

互联网体系结构

第三章 网络体系结构

1. 网络体系结构的基本概念
2. 开放系统互连基本参考模型
3. OSI 各层次中各层功能
4. TCP/IP 模型
5. OSI 参考模型与 TCP/IP 模型比较

一、网络体系结构基本概念

一个计算机网络系统是由：各个结点（具有通信功能的计算机系统）连接构成，实现相互通信和资源共享。

怎样构造计算机系统的通信功能，实现各系统之间的相互通信，尤其是异种计算机系统之间的相互通信？

——依靠**网络体系结构**解决

计算机网络体系结构与网络协议是计算机网络的两个最基本的概念。计算机网络已发展成为复杂的规模庞大的系统，因此有必要用体系的方法来代替那些特定的网络设计方法，以便更好地理解计算机网络的细节。

一、网络体系结构基本概念

网络体系结构要解决的问题：

1. 网络应该具有哪些层次？每一层的功能是什么？（**分层与功能**）
2. 各层之间的关系是怎样的？它们如何进行交互？（**服务与接口**）
3. 通信双方的数据传输要遵循哪些规则？（**协议**）

1、网络通信协议

(1) **定义**：为进行计算机网络中的数据交换而建立的规则、标准或约定。

(2) **协议栈**：一套完整计算机协议的集合。

(3) 网络通信协议的**组成**（三要）：

语义：涉及用于协调与差错处理的控制信息。

语法：涉及数据及控制信息的格式、编码及信号电平等。

时序：涉及速度匹配和排序等。

1、网络通信协议

协议主要由**语法、语义、时序**三个要素组成：

1. **语法**：规定通信双方彼此“**如何讲**”，是用于规定将若干个协议元素和数据组合在一起来表达一个更完整的内容时所应遵循的格式，既确定协议元素的格式，即**用户数据和控制信息的结构和格式**。
2. **语义**：规定通信双方彼此“**讲什么**”，是指对构成协议的协议元素含义的解释，即确定协议元素的类型，即**需要发生何种控制信息以及完成的动作和做成的响应**。
3. **时序**：（定时、同步）规定通信双方彼此之间的“**讲的顺序**”，既**对事件实现顺序的详细说明**。

2、网络分层结构

(1) 为什么要分层？

计算机网络系统十分复杂；

将一个复杂系统分解为若干个容易处理的子系统，然后逐个解决；

分层就是系统分解的最好方法之一；

把一个复杂的系统设计问题分解成多个层次分明的局部问题，并规定每一层次所必须完成的功能。

分层使研究的系统抽象化；

2、网络分层结构

(2) 如何分层

将网络通信按逻辑功能分解成若干层来组织；

各功能层的关系是由低层向高层提供不同的可靠服务，高层使用低层提供的服务完成自己的功能，同时再向上一层提供服务；

上层可视为下层的**用户**，下层是上一层的**服务提供者**。

N 层是相邻低层 N-1 层的用户；

N 层是相邻高层 N+1 层的服务提供者；

N+1 层通过 N 层间接使用了 N-1 层及以下所有各层的服务。



网络通信系统设计举例

三个层次：

哲学家（认识层）：对通话的内容有共同的兴趣和认识，但使用不同的语言不能直接通话—最高层。

译员（语言层）：将他们各自的语言翻译成两个译员都懂得第三国语言—中间层。向高层提供语言翻译服务。

工程技术员（传输层）：按事先预定的方式（电话或电报）将交谈的内容转换成电信号在物理媒体上传送至对方—最底层，为中间层的译员提供传输服务。

网络通信系统设计举例

三个层次：

传送层（模块）：完成文件数据发送、接收前的准备工作和数据格式转换—最高层。

通信层（模块）：保证文件数据和通信控制命令在发、收双方之间可靠地传送—中间层，为文件传送层（上层）提供数据通信服务。

接入层（模块）：负责数据通信与网络的接口，利用公共通信网络将文件数据进行发送与接收—最底层，为通信层提供网络接口服务。

2、网络分层结构

(3) 分层原则

总体要实现的功能分配在不同的层次中。

把一组相近的功能放在一起，形成网络的一个结构层次。

网络中各结点都有相同的层次，相同的层次具有同样的功能；

网络中同一结点内相邻层之间通过接口通信；

每一层使用下层提供的服务，并向其上层提供服务；

不同结点的同等层按照协议实现对等层之间的通信。

每层的功能明确，并且相互独立。

当某一层的具体实现方法更新时，只要保持上下层的接口不变，便不会对相邻层产生影响。

层间接口必须清晰，跨越接口的信息量应尽可能少。

层数应适中。若层数太少，则多种功能混杂在一层中，每一层协议太复杂；若层数太多，则体系结构复杂，各层功能描述和实现困难。

2、网络分层结构

(4) 分层好处

每一层实现一种**相对独立**的功能。每一层不必知道下一层是如何实现的，只要知道下层通过层间接口提供的服务是什么及本层向上层提供什么样的服务，就能独立地设计。

每一层次的功能相对简单且易于实现和维护。

若某一层需要做改动或被替换时，只要不去改变它和相邻上下层之间的接口服务关系，则其它层次都不会受其影响。

有利于交流，理解和**标准化**。

3、网络协议的层次

(1) 为了保证各层功能的实现，通信双方的同等层之间必须遵守一些规则和约定，这些规则、标准和约定称为**同等层网络协议**，简称为**协议**。

(2) 整个网络的网络协议是由各个层次协议共同组成的。

(3) 各层协议只对所属层的操作有约束力，对其它层没有约束力。

(4) 通常，网络通信功能划分为几个层次，网络协议也相应划分为几个层次。

4、实体、接口、进程的概念

(1) 实体 (entity) :

就是任何可以发送或接收信息的硬件 (比如一块网卡) 或软件进程 (比如一个进程)。

实体是子系统中每一层的活动元素。

许多情况下, 实体是一个特定的软件模块。这样, 每一层都可以由若干个实体组成。

(2) 对等层实体 : 位于不同子系统的同一层内相互交换的实体。

4、实体、接口、进程的概念

(3) **接口**：同一节点**两相邻之间**的通信需要遵守的规则和约定。

相邻低层通过接口向高层提供服务。

接口可以是硬件也可以是软件。

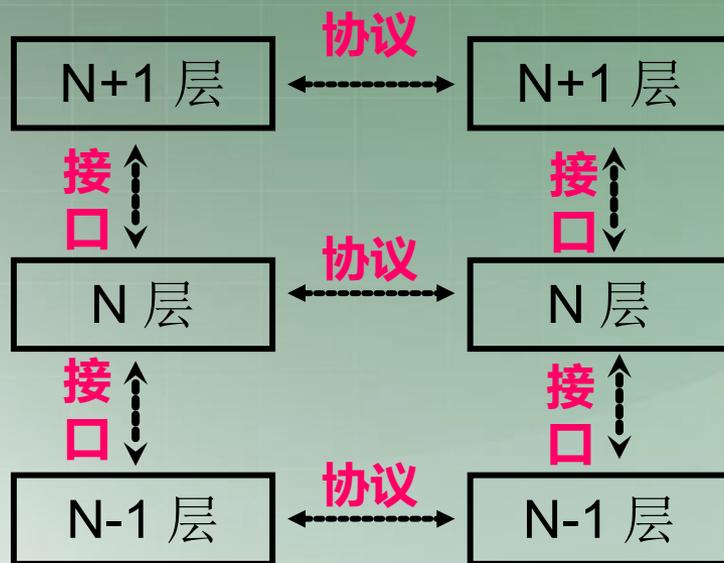
只要接口条件不变，低层功能不变，低层功能的具体实现方式与技术的变化不会影响整个系统的工作。

4、实体、接口、进程的概念

(4) 协议与接口不同：

协议是指同等层实体之间通信的一组规则。

接口是指相邻层实体之间通信的一组规则。



5、服务、服务访问点

(1) 服务：表示下层向上层提供的一种抽象的能力。

下层称为服务提供者。

上层称为服务的用户。

(2) 服务访问点 (**Service Access Point , SAP**) :

同一系统相邻两层实体交换信息的地方称为服务访问点 (SAP) 。

是一个确定的数据结构，定义了两个功能层次之间的交互所需的所有内容，两个相邻层之间的一切与“服务控制”相关的参数都通过这个地方来交换。某层SAP是相邻上层访问本层服务的地方，每个SAP有一个唯一属于它的地址。

(3) 服务与功能区别：服务是相对高一层来说的。功能是本层内部的活动，是为了实现对外服务从事的活动。

“进程”就是“程序的执行”或“一个在执行中的程序”

服务

服务 (service) : 网络中各层向其相邻上层提供的一组操作, 是相邻两层之间的界面。

由于网络分层结构中的单向依赖关系, 使得网络中相邻层之间的界面也是单向性的: 下层是服务提供者, 上层是服务用户。而服务的表现形式就是**原语 (primitive)**, 比如库函数或系统调用。这些原语供用户实体访问该服务或向用户实体报告某事件的发生。

在同一系统中, $N+1$ 层实体向 N 层实体请求服务时, 服务用户和服务提供者之间要进行信息互换, 互换的信息既为**服务原语**。

4 类服务原语

原语	意义
1. 请求 (Request) :	用户实体要求服务做某项工作
2. 指示 (Indication) :	用户实体被告知某事件发生
3. 响应 (Response) :	用户实体表示对某事件的响应
4. 确认 (Confirm) : 复	用户实体收到关于它的请求的答复

服务有“有确认”（证实服务）和“无确认”（无证实服务）之分。

有确认服务，包括“请求”、“指示”、“响应”和“确认”4个原语。

无确认服务只有“请求”和“指示”两个原语。

建立连接的服务总是有确认服务，可用“连接响应”作肯定应答，表示同意建立连接；或者用“断连请求”（DISCONNECT_request）表示拒绝，作否定应答。数据传送既可以是确认的也可能是无确认的，这取决于发送方是否需要确认。

服务

请考虑一下打电话邀请你的姑姑到家来喝茶的步骤：

为了使服务原语更易于理解，我们将考查一个简单的面向连接服务的例子。它使用了下述 8 个服务原语：

1) 连接请求：服务用户请求建立一个连接。

拨姑姑家的电话号码。

她家的电话铃响了。

2) 连接指示：服务提供者向被呼叫方示意有人请求。

她拿起电话。

3) 连接响应：被呼叫方用来表示接受建立连接的请求。

你邀请她来喝茶。

4) 连接确认：服务提供者通知呼叫方建立连接的请求已收到。

她听到了。

她说她很高兴来。

5) 数据请求：请求服务提供者把数据传至对方。

6) 数据指示：表示数据的到达。

你挂断电话。

你听到响铃停止。

7) 断连请求：请求释放连接。

8) 断连指示：将释放连接请求通知对等端。

你听到她接受邀请。

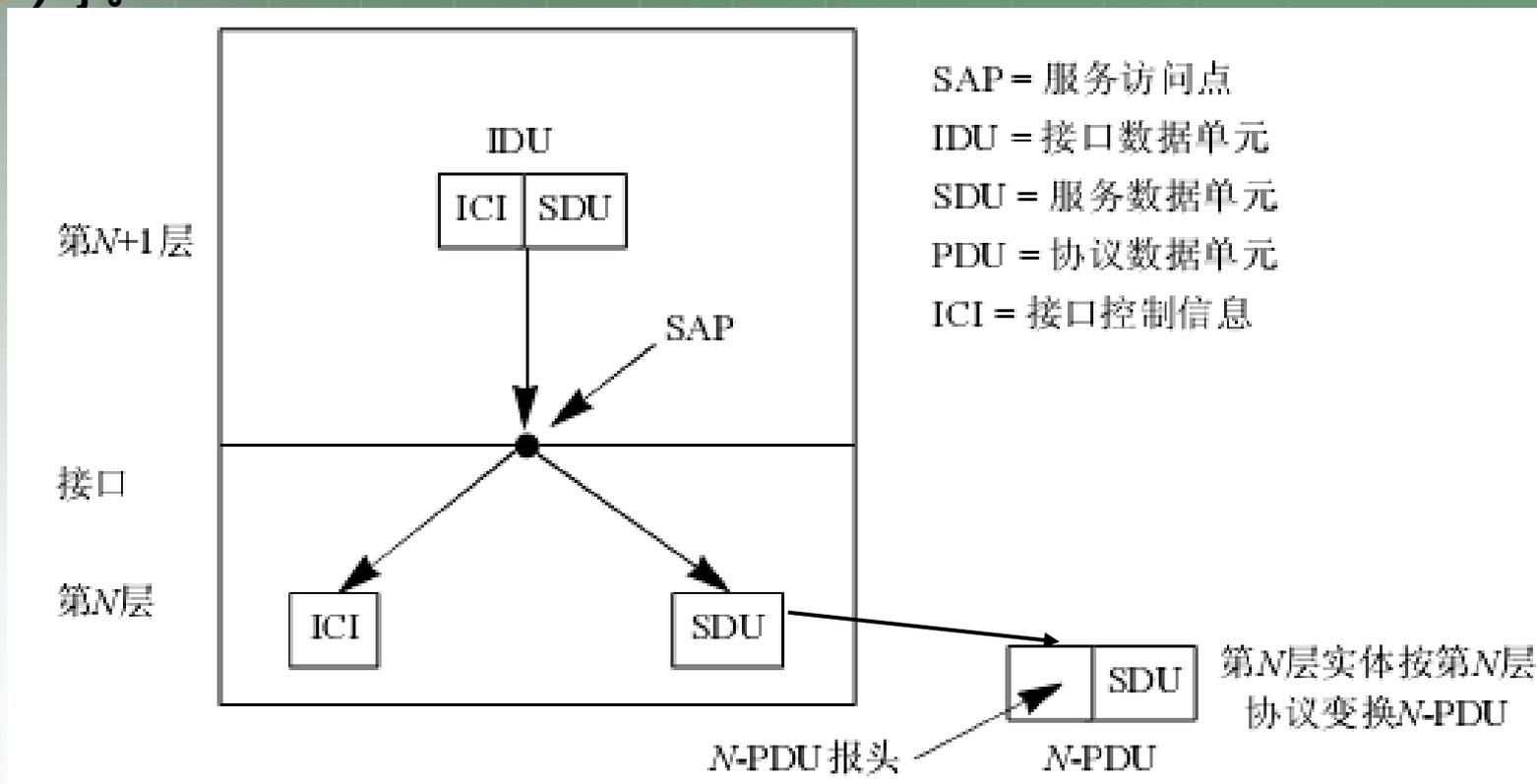
在本例中，连接是有确认服务（需要一个明确的答复），而断连是无确认的（不需要应答）。

她听到了，也挂断电话。

处于接口两边的两层之间的关系：

邻层间通过接口要交换信息。N+1层通过SAP把一个接口数据单元（IDU，Interface Data Unit）传递给N层，如图所示。

IDU = { 服务数据单元（SDU，Service Data Unit） + 接口控制信息（ICI） }。



➤ **接口数据单元 IDU** (Interface Data Unit)

IDU 是通过 SAP 进行传送的层间信息单元；

IDU 由上层的服务数据单元 SDU (Service Data Unit) 和接口控制信息 ICI (Interface Control Information) 组成；

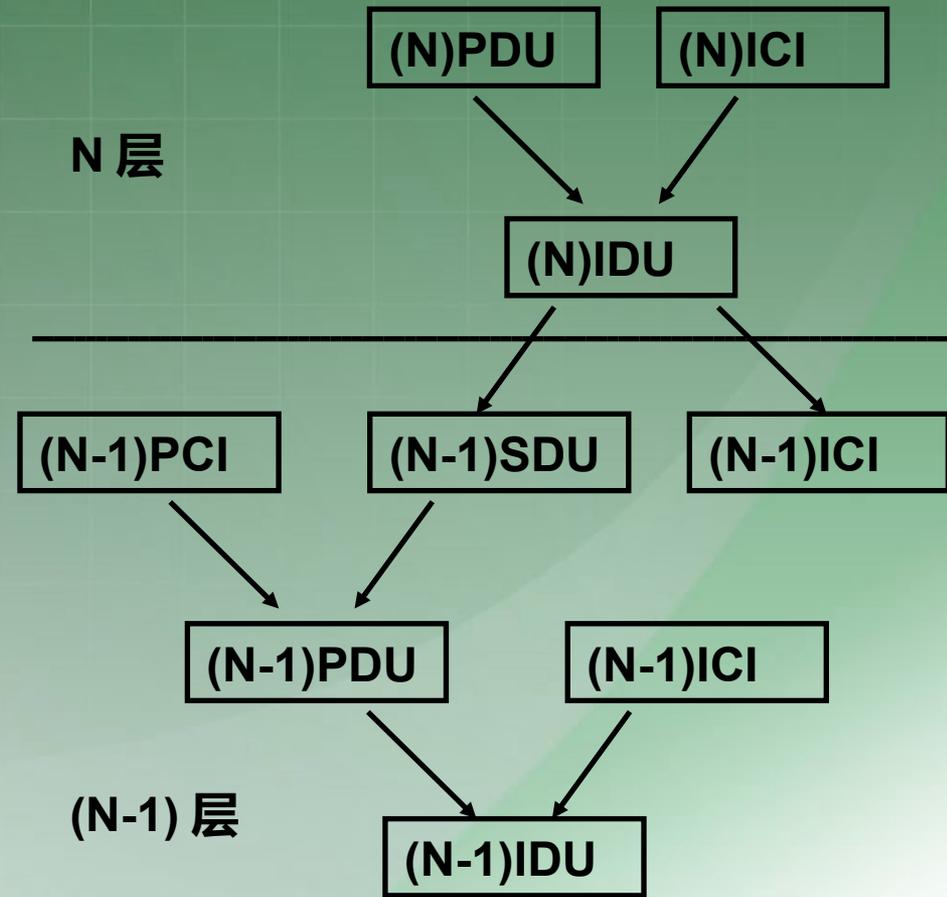
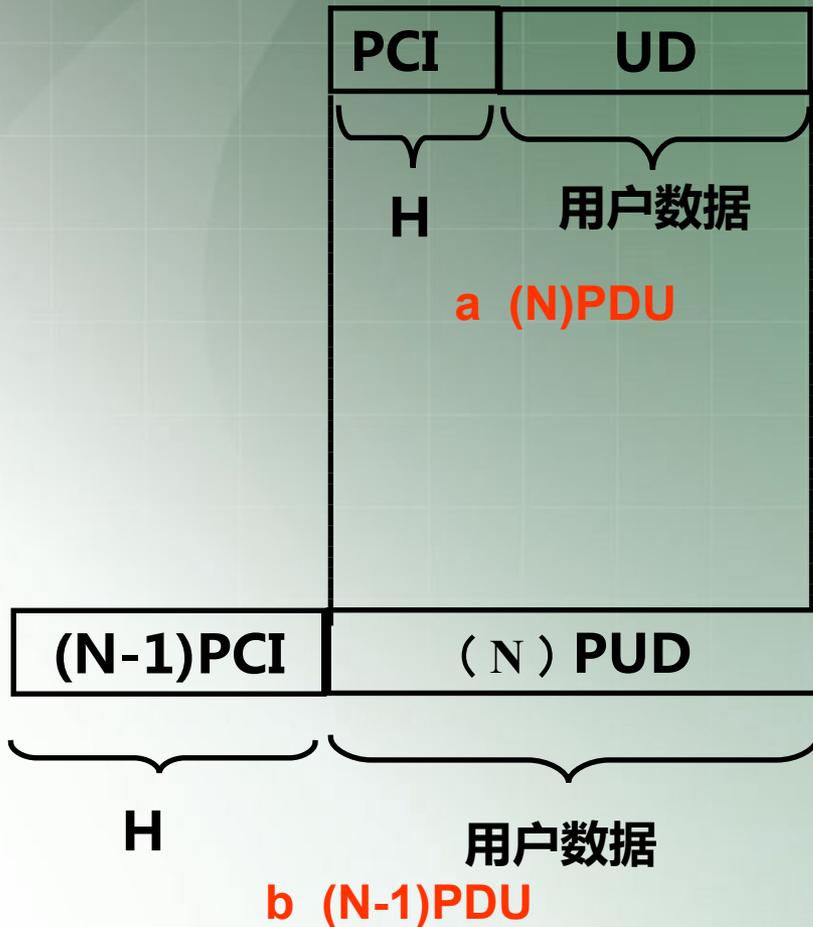
➤ **协议数据单元 PDU** (Protocol Data Unit)

把 N 层同等实体之间所传输的数据称为 N 层协议数据单元，用 (N)PDU 表示，PDU 由上层的服务数据单元 SDU 或其分段和协议控制信息 PCI (Protocol Control Information) 组成；

网络体系结构

☞	用户数据	UD	User Data
☞	协议数据单元	PDU	Protocol Data Unit
☞	接口数据单位	IDU	Interface Data Unit
☞	服务数据单元	SDU	Server Data Unit
☞	协议控制信息	PCI	Protocol Control Information
☞	接口控制信息	ICI	Layer Control Information

用户数据 UD、协议数据单元 PDU、接口数据单位 IDU、服务数据单元 SDU、协议控制信息 PCI、接口控制信息 ICI



服务形式：

在网络中，下层向上层提供的服务分为两大类：

1. 面向连接服务 (connection-oriented service)

是电话系统服务模式的抽象。每一次完整的数据传输都必须经过建立连接、数据传输和终止连接三个过程。在数据传输过程中，各数据包地址不需要携带目的地址，而是使用连接号。连接本质上类似于一个管道，发送者在管道的一端放入数据，接收者在另一端取出数据。其特点是接收到的数据与发送方发出的数据在内容和顺序上是一致

2. 无连接服务 (connectionless service)

是邮政系统服务模式的抽象。其中每个报文带有完整的目的地址，每个报文在系统中独立传送。无连接服务不能保证报文到达的先后顺序，原因是不同的报文可能经不同的路径去往目的地，所以先发送的报文不一定先到。无连接服务一般也不对出错报文进行恢复和重传。换句话说，无连接服务不保证报文传输的可靠性。

6、层次结构的要点

- (1) 不同系统之间各对等层实体间进行的是虚通信。
- (2) 对等层的虚通信必须遵循该层的协议。
- (3) N层的虚通信是通过N层与N-1层间的接口处，N-1层提供的服务以及N-1层的虚通信实现。

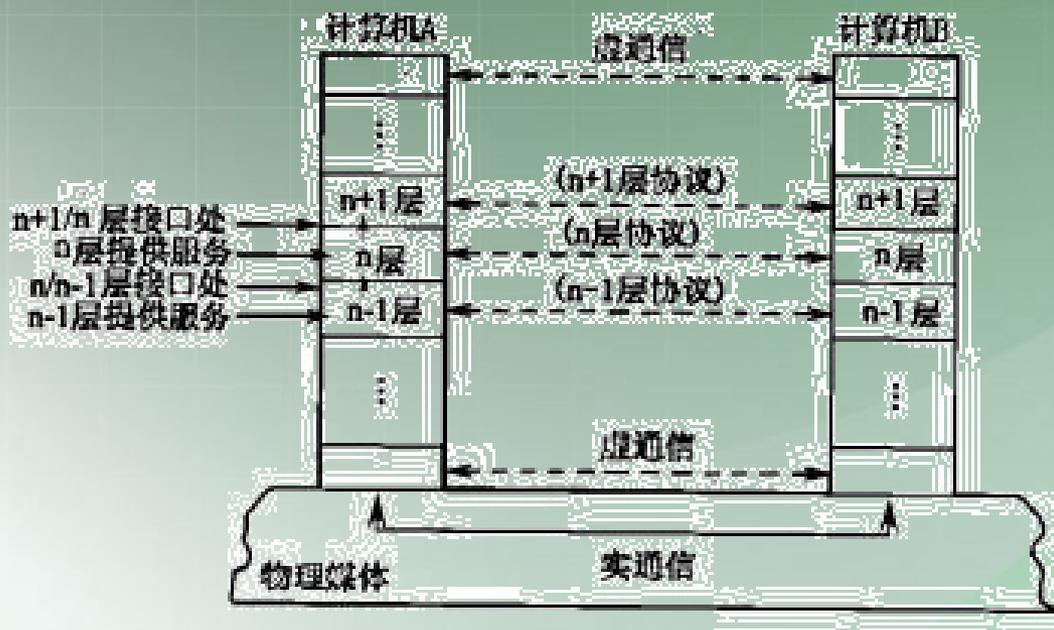
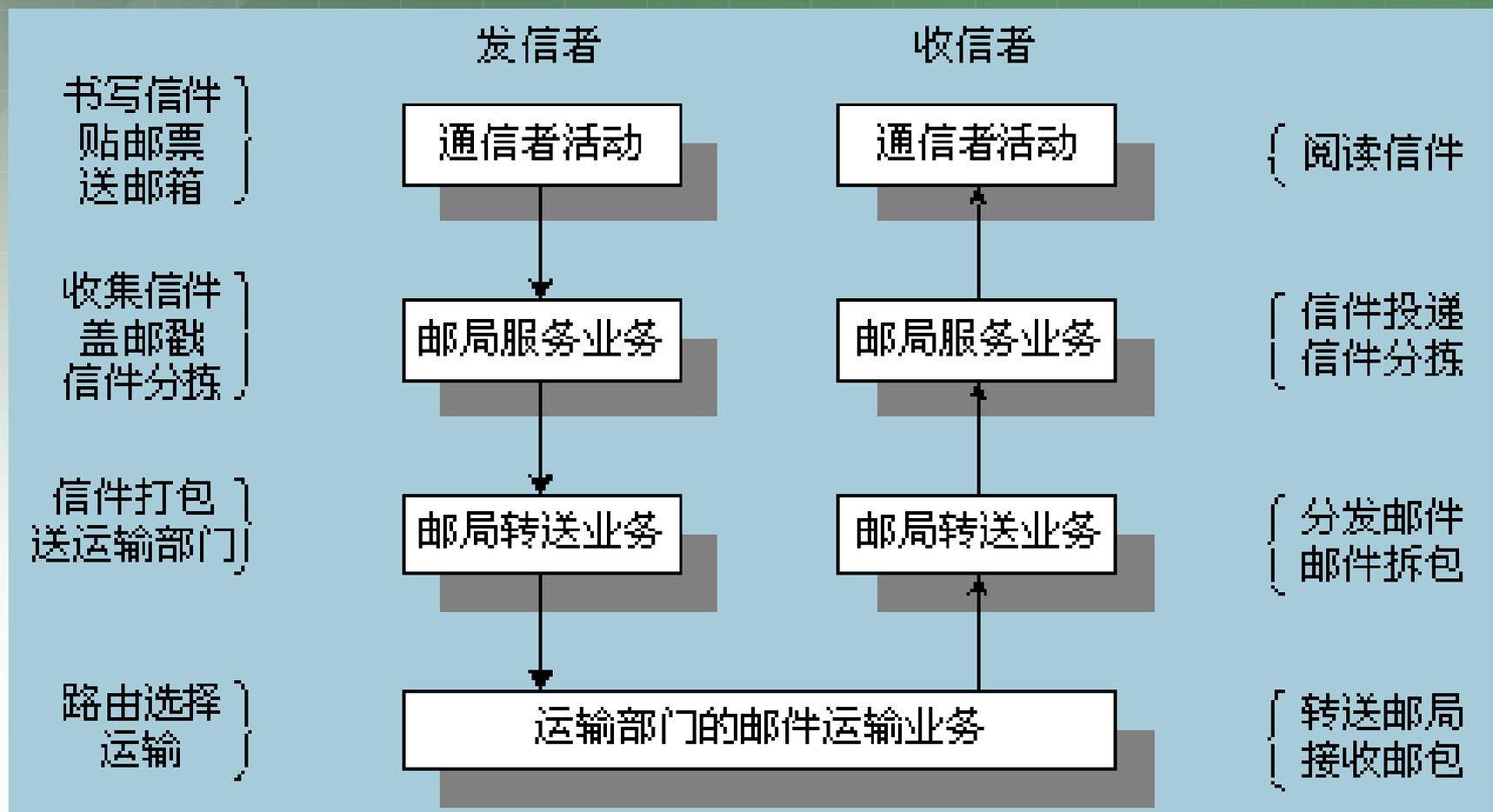
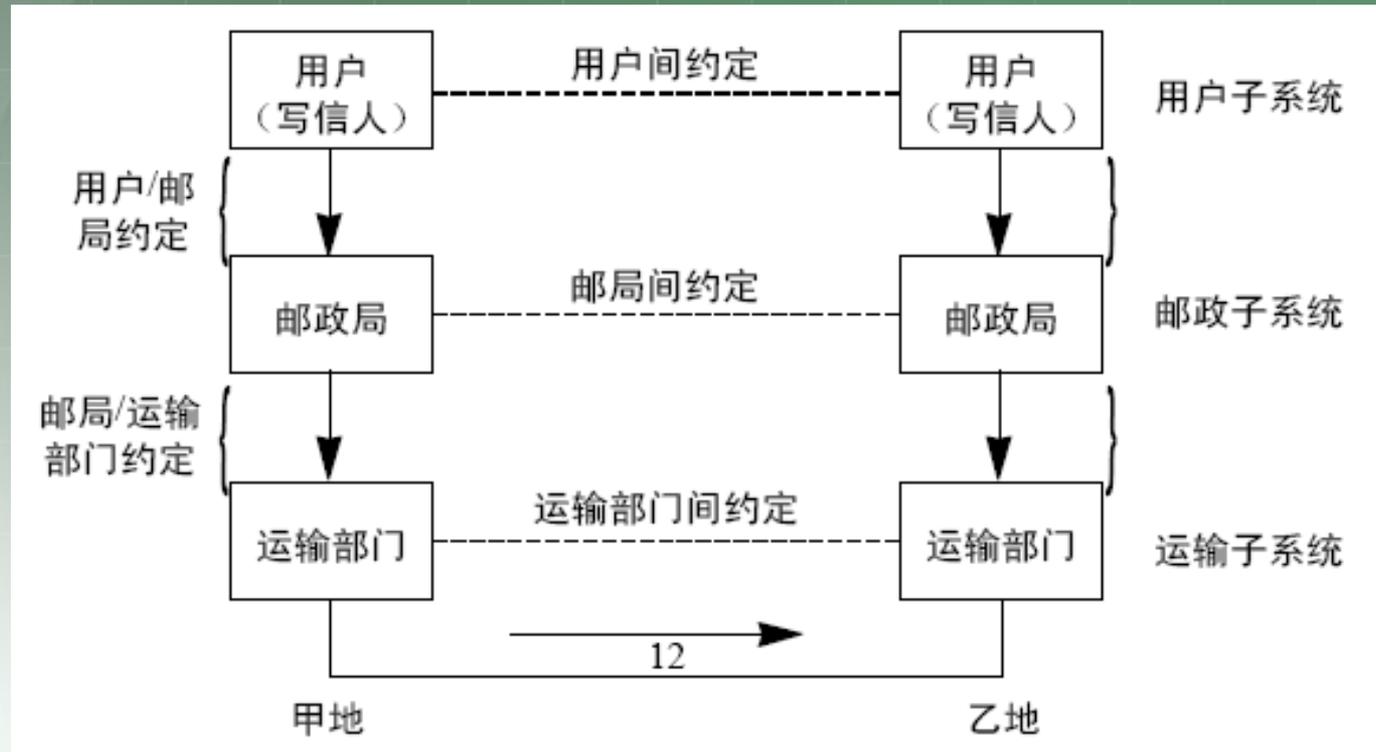


图 3.2 计算机网络的层次模型

网络通信系统设计举例



网络通信系统设计举例



邮政系统分层模型

网络通信系统设计举例

从上例可以看出，各种约定都是为了达到从源点送到目的点的这个目标而设计的。可以将这些约定分为：

- **同等机构的约定**（用户间约定、邮局间约定）；
- **不同机构间的约定**（用户与邮局间、邮局与运输部门间）

在计算机网络环境中，两个端点的两个进程之间的通讯过程类似于信件的投递过程。

- **用户进程**对应于用户；
- **计算机的通信进程**（或是专门的通信处理机）对应于邮局；
- **通信设施**对应于运输部门。

7、网络体系结构定义

定义：

计算机网络的体系结构：为了完成计算机间的通信合作，把每个计算机互连的功能划分成有明确定义的层次，规定了同层次进程通信的协议及相邻层之间的接口及服务。将这些同层进程通信的协议以及相邻层接口统称为网络体系结构。

计算机网络各**层次结构规模**以及各层**网络协议规范**和相邻**接口规范**的集合。

网络体系结构是一个抽象的概念，对于这些功能是由何种硬件和软件实现而未加说明。因此，网络体系结构与网络的实现不是一回事，前者是抽象的，仅告诉网络设计者“做什么”，而不是“怎样做”；而后者是具体的，是需要硬件和软件来完成的。

小结

网络体系结构实际上是一组设计原则，说明网络能够“做什么”，而不涉及具体的网络实现。

网络体系结构的概念：层次结构、协议、接口、服务等。

网络协议的三个要素：语法、语义、时序。

二、开放系统互连参考模型

产生的历史背景：

随着通信技术的发展，不同结构的计算机网络互连已成为人们迫切需要解决的问题。在这个前提下，开放系统互连参考模型 OSI 就提出来了。

为此，国际标准化组织 ISO 于 1977 年成立信息技术委员会 TC97 专门进行网络体系结构标准化的工作。

1984 年，国际标准化组织（ISO）发布了建立**开发系统互联参考模型（OSI RM，Open System Interconnect Reference Model）**，旨在提供通用的开放的数据协议开发规则。

该模型的主要作用：

1. 为计算机或网络厂商提供标准，使的不同类型的网络技术之间能够具有更好的兼容性和互操作性。
2. 目前大多数厂家仍然参照 OSI 来解释如何在网络上发送和接收数据。

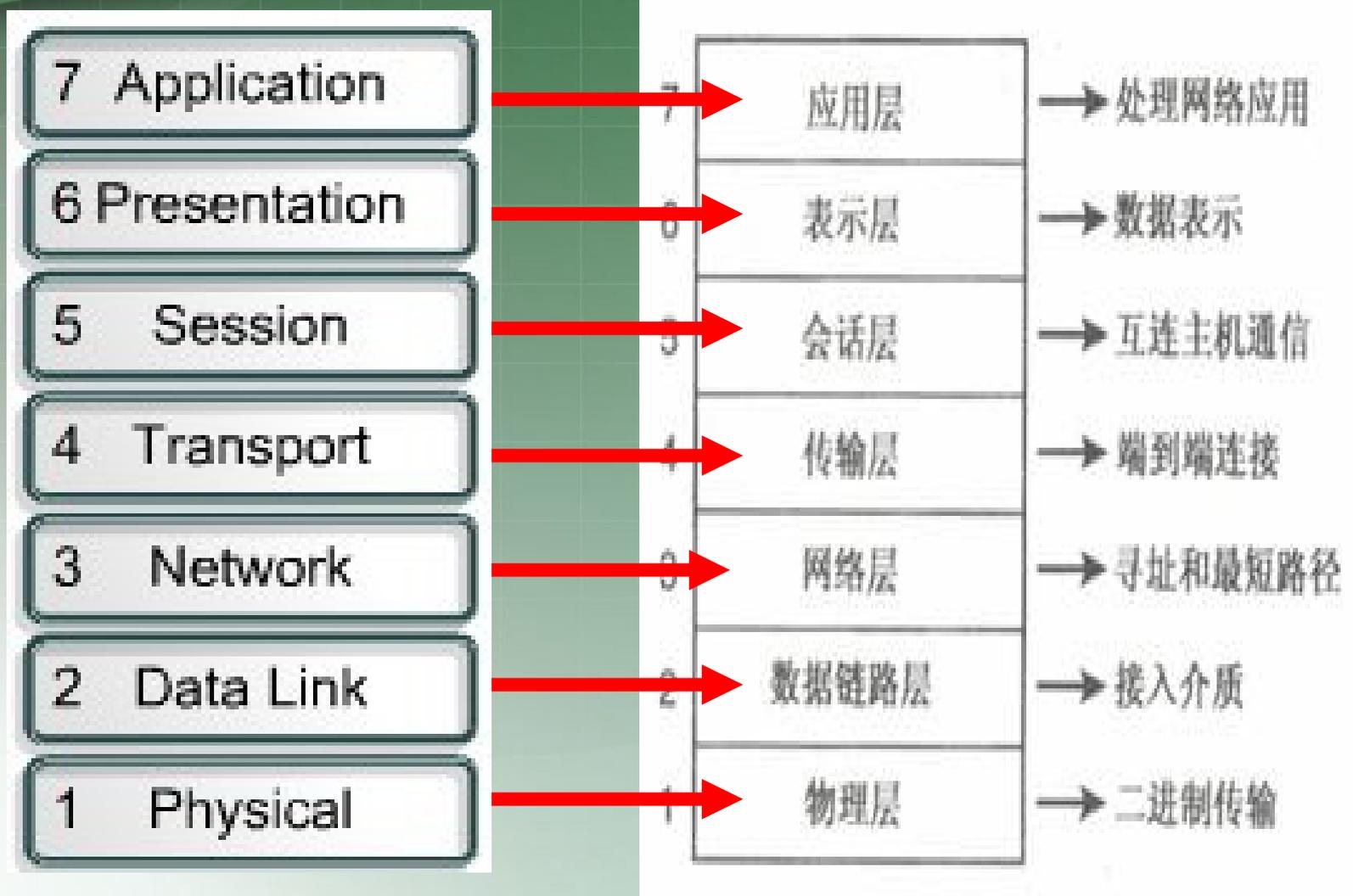
。

2、开放系统互连参考模型

OSI 开放系统互连

- 1、**开放**：表示能使任何两个遵守参考模型和有关标准的系统进行互连。
- 2、**开放式系统**：凡是通过计算机网络可以互相访问的计算机系统。
- 3、**开放系统互联**：计算机系统在网络中的互联性、互通性和互操作性。

2、开放系统互连参考模型

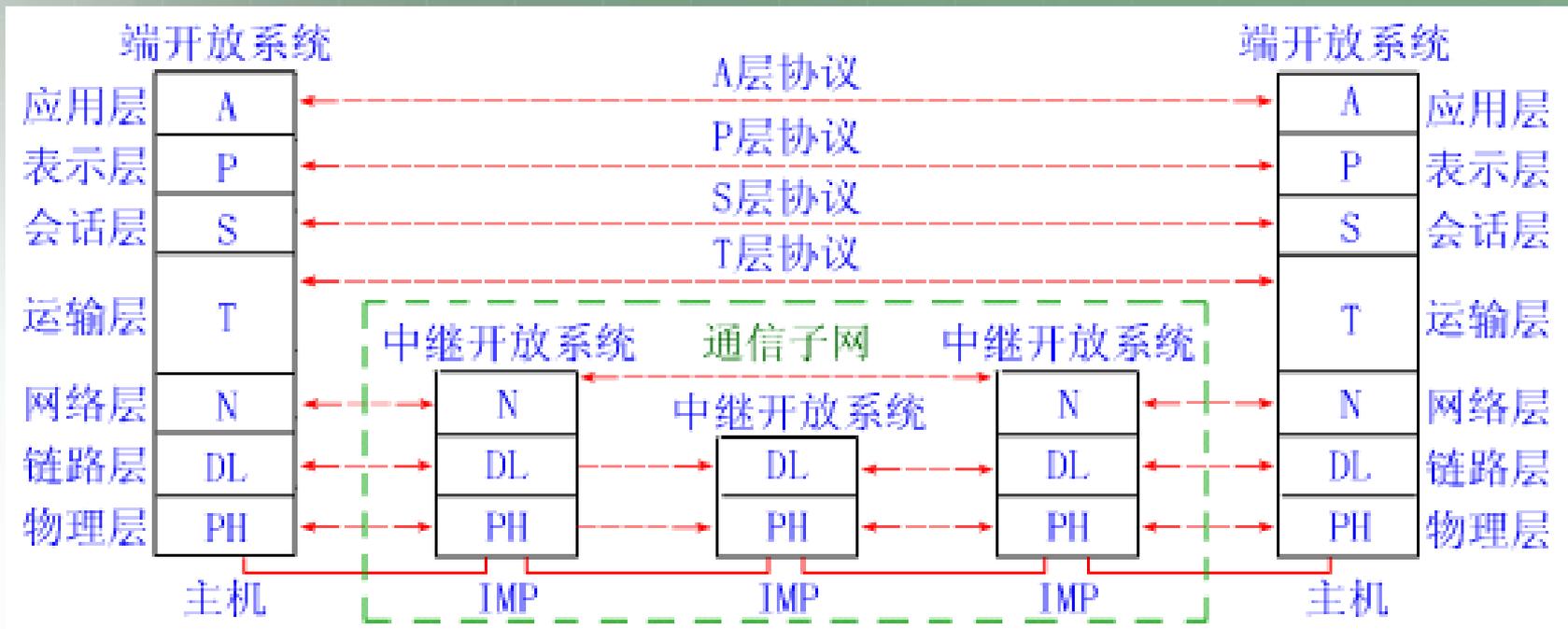


OSI 七层模型

整个开放系统环境由作为信源和信宿的端开放系统及若干中继开放系统通过物理媒体连接构成。

端开放系统相当于资源子网中的主机，中继开放系统相当于通信子网中的节点机 IMP。

主机可能需要包含所有七层的功能，通信子网中的 IMP 一般只需要最低三层甚至只需要最低两层的功能。



层次模型中各层功能

下面将简单介绍 ISO/OSI 层次模型中每层的主要功能

物理层

物理层是整个 OSI / RM 的最底层，是建立在物理介质上的。

作用：在物理媒体上传输原始的数据比特流。

当一方发送二进制比特 1 或 0 时，对方能正确地接收，并识别出来。

物理层

物理层是为在物理介质上建立、维持和终止传输数据比特流的物理连接提供的手段：

机械特征：

规定线缆与网络接口卡的连接头的形状、几何尺寸、引脚线数、引线排列方式、锁定装置等一系列外形特征。

电气特征：

定义了信号电压幅度、比特宽度、噪声容限和负载阻抗等电气参数（规定了在传输过程中多少伏特的电压代表“1”，多少伏特代表“0”）

功能特征：

规定了连接双方每个连接线的作用：用于传输数据的数据线、用于传输控制信息的控制线、用于协调通信的定时线、用于接地的地线。

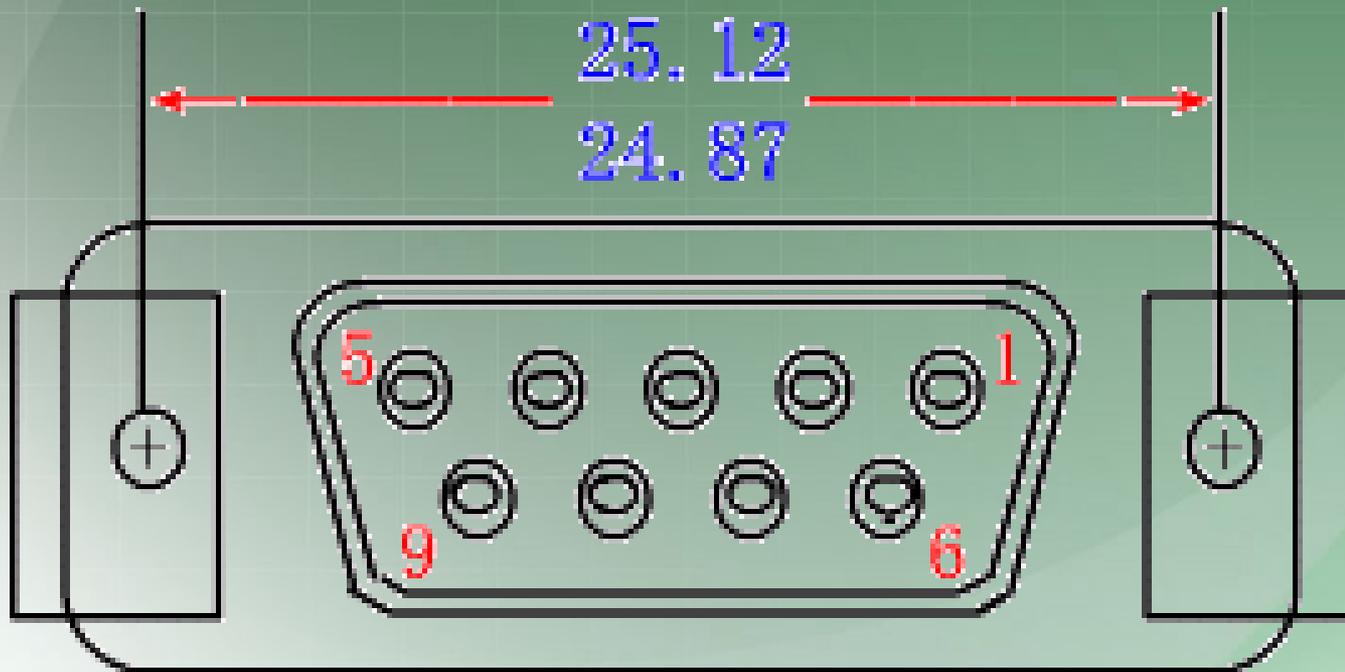
规程特征：

定义了信号的时序和规则，即具体规定了通信双方的通信步骤，如最初的连接如何建立，传输完成后连接又如何终止等。

—— 物理层协议的 4 个主要特性

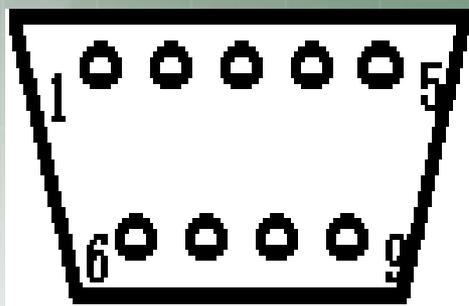
机械特性

机械特性：使用 9 针连接器，并对该连接器的尺寸及阵脚排列位置等做了确切的规定



9 针与 25 针串行口插针示意图

9 针



25 针



电气特性

RS-232C 电气特性

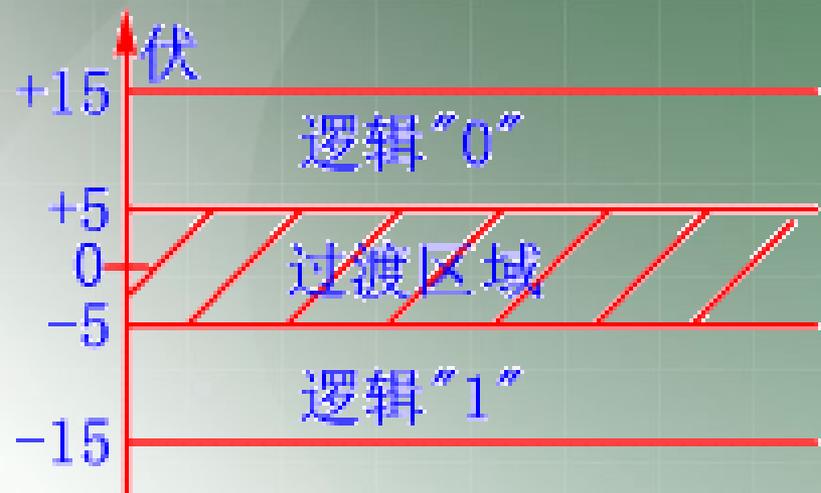


图3.9 RS-232C电器特性

表3.1 RS-232C电器信号表示

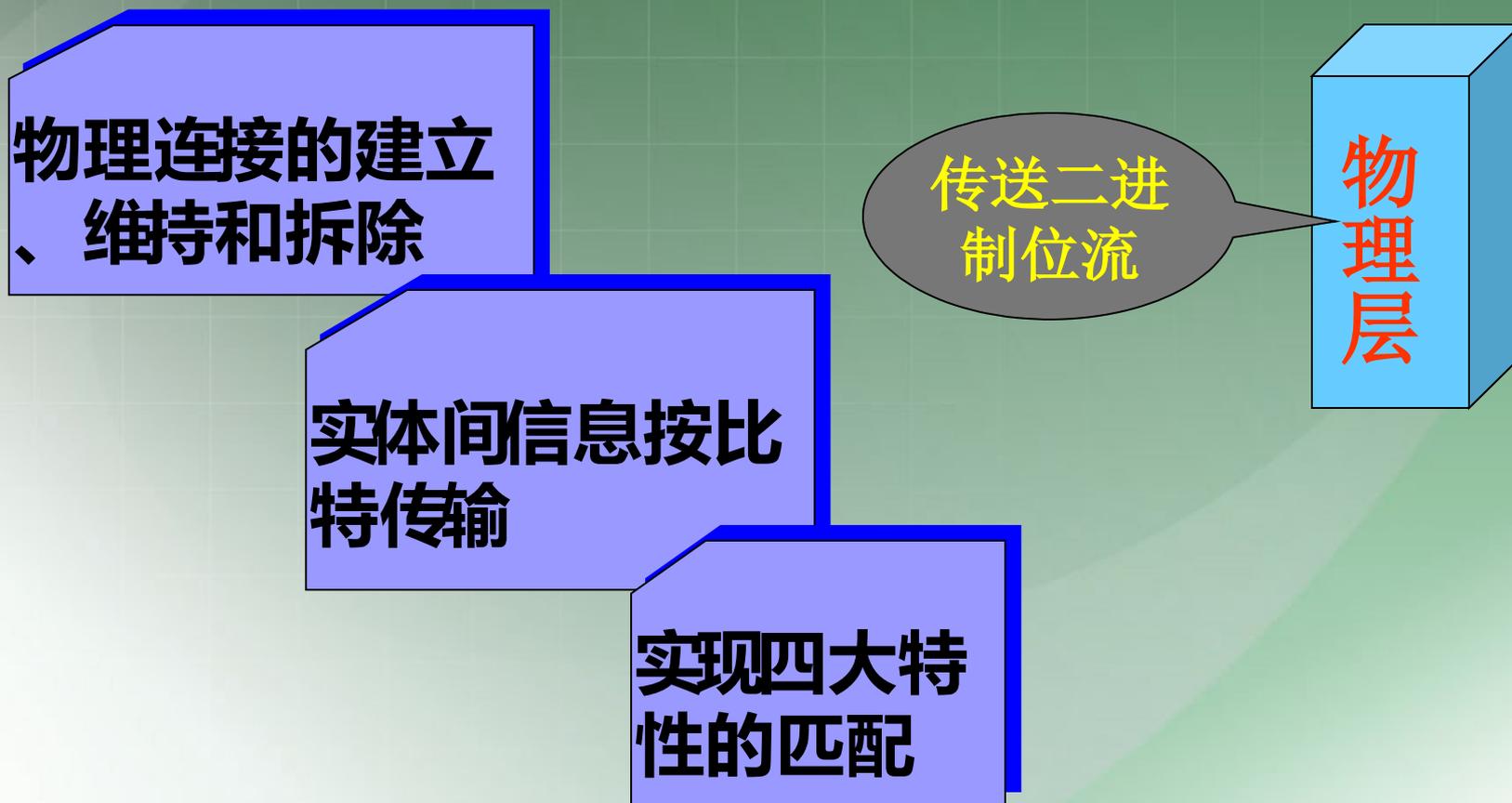
	负电平	正电平
逻辑状态	1	0
信号状态	传号	空号
功能状态	OFF (断)	ON (通)



功能特性



物理层的功能



数据链路层

作用：相邻结点之间的数据传输问题。

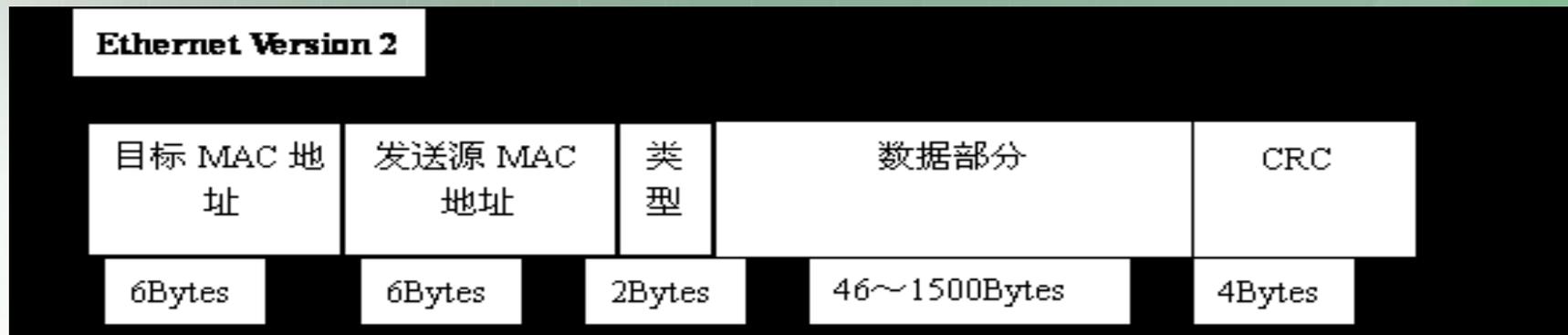
数据链路层在物理层提供的比特流服务基础上：

- 1、建立相邻节点之间的无差错的数据链路；**
- 2、解决流量控制问题；**
- 3、将比特组合成数据链路协议数据单元，即数据帧。**

功能：

1) **封装成帧**：把数据组成一定大小的数据块，我们称之为帧。然后以帧为单位发送、接收、校验数据，保持帧同步。

帧是数据链路层的数据传输单位，由地址段，控制段，数据段，差错控制段组成。



2) 物理寻址

在多点连接进行数据传输时要保证每一帧正确送到目标地址，接收方要知道谁是发送方。故数据链路层需要有寻址功能。

3) 流量控制：

对发送数据的一方，根据接收站的接收情况，实时地进行传输速率控制，以免出现发送数据过快，接收方来不及处理而丢失数据。对信息流的控制主要为了避免严重的过载和死锁。

4) 差错控制

在计算机通信中，一般都要求有极低的比特差错率。为此，广泛地采用了编码技术，同时采用校验、确认和反馈重发等手段，使用检错或纠错技术来确保正确的传输。

5) 链路管理

当网络中的两个结点要进行通信时，数据的发送方必须确知收方是否已经处在准备接受的状态。为此，通信的双方必须先要交换一些必要的信息。即必须先建立一条数据链路。同样地，在传输数据时要维持数据链路，而在通信完毕时要释放数据链路。

数据链路的建立，维持和释放就叫做链路管理。

数据链路层的功能

数据链路的
建立、维持
和释放

帧的同步传
输

差错检测
与控制

流量控制

传送帧

数据链路层

网络层

网络层：OS 中的第 3 层。

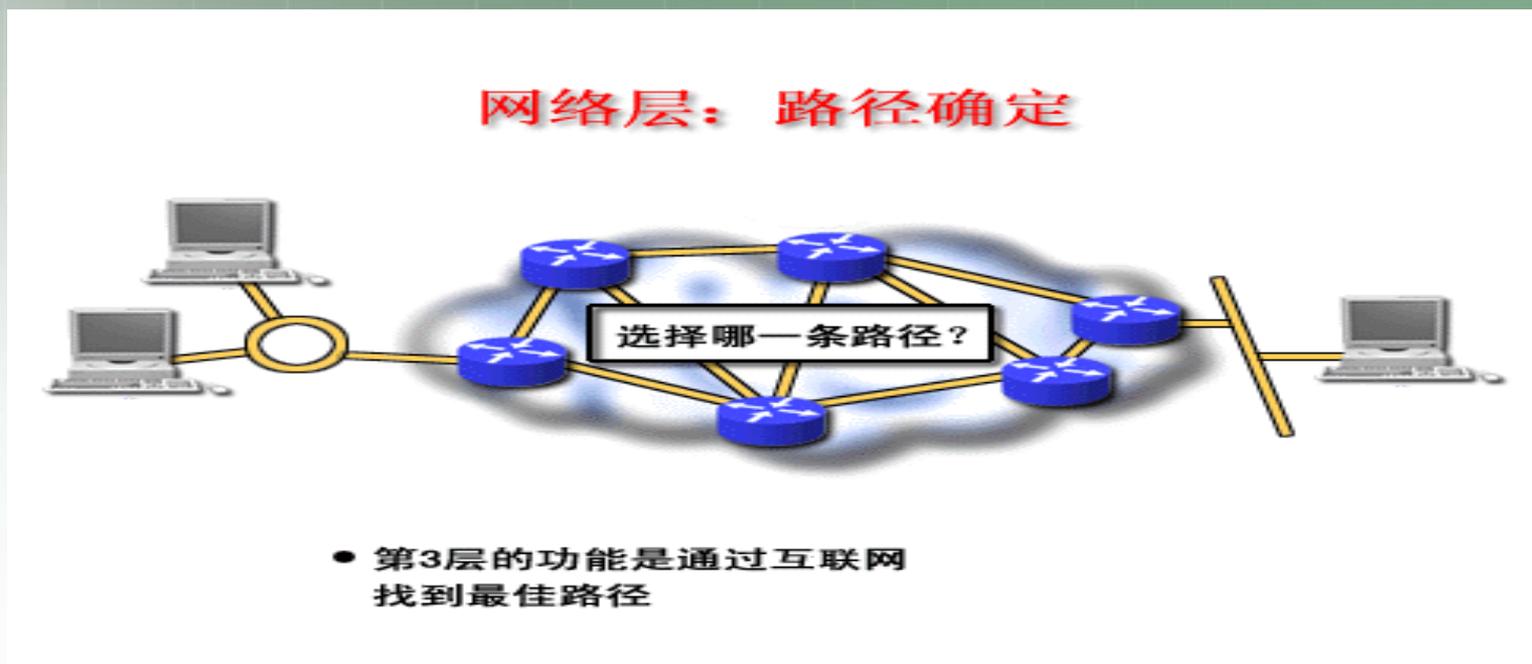
作用：如何把网络协议数据单元（分组）从源传送到目标。

网络层的主要功能：

1) 路径选择 (提供路由)

路径选择，又称为路由选择，它解决的问题是：

在具有许多节点的广域网里，通过哪条或哪几条通路，能将数据从信源发送到信宿主机上。即选择到达目标主机的最佳路径，并沿该路径传送数据包。主要依据 IP 地址寻径。



2) 流量控制、拥挤控制

太多分组涌入时，可能引起网络阻塞，使网络性能急剧下降，网络层必须进行拥塞控制。

- **数据链路层**的流量控制：是针对数据链路层相邻节点间进行的。
- **网络层**的流量控制：是对整个通信子网内的流量进行控制，是对进入分组交换网的通信量进行控制。

3) 包的拆分和重组

当数据包的大小比数据链路层允许的最大数据帧还要大时，网络层将其分成多个数据段，在接收端负责将多个数据段组合**数据包**

网络层的功能

用于控制两端点之间单个通信子网或多个互连子网的操作，为两端点之间的数据传输提供网络连接。

网络连接的
建立、维护
和拆除

组包 / 拆
包

传送分组

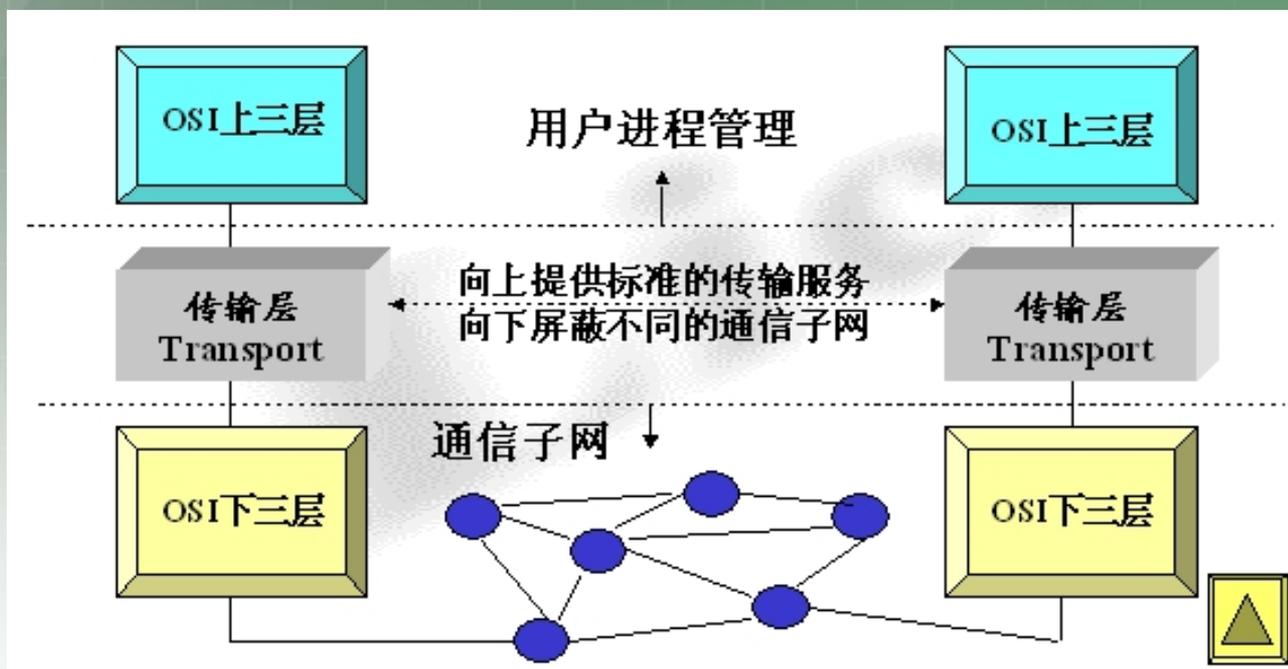
网络层

路径选择

流量控制

传输层

传输层是网络体系结构中高低层之间衔接的一个接口层。



传输层

传输层的目的是提供在不同系统的进程间数据交换的可靠服务，在网络内两实体间建立**端到端（主机到主机之间）**通信信道，用以传输**信息或报文分组**。

本层是面向应用的高层和与网络有关的下层协议之间的接口。它为上层提供与网络类型无关的可靠信息传送机制，对会话层遮蔽了下层网络的细节操作。

作用：

- 1、建立**端到端**通信；
- 2、为上层**优质透明数据传输**；
- 3、数据的**分组/重组**

会话层

作用：组织并同步不同主机上各种进程之间进行会话。

例如：WEB服务器在为多用户提供服务时，需要打开很多通信进程

可管理对话允许单工通信还是双工通信，并提供同步服务（如在数据流中插入同步点的机制，网络故障后仅重传最近的一个同步点以后的数据）。

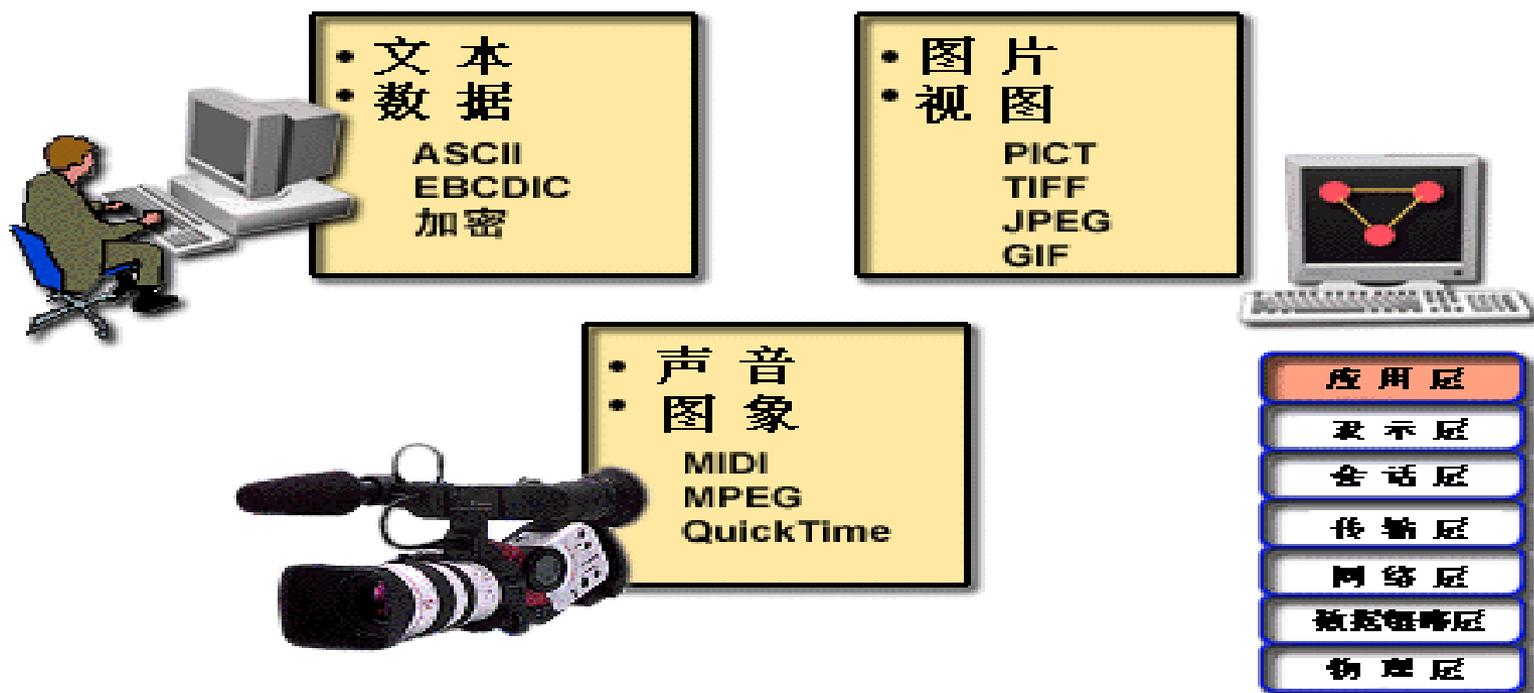
会话层为表示层提供服务：

表示层

➤作用：

- 1、为上层用户提供共同需要的数据或信息语法去表示变换。其目的在于解决格式、代码、数据和语法的转换，诸如 ASCII 码和 EBCDIC 之间的转换。
- 2、还提供数据的安全措施，如数据加密和数据压缩。

表示层



• 为应用程序提供编码格式和转换

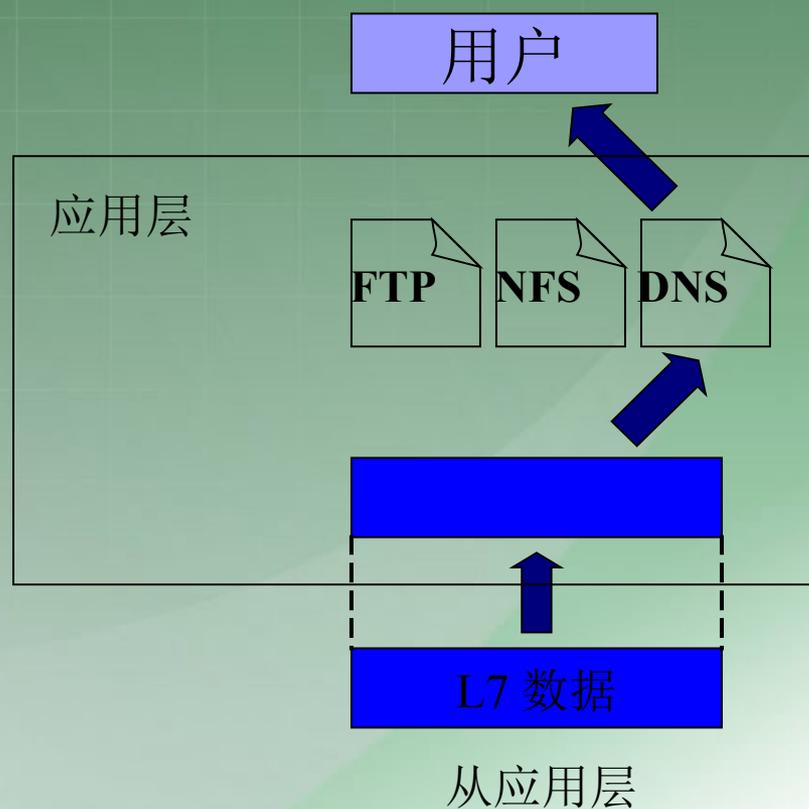
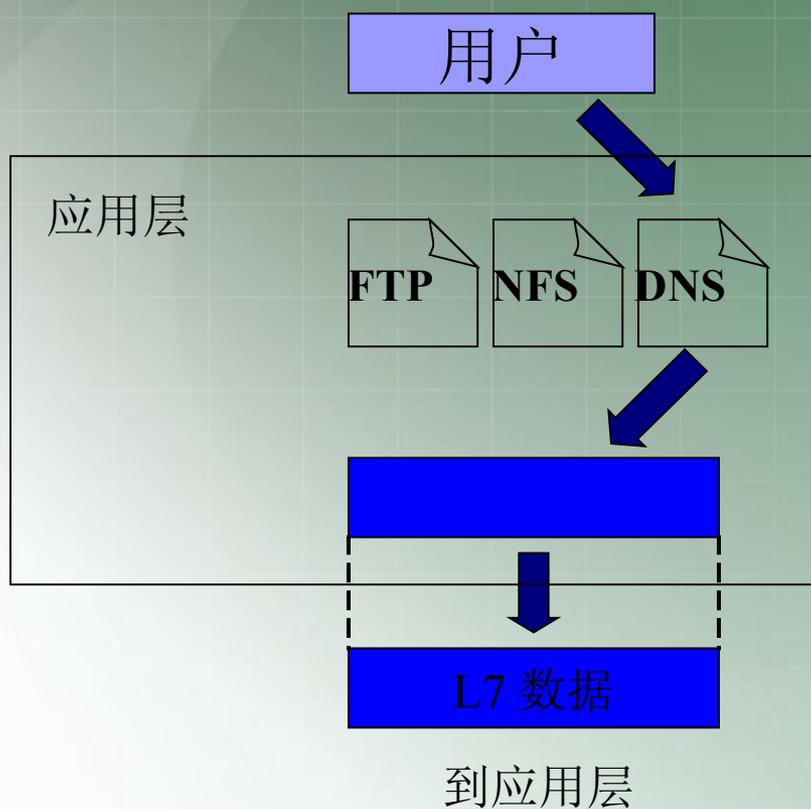
应用层功能

应用层是 OSI 参考模型的最靠近终端用户的一层。它只为存在于 OSI 模型之外的**应用进程提供服务**，负责管理和执行应用程序。

例如：文件传输、访问和管理、虚拟终端以及诸如电子邮件这样的一般文档和信息交换服务等。

此外，通过使用网络应用程序，应用层提供了与 OSI 模型中的其他层进行直接交互的界面（如浏览器、电子邮件、FTP 及 Telnet）

应用层功能示意图



OSI参考模型

◆ OSI层级功能

7	应用层	提供网络与用户应用软件之间的接口服务
6	表示层	提供格式化的表示和转换数据服务
5	会话层	提供包括访问验证和会话管理在内的建立和维护应用之间通信的机制
4	传输层	提供建立、维护和取消传输连接功能，负责可靠地传输数据
3	网络层	处理网络间路由，确保数据及时传送
2	数据链路层	负责无错传输数据，确认帧、发错重传等
1	物理层	提供机械、电气、功能和过程特性



应用层：为应用程序提供网络服务应用层：

表示层：数据表示

会话层：互连主机通信

传输层：端到端连接

网络层：确定地址和最佳路径

数据链路层：介质访问

物理层：二进制传输

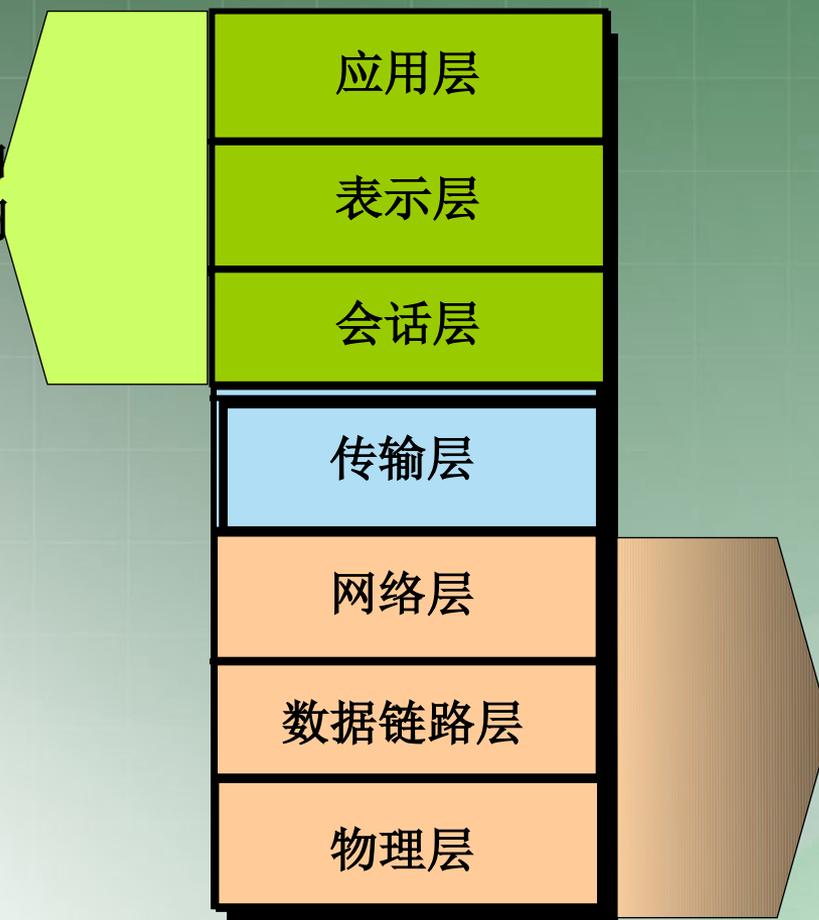
OSI 参考模型

应用层（高层）

功能：为处理用户接口、数据格式及应用访问。

主要由操作系统实现。

实现资源子网的功能，实现端到端的通信。

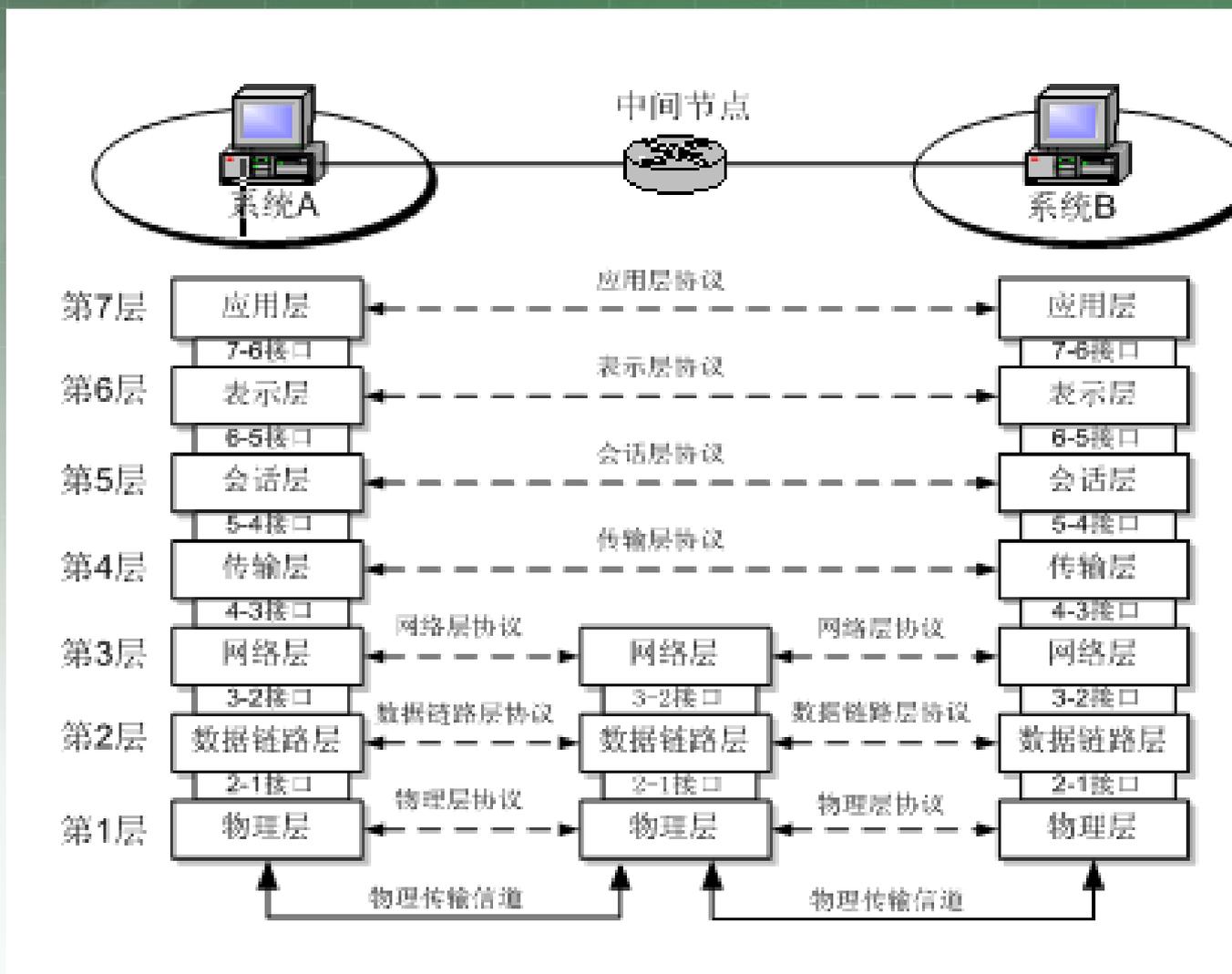


数据流层（低层）

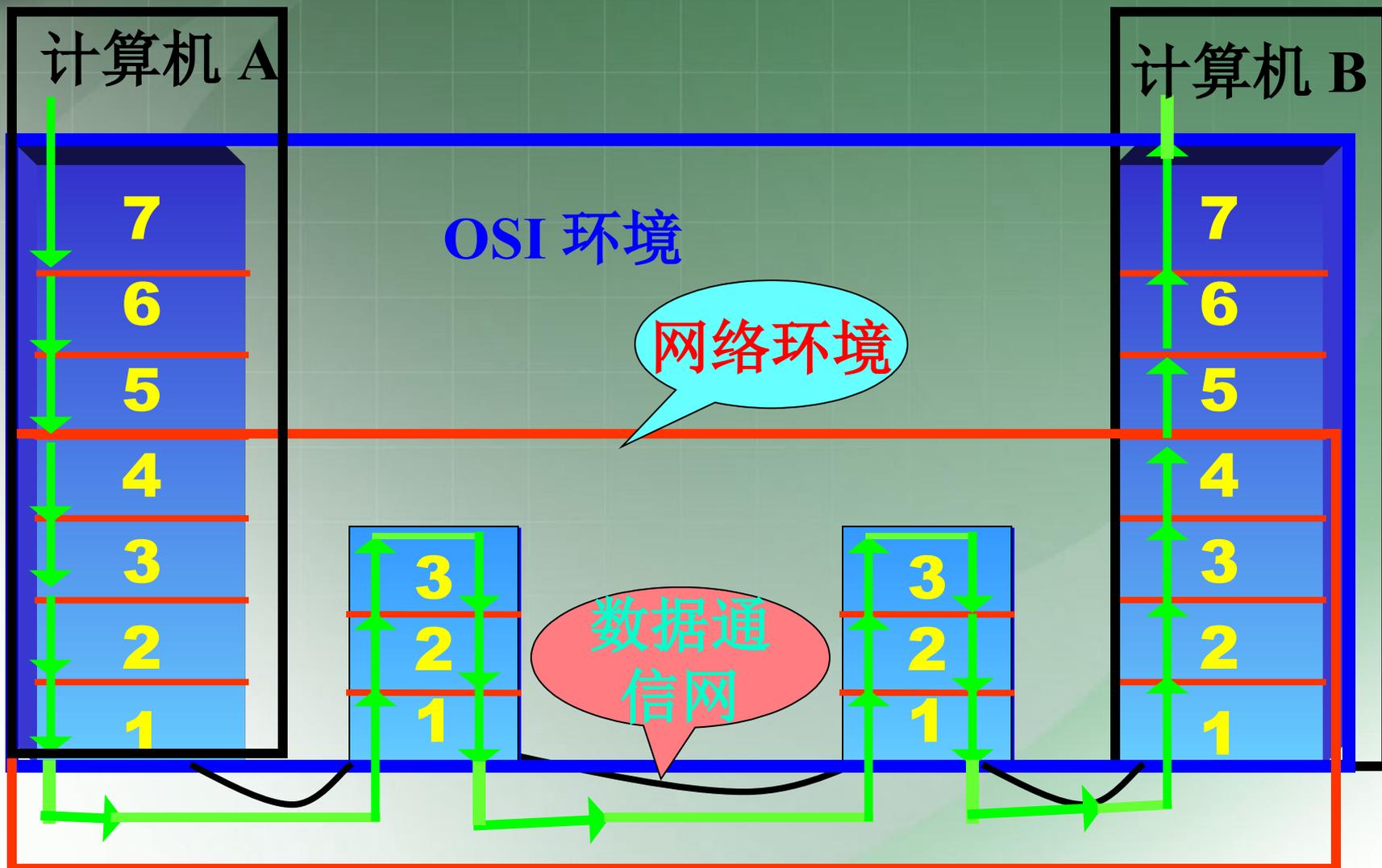
定义了数据如何在网络传输介质之间传送，及数据如何通过网线和网络设备传输到期望的终端。

下 3 层协议实现通信子网的功能，主要由与网络互连设备实现，实现点到点的通信

OSI 模型中数据的实际传送过程



OSI 模型中数据的实际传送过程



3.2.2 OSI 数据流过程

数据封装：

OSI 模型的每一层用其自己的协议与目的设备的相同层进行通信。为了交换信息，每层都使用协议数据单元（ PDU ）。 PDU 包括控制信息和用户数据。

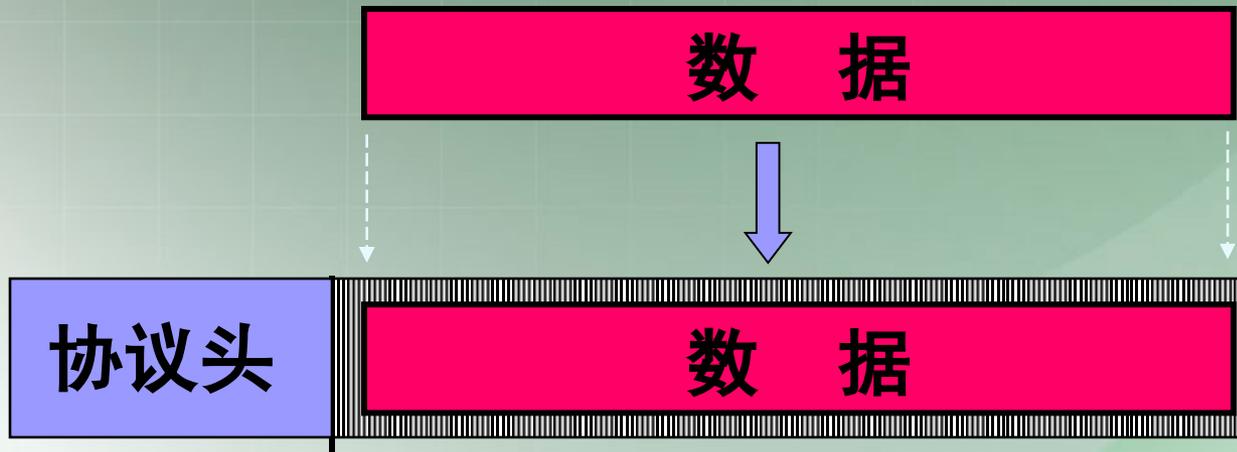
将控制信息添加到一个 PDU 的过程称作封装。

3.2.2 OSI 数据流过程

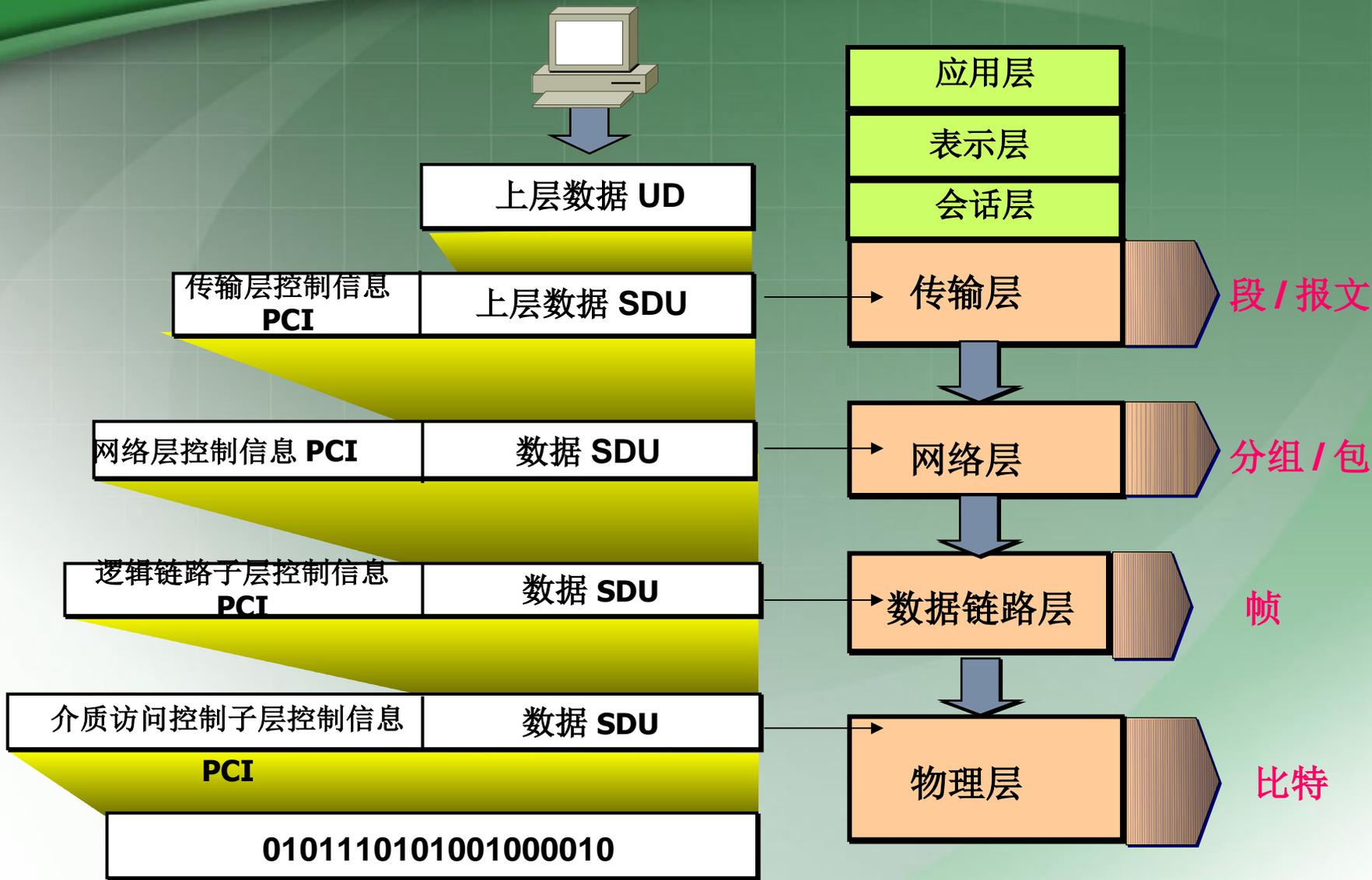
数据封装

一台计算机要发送数据到另一台计算机，数据首先必须打包，打包的过程成为封装。

封装就是在数据前面加上特定的协议头。



封装



3.2.2 OSI 数据流动过程

解封装数据：

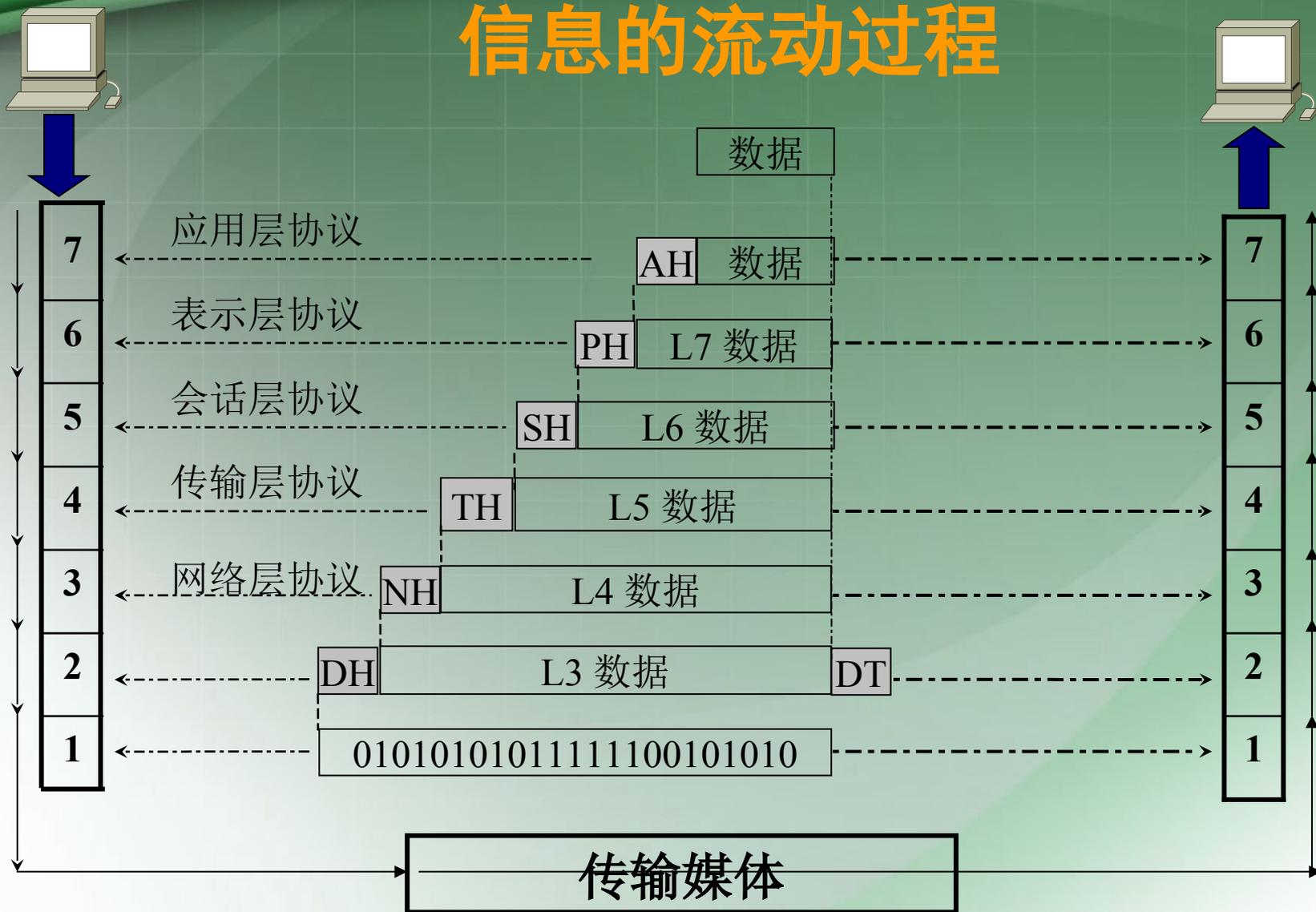
当一个远端设备收到一串比特数据时，它完成以下工作：读出同层源设备提供的控制信息，从 PDU 中剥离控制信息，再传给上一层，同时执行在帧的控制域给出的指令。

这样，将控制信息从 PDU 剥离的过程就称作解封装。

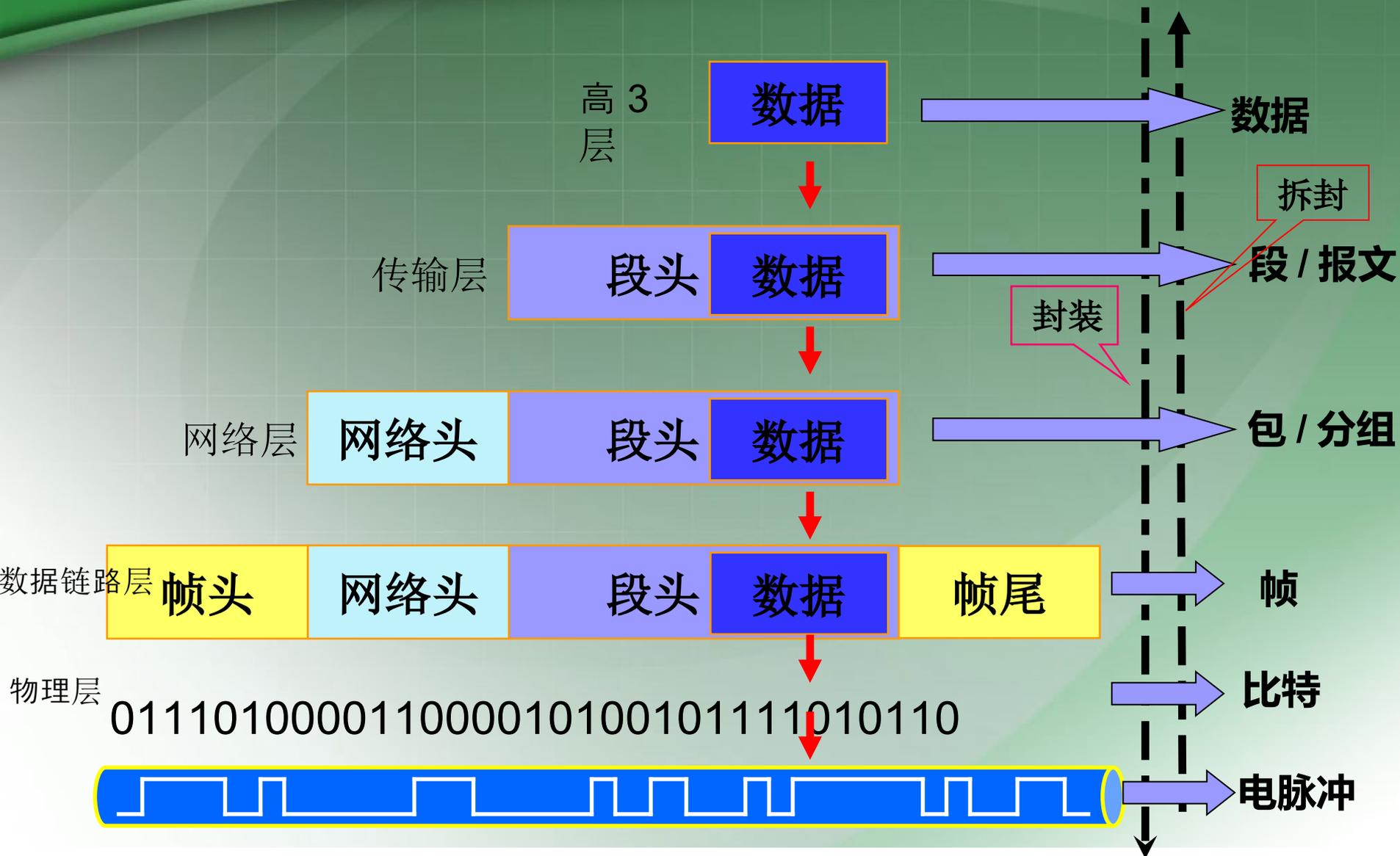
解封装



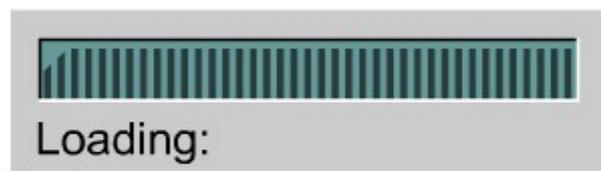
信息的流动过程



数据多层封装



数据封装



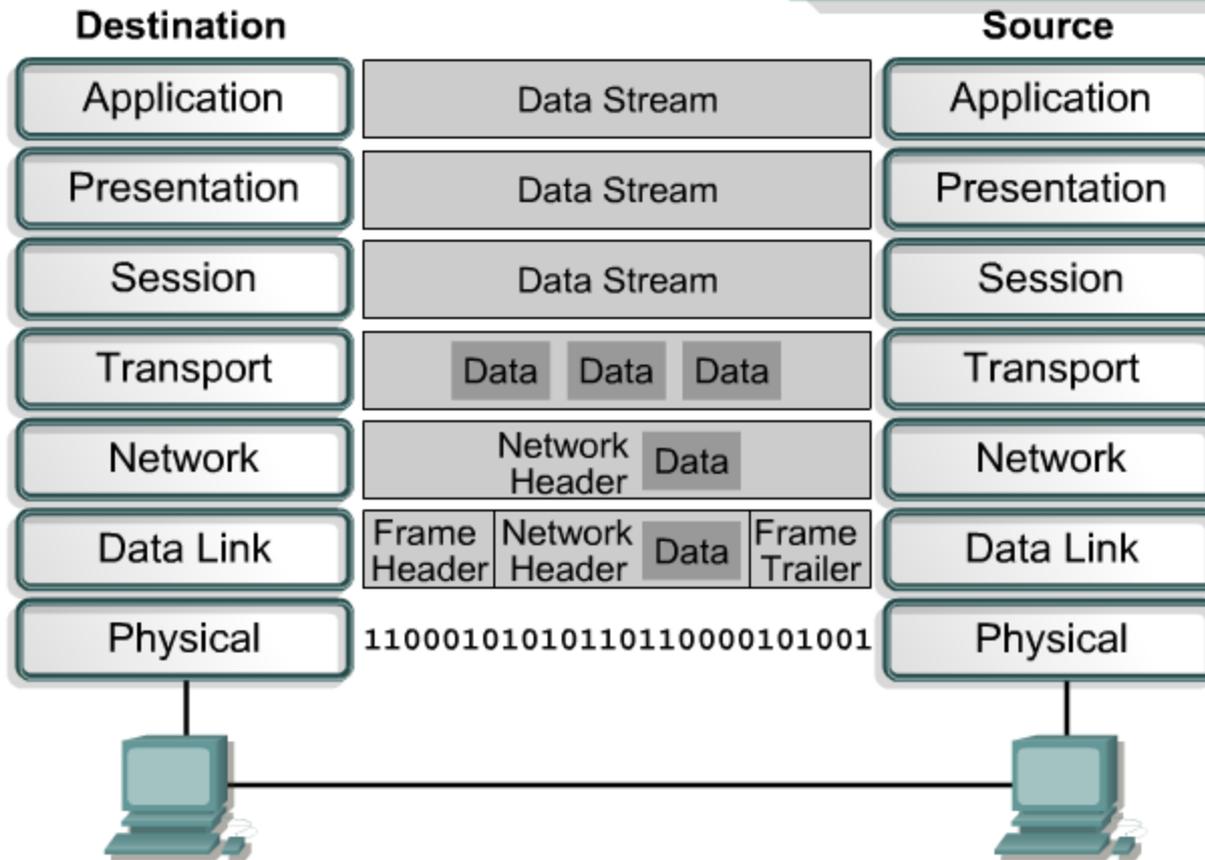
Data Encapsulation

FIGURES

1

2

Toolbar: Roll over tools



举例

设想：主机 A 想利用 E-mail 发送以下信息给主机 B：

“好久不见了，你还好吗？”

数据封装会将此消息传送到合适的目的地，在数据封装过程中会经过以下 5 个转换步骤：

步骤一：当用户发送一个消息时，它的字符转化为数据，从第七层向下传送到第五层，然后在网络上传输；

步骤二：在第四层通过使用段，传输功能把数据进行打包，使它能用于网络传输，并确保系统两端的消息主机能可靠通信；

步骤三：数据在第三层被放入一个数据包（或数据报）中，它有一个包含有源和目的逻辑地址的包头。然后，网络设备沿着一个选定的路径在网络上发送这些数据包。

步骤四：每个网络设备在第二层必须把数据包放入一个帧。帧可以连接到同一链路上的一个直连设备。在一个选定的网络路径上的每一个设备都要求用帧来连接下一个设备。

步骤五：在第一层，帧必须转换成可以在介质（导线）中传送的“1”和“0”模式。时钟功能使得设备能区分在介质中传输的比特流。

OSI 分层的优势

- 降低协议设计的复杂性：

各层相对独立，各层不必关心其它层的具体实现，只需知道上下层接口；

- 标准化接口：

任何一层发生修改，只要接口关系不变，其它层均不受影响；

- 模块化结构：

结构分层，各层都可独立实现；便于网络模型的设计；

- 保证技术的互操作性：

各层向下单向依赖，易于实现和维护

- 不同厂商生产设备时有一个共同操作的标准，使不同产商设备互相兼容。