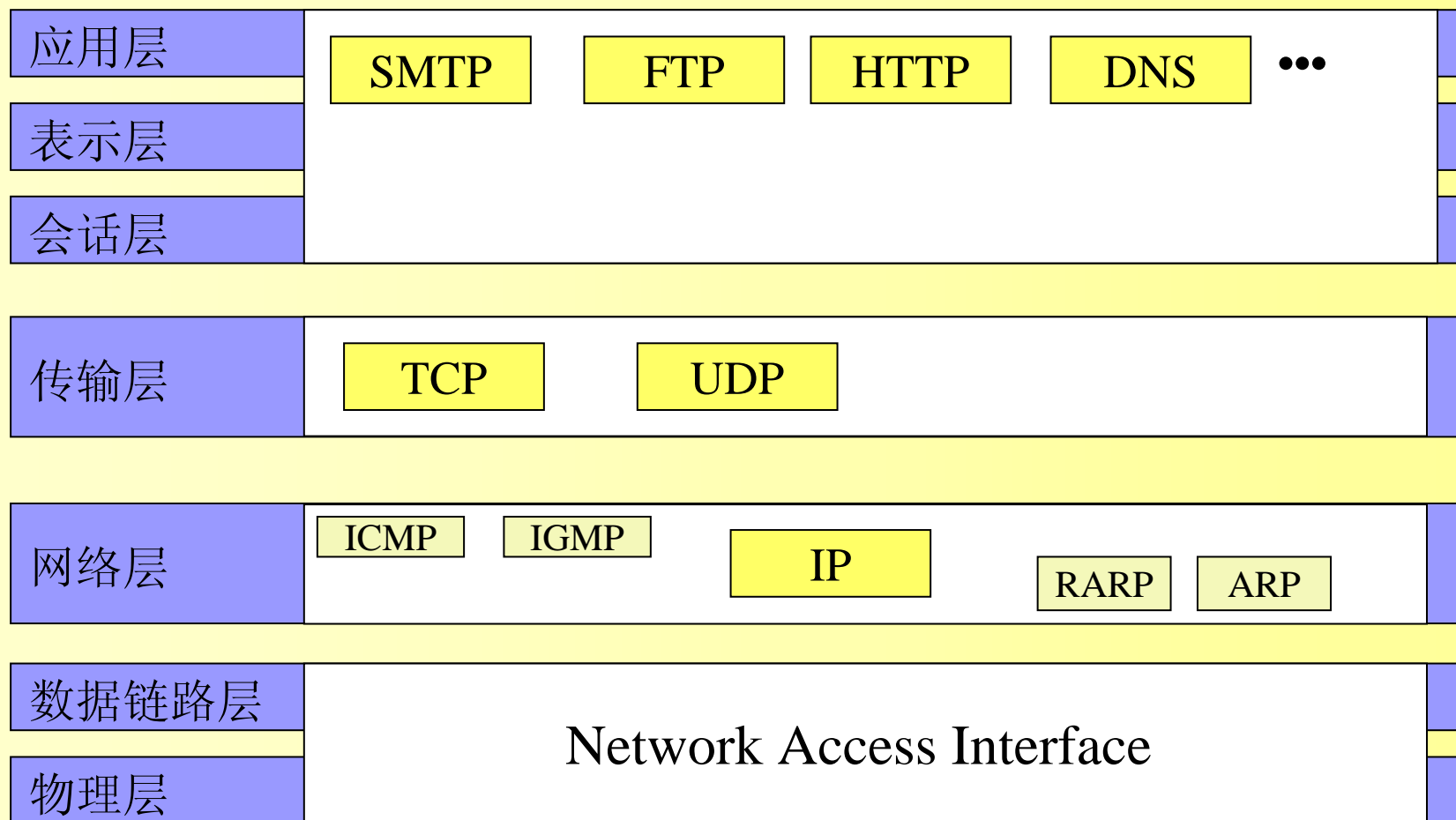


# TCP/IP协议体系结构

- TCP/IP分层模型（TCP/IP layering model）也被称为互联网分层模型（Internet Layering Model）或互联网参考模型（Internet Reference Model），包括了五层。



# TCP/IP 和 OSI模型







# 4层地址

- 应用层：专用地址（如：URL, 邮件地址等）
- 运输层：端口地址
- 网络层：逻辑地址（IP地址）
- 链路层：物理地址(MAC地址)



# 作业

- 习题 2.7 , 2.8。