

2-3. 开放系统互连参考模型

开放系统互连参考模型 (OSI/RM : Open Systems Interconnection Reference Model) **是由国际标准化组织 (ISO) 提出和定义的计算机网络国际标准。**

- **OSI** 目标: 全世界遵循这一标准的计算机都能很方便的进行互连和交换数据。
- **RM**: 只说明了需要做什么 (WHAT TO DO) 而未规定怎样做 (HOW TO DO), 即 OSI/RM 只是一个 参考模型, 它并不是 网络协议, 它也不规定每层应该采取什么协议



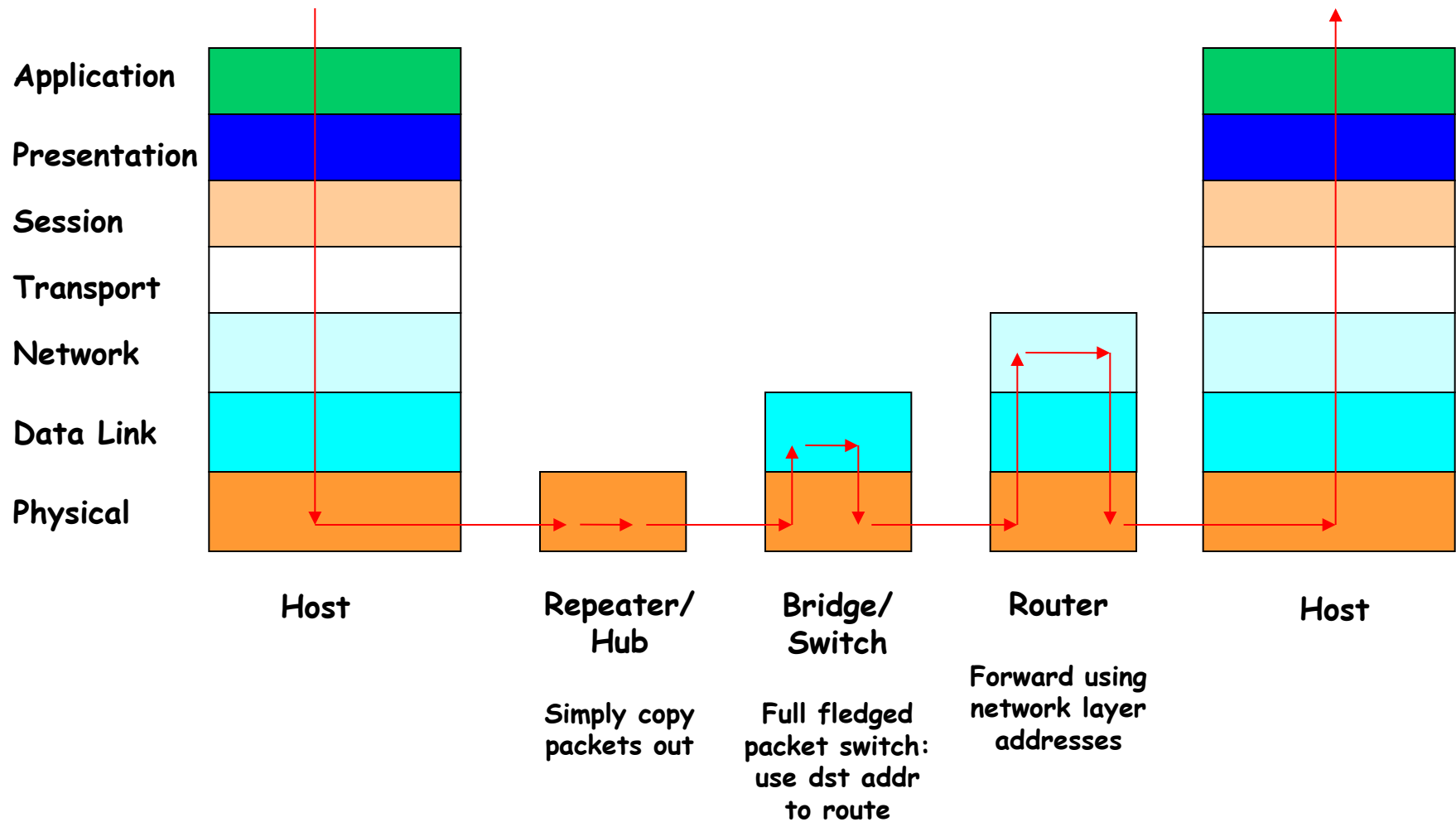
注意区分:

ISO : 国际标准化组织
OSI : 开放系统互连

OSI参考模型



OSI Layers and Locations



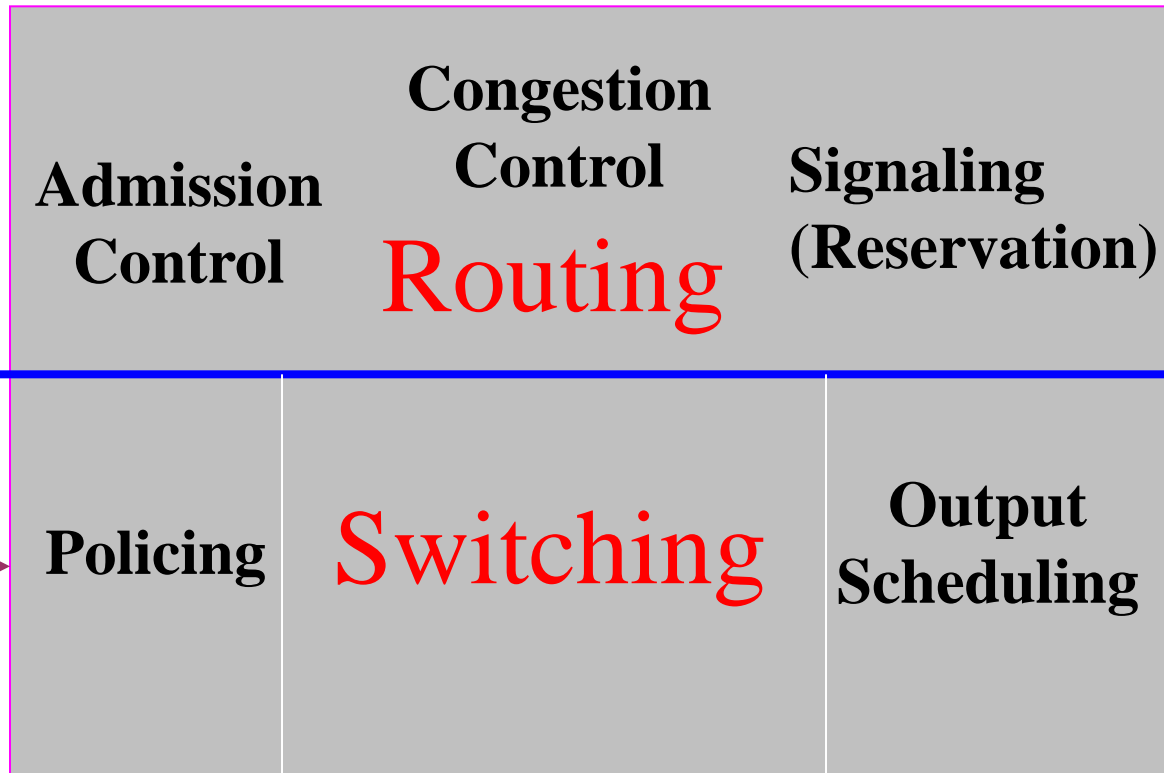
需要说明的问题

- 1.交换机属于哪一层的设备？
- 一般所说的交换机属于2层交换--链路层设备
- 但有的交换机具有路由功能(路由交换机)----网络层设备
- 因此，不能简单地說交换机属于哪一层的设备，要根据具体情况而定

需要说明的问题

- 2. 路由器属于哪一层的设备?
第3层?
第7层?

Basic Architectural Components



控制平面：（很多协议均在应用层实现）。需要所有7层的协议

Control Plane

Data Plane

数据平面：负责数据传输,只需要路由、转发等功能,所以只需要低三层协议

带外信令的概念
(**Out-of-band Signaling**)

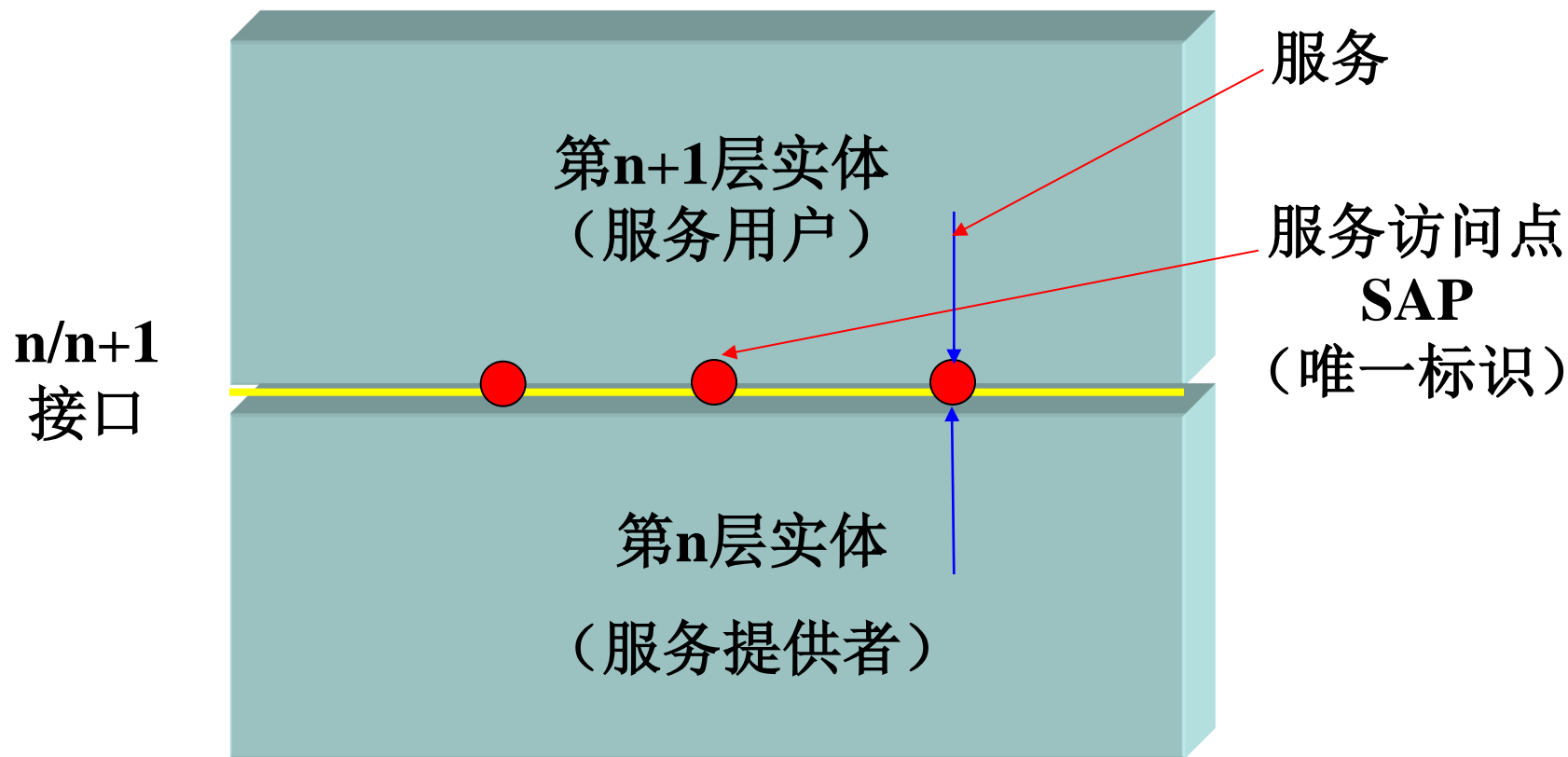
4. OSI/RM要点

**分层、层接口、层功能、
服务、协议、服务访问点、对
等通信、物理通信**

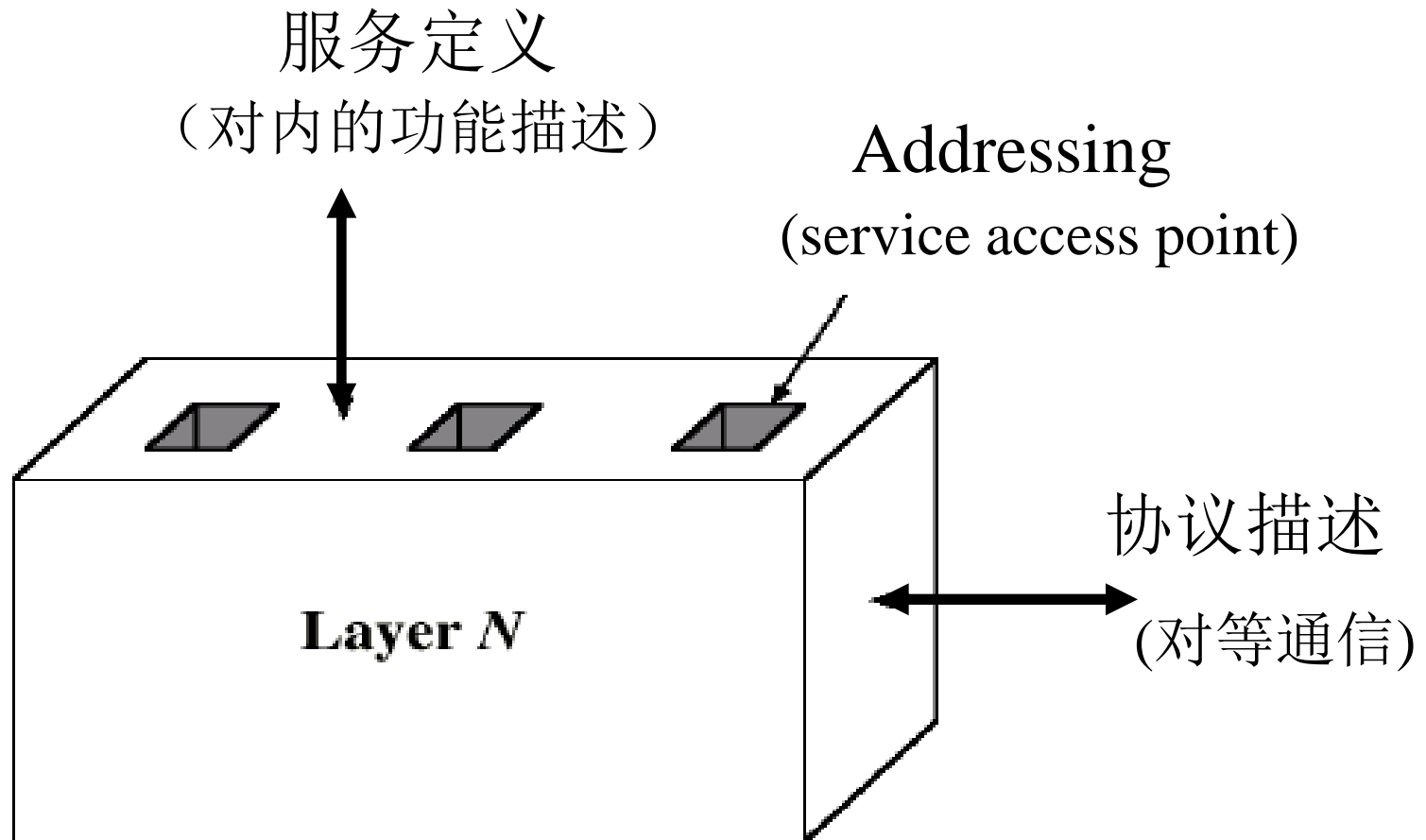
实体、协议、服务和访问点

- **实体(entity)** 表示任何可发送或接收信息的硬件或软件进程。
- **协议** 是控制**两个对等实体**进行通信的规则的组合。
- 在协议的控制下，两个对等实体间的通信使得**本层能够向其上一层提供服务**。
- 要实现**本层**协议，还需要**使用下层所提供的服务**。

实体、接口、服务、服务访问点的关系



层接口、服务、协议



面向连接服务与无连接服务

- 面向连接服务(connection-oriented)
 - 面向连接服务要经历连接建立、数据传输和连接释放这三个阶段。
- 无连接服务(connectionless)
 - 两个实体之间的通信不需要先建立好连接。
 - 是一种不可靠的服务。这种服务常被描述为“尽最大努力交付”(best effort delivery)或“尽力而为”。

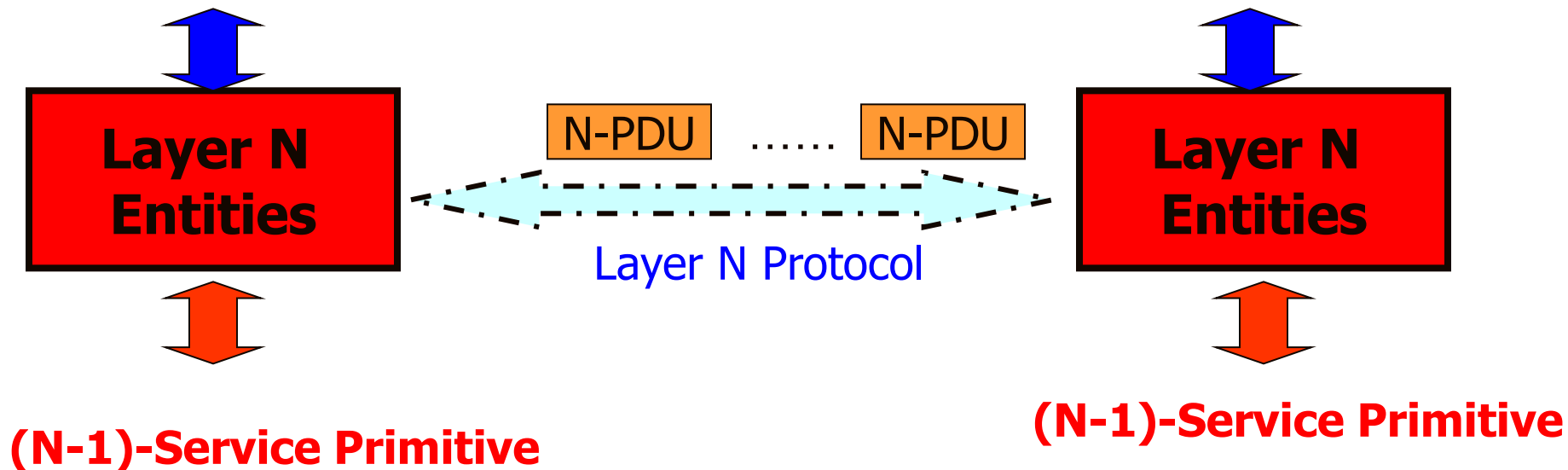
服务与协议的关系

服务与协议是完全不同的两个概念，但二者常常被混淆。

- Service – says **what** a layer does
- Interface – says **how** to **access** the service
- Protocol – says **how** is the service **implemented**

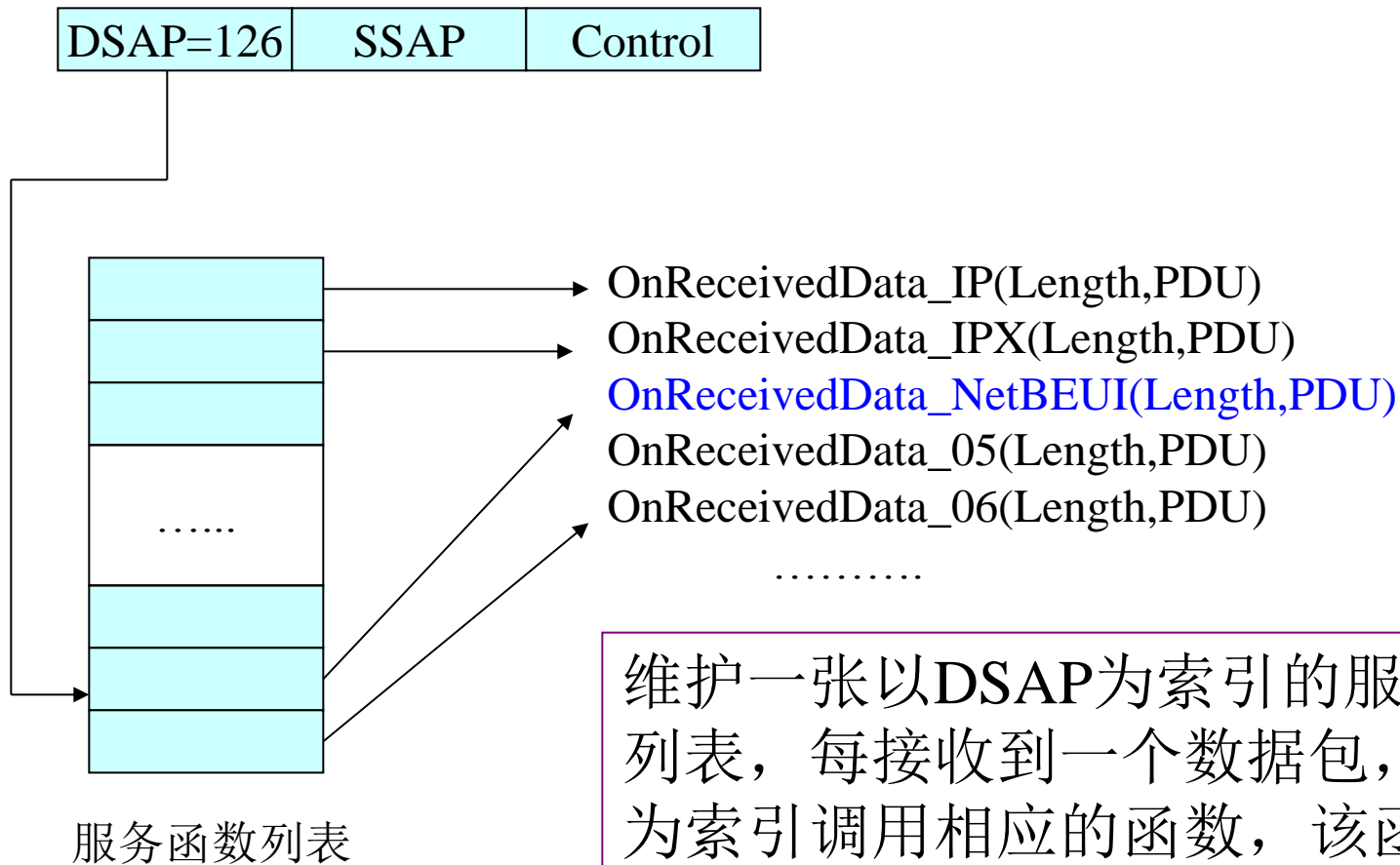
N-Service Primitive

N-Service Primitive



服务访问点(SAP)

服务访问点SAP——服务的入口地址



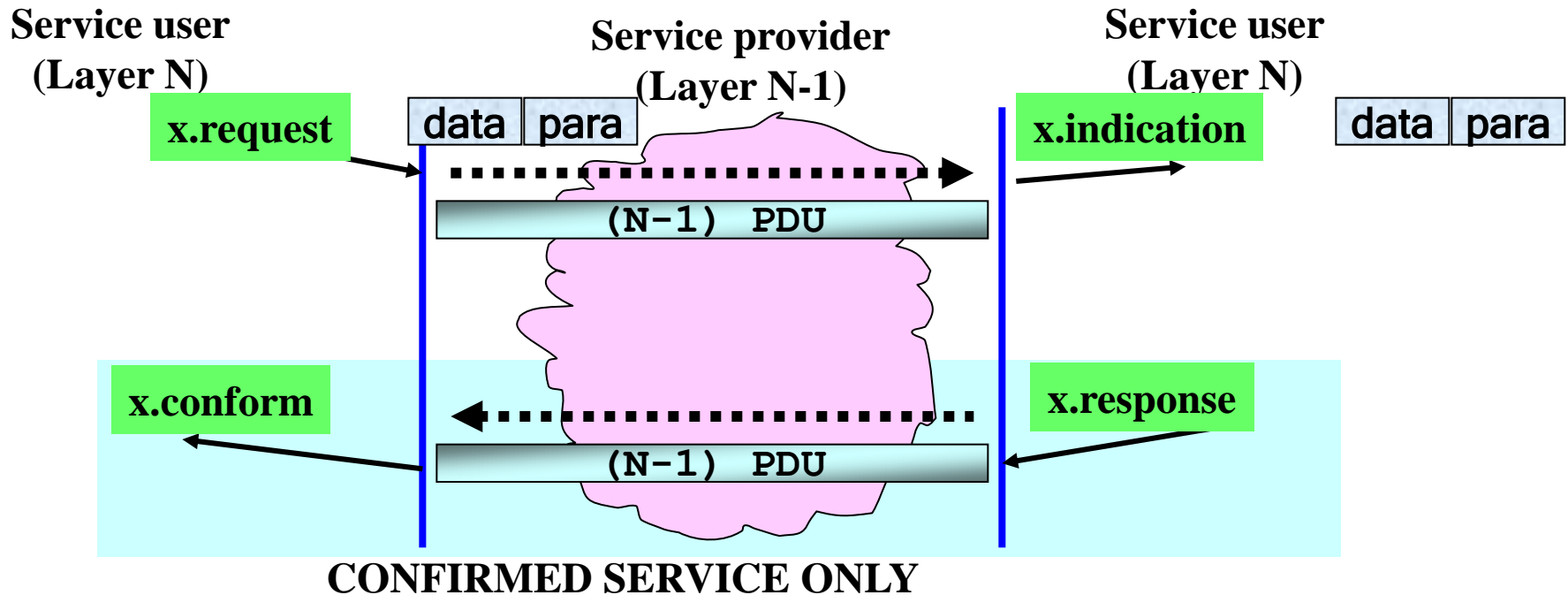
维护一张以DSAP为索引的服务函数列表，每接收到一个数据包，以DSAP为索引调用相应的函数，该函数把数据包挂到相应接收处理队列。

服务原语

- 下层为上层提供的服务用一组原语（Primitive）来描述，称为**服务原语**。
- 服务原语分为4类：**请求、指示、响应、证实**
- 大多数原语带有参数，以明确服务的具体要求，实现实体双方的协商（negotiation）。服务有“**有证实（Confirmed）**”和“**无证实（Unconfirmed）**”之分。
有证实服务用到请求、指示、响应和证实4个原语，而无证实服务只用到请求和指示2个原语。
- 单个服务访问点（SAP – Service Access Point）用“**状态变迁图（State Transition Diagram）**”来描述；一对服务访问点之间的状态变迁关系用“**时序图（Time Sequence Diagram）**”来描述。

OSI服务原语

原语类型	英文对照	含义
请求	request	一个实体希望得到完成某些操作的服务
指示	indication	通知一个实体，有某个事件发生
响应	response	一个实体对某个事件作出响应
证实	confirm	返回对先前请求的响应

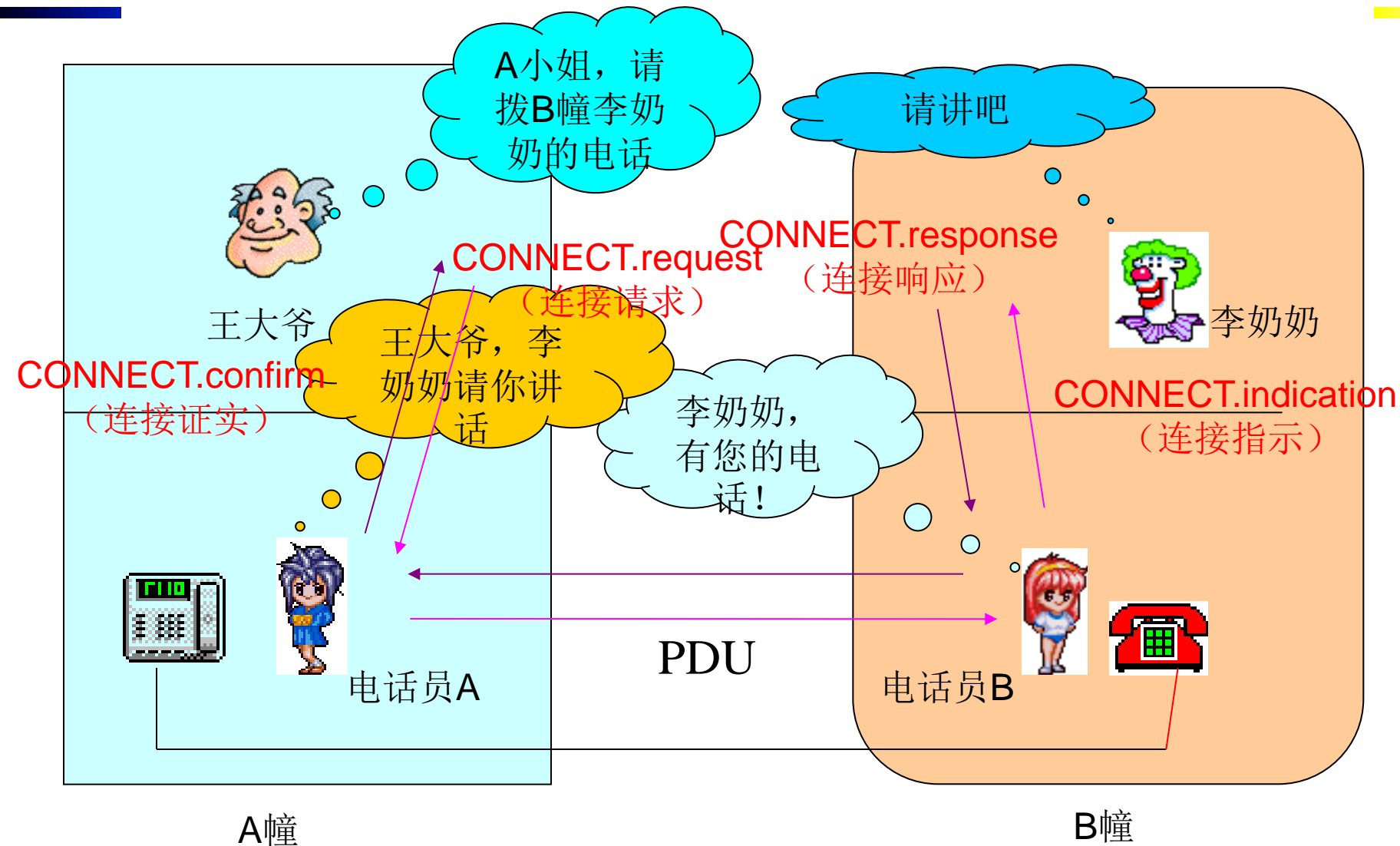


服务原语实例—面向连接通信过程

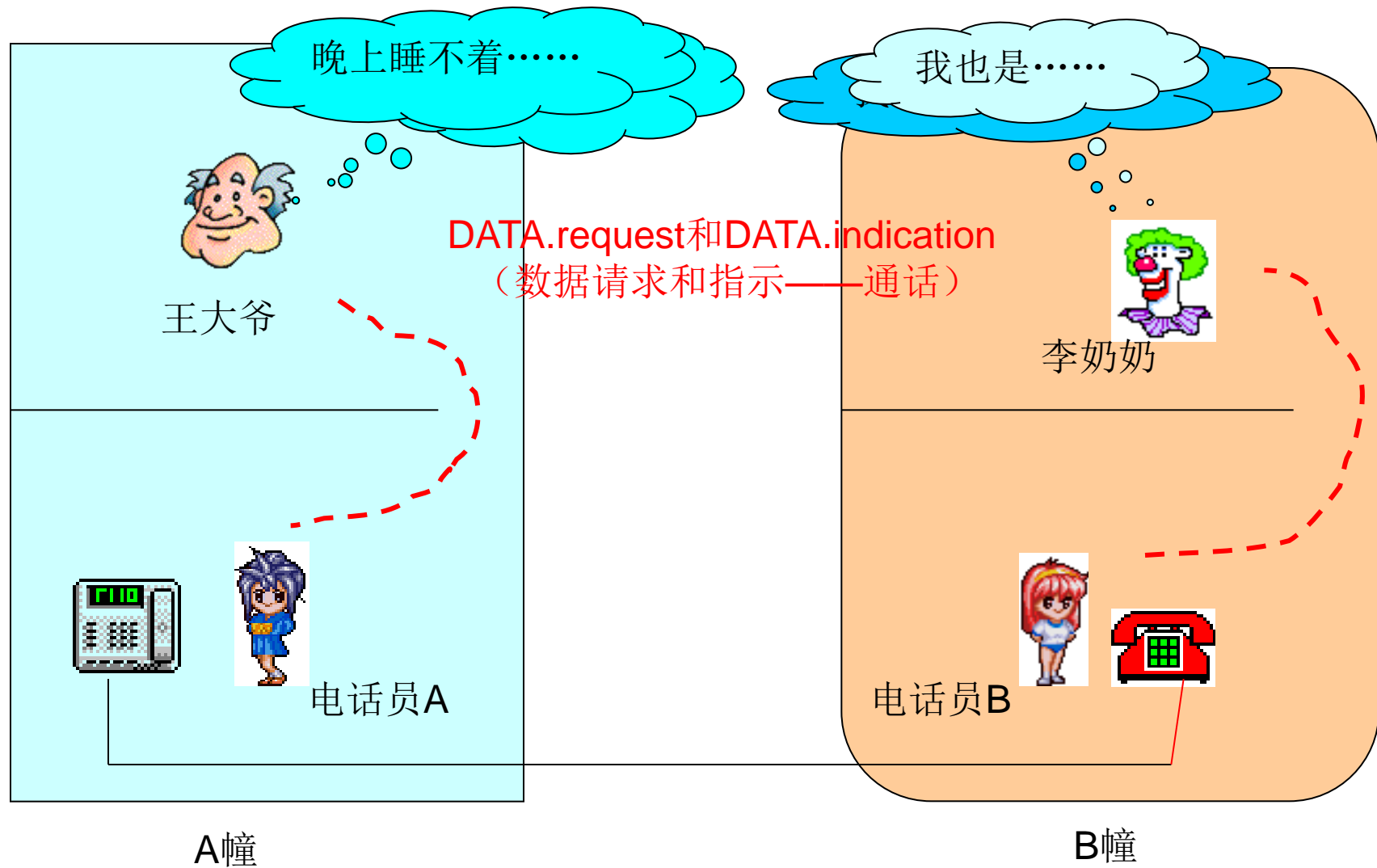
- (1) `CONNECT.request`: 请求建立连接。
- (2) `CONNECT.indication`: 向被呼实体指示连接请求。
- (3) `CONNECT.response`: 被呼方用以表示接受或拒绝连接请求。
- (4) `CONNECT.confirm`: 通知呼叫方建立连接的请求是否被接受。
- (5) `DATA.request`: 请求发送数据。
- (6) `DATA.indication`: 表示数据的到达。
- (7) `DISCONNECT.request`: 请求释放连接。
- (8) `DISCONNECT.indication`: 通知对等实体释放的要求。

在本例中，*CONNECT*是有证实的服务（需要有明确的答复），而*DISCONNECT*是无证实服务（不需要答复）。

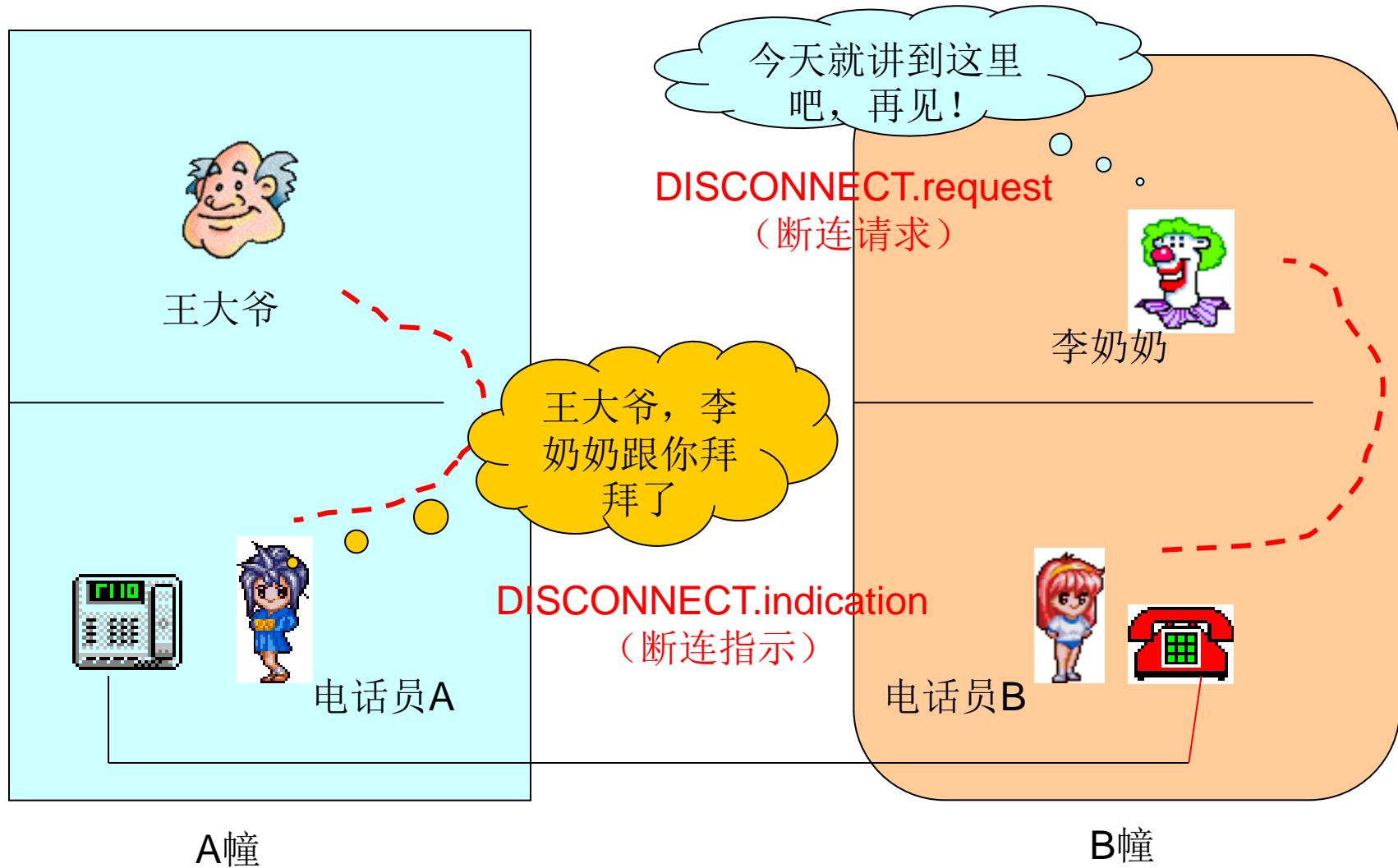
假设，两座楼上有两位孤独的老人（服务用户）要进行电话通信…… 建立连接



假设，两座楼上有两位孤独的老人（服务用户）要进行通信..... 通话



假设，两座楼上有两位行动不便的老人（服务用户）要
进行通信..... 结束通话



对等通信：逻辑通信和物理通信

中国
教师

“你好”

对交谈内容的共识

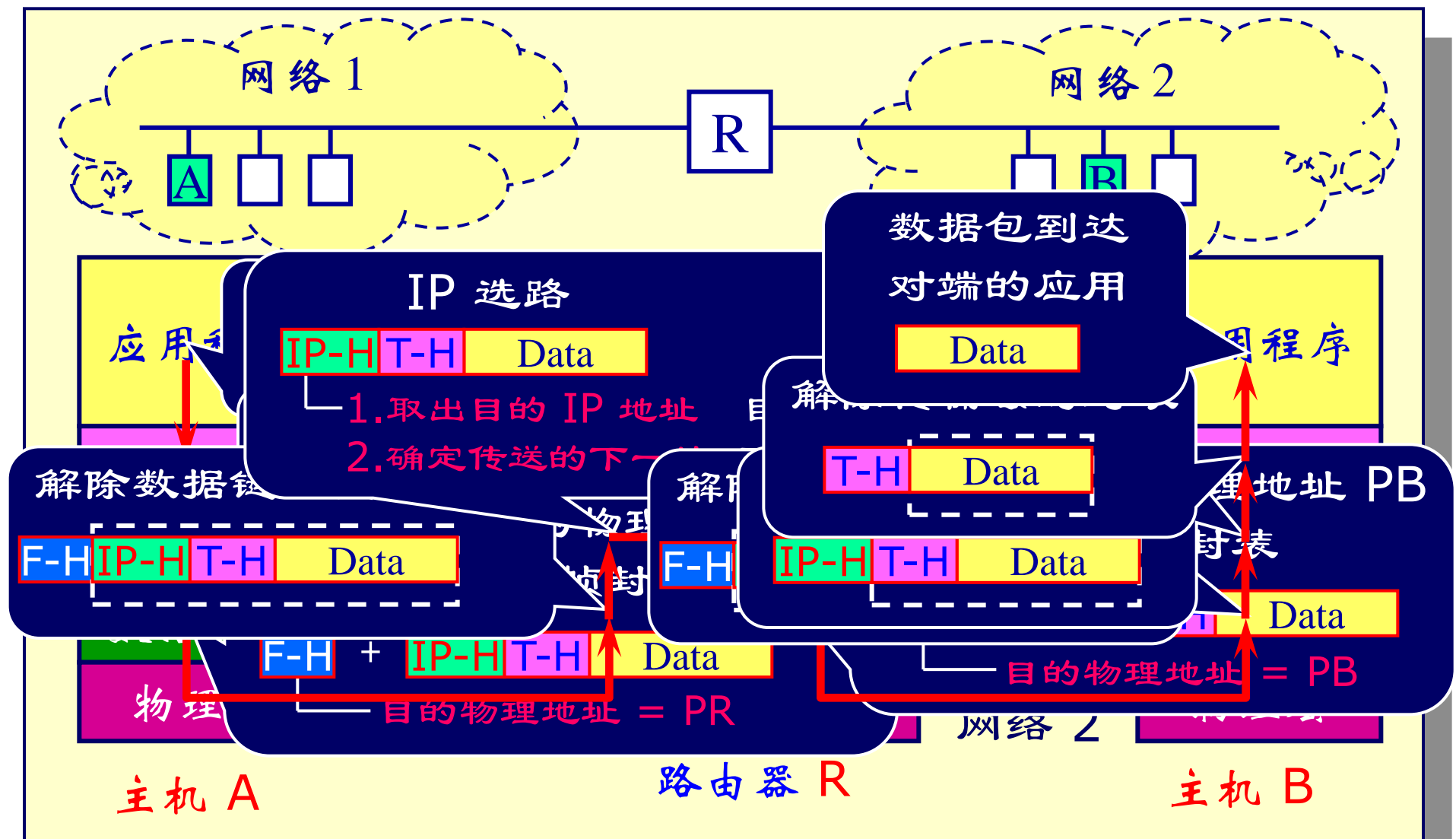
“你好”

德国
教师

实际上，数据不是从一台机器的第 n 层直接传送到另一台机器的第 n 层，而是把数据和控制信息**层层下传，直到最低层**。第一层下面是物理介质，实际数据通信是在它的上面进行的。

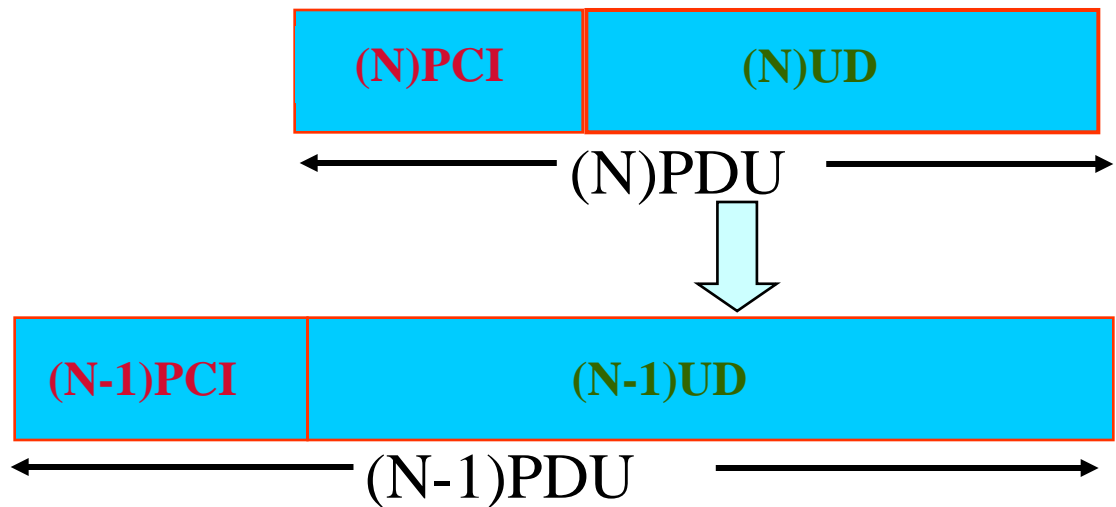
- 翻译、
谁提供什么样的服务？
- 中德教师、翻译各使用谁提供的什么服务？

物理通信



协议数据单元PDU

- 不同站点的对等实体之间所交换的信息，都是按照相应的协议进行的。这些信息传递单元称为PDU (Protocol Data Unit)。N层的PDU表示为(N)PDU。
- 一个PDU由两部分组成：
 - 协议控制信息PCI(Protocol Control Information):报头
 协议头部中含有完成数据传输所需的控制信息：如地址、序号、长度、分段标志、差错控制信息、...
 - 用户数据UD



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，打包的过程称为封装。

封装就是在数据前面加上特定的协议头部。



类比：发送邮件的例子：信装入写有源地址和目的地址的信封中发送，可能还要写明用航空或挂号...

下图是一个现实生活中的数据封装的实例

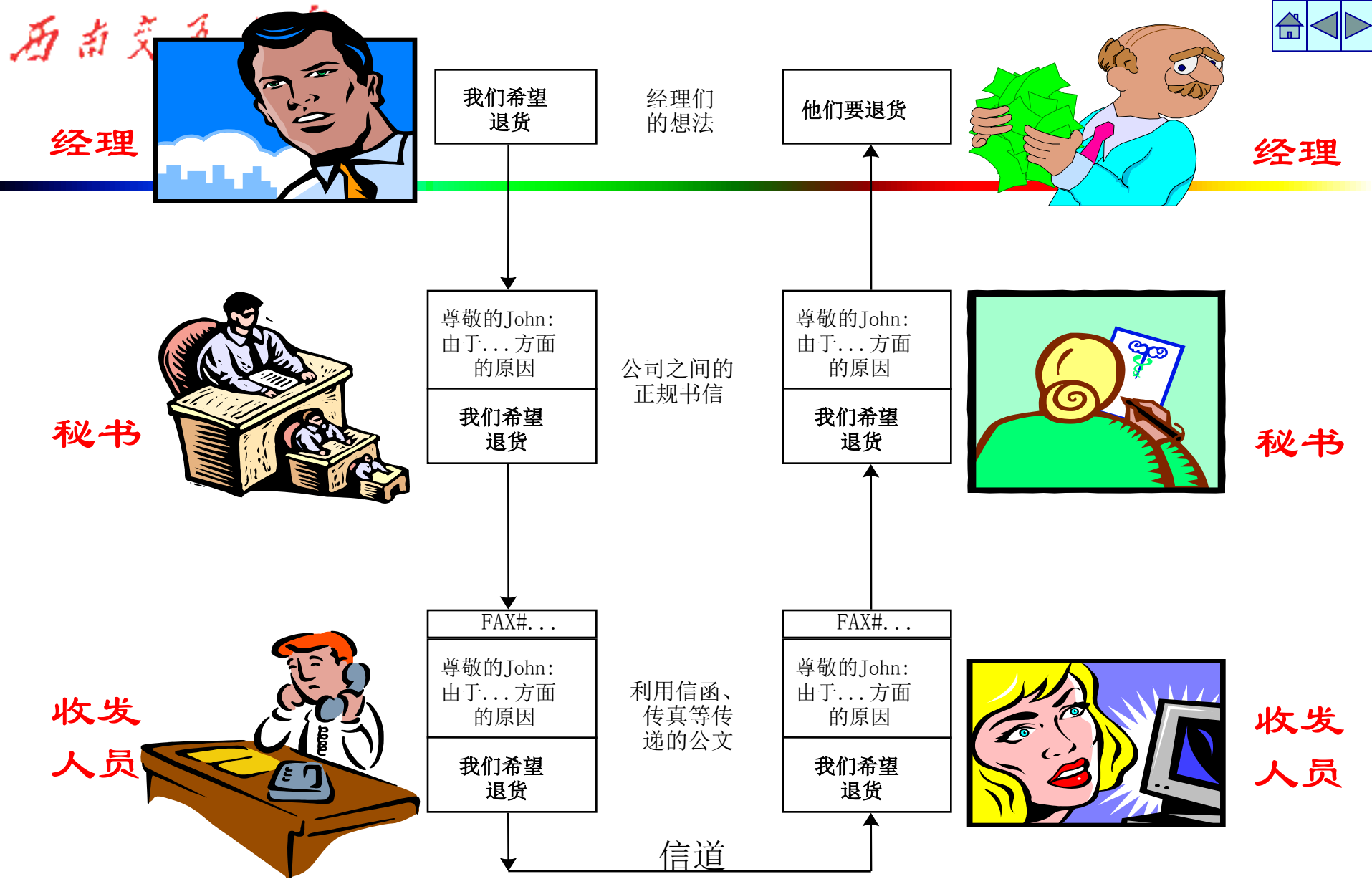
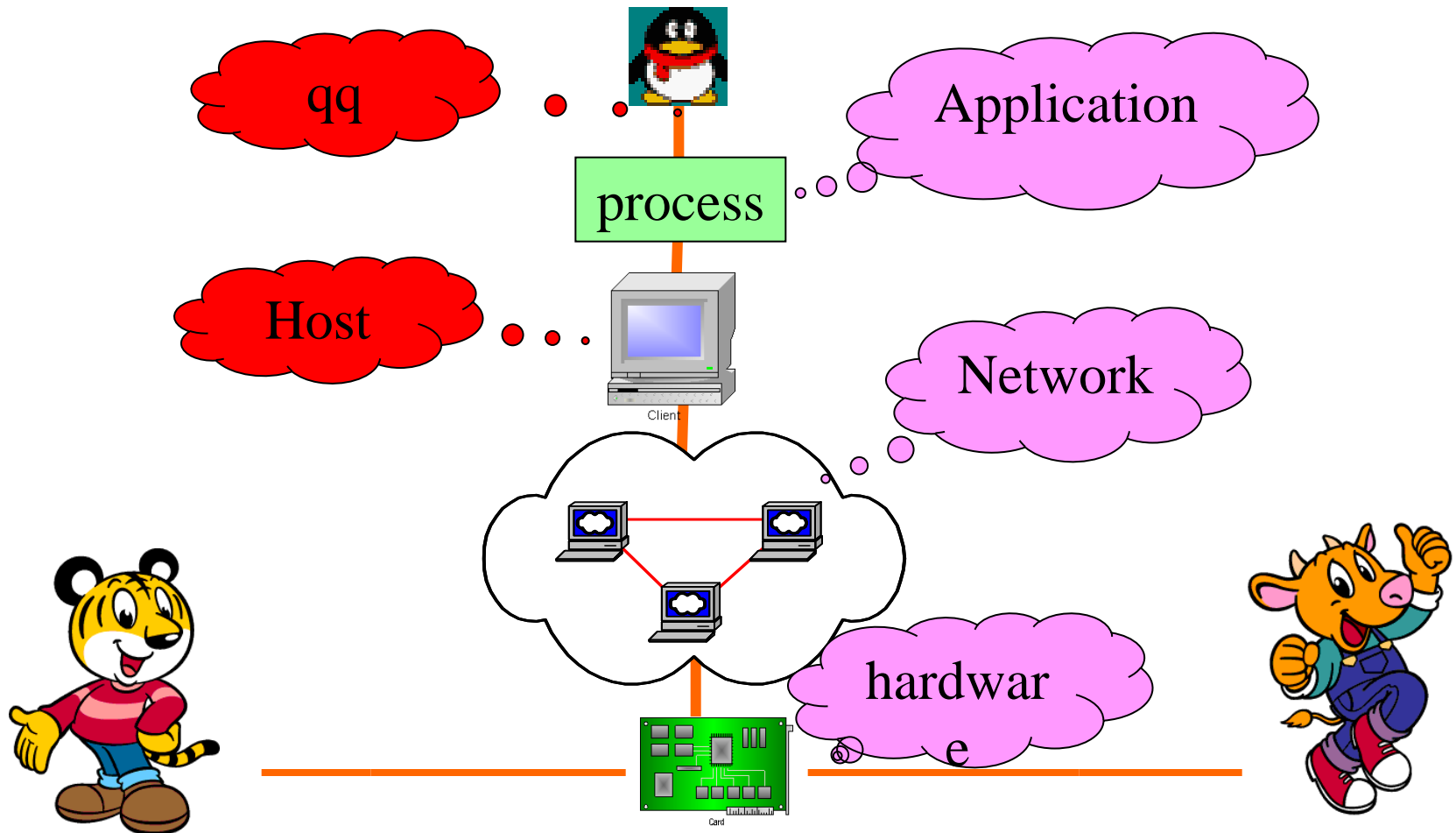


图 协议层次结构与数据封装实例

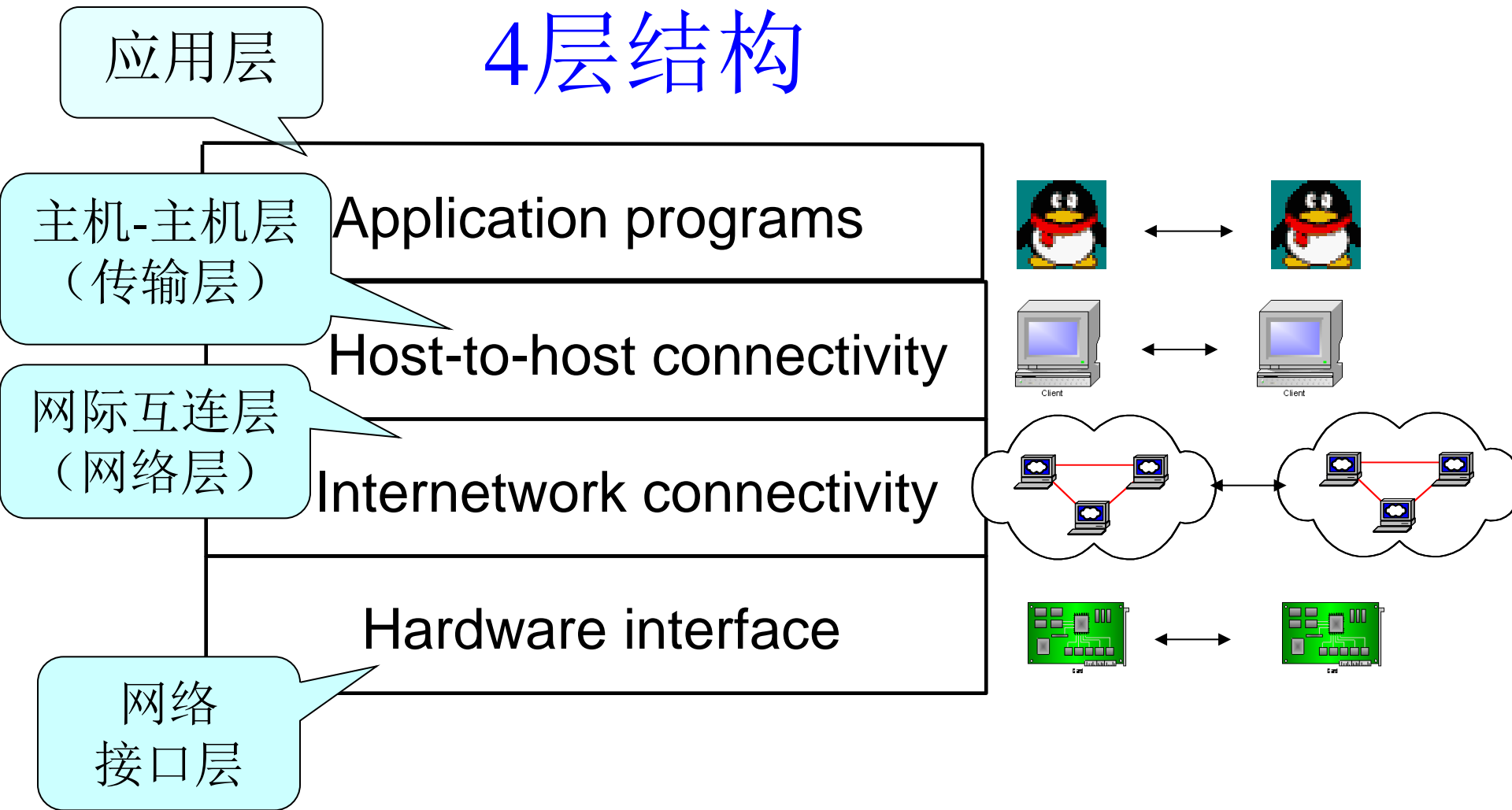
■ Internet 体系结构

Internet体系结构——如何分层？

Example1 ---- Chat with QQ



Example1 ---- Chat with QQ



5. Internet 体系结构

TCP/IP 体系结构与 OSI 体系结构的比较：

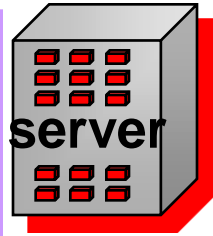
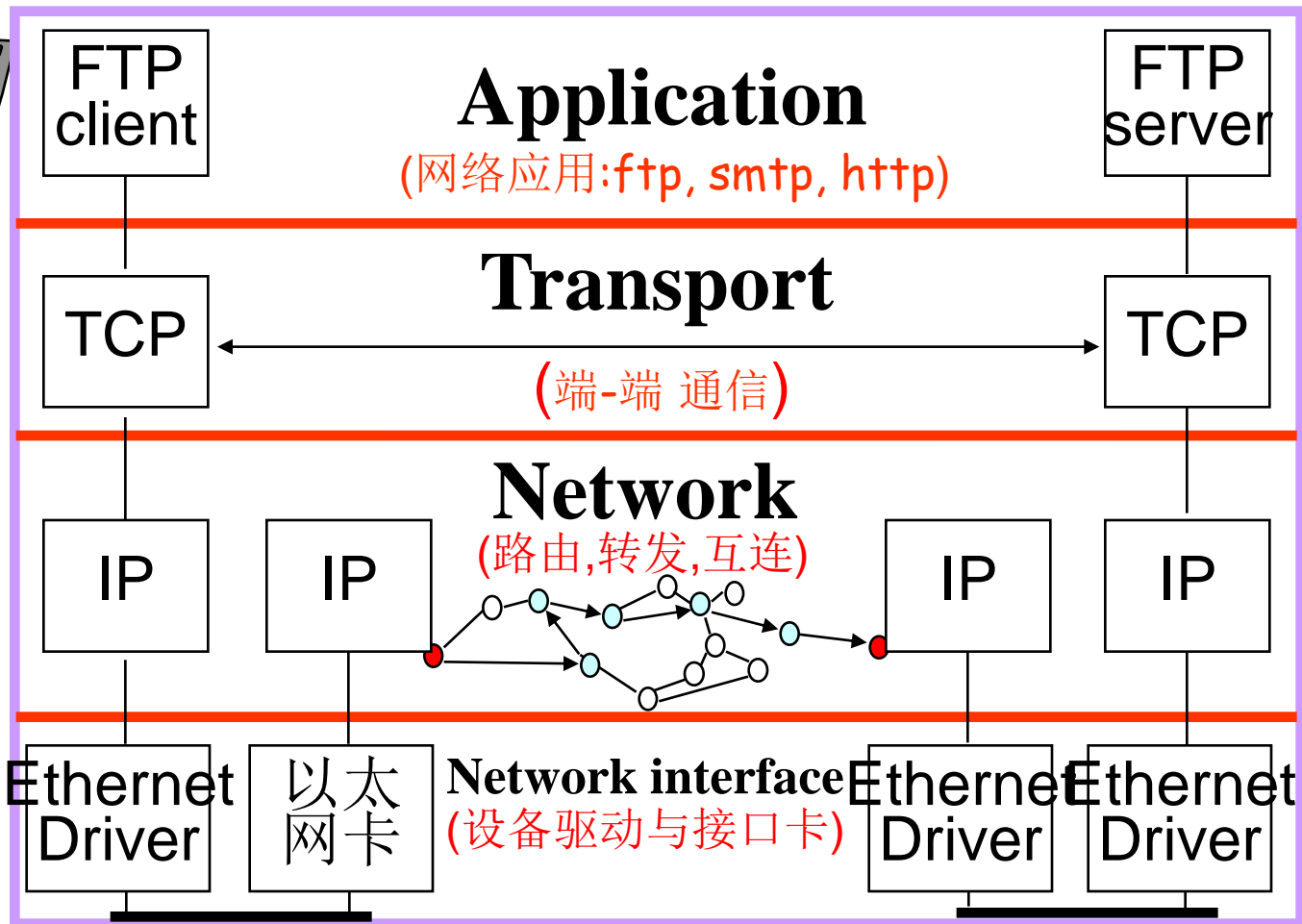
OSI模型

TCP/IP模型

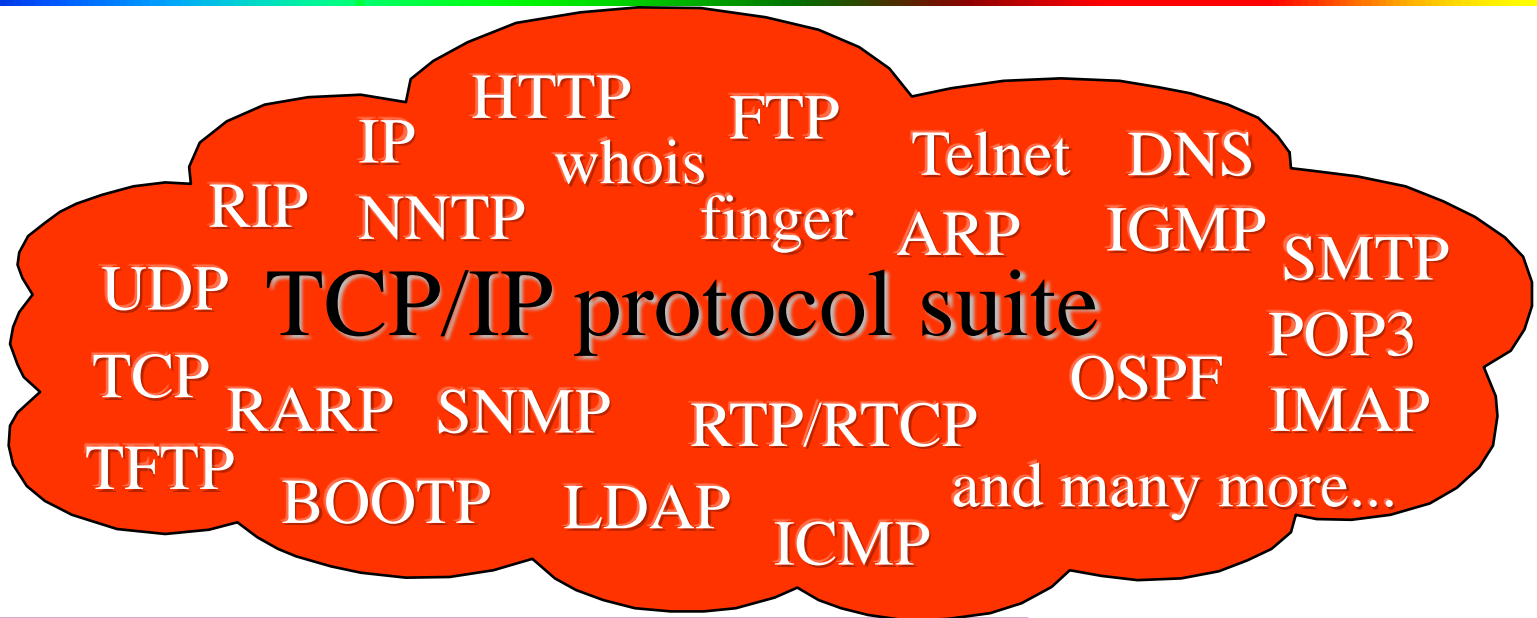
PDU

应用层	应用层	Telnet	FTP	DNS	HTTP	SMTP	Message（报文）
表示层							
会话层							
传输层	传输层	TCP		UDP			Segment（段）
网络层	网际互联层	IP (ICMP、ARP/RARP、IGMP)					Packet（分组）
	网络接口层	CSMA/CD	Token Ring	Token Bus			Frame（帧）
数据链路层							
物理层		Hardware					Bit（比特）

TCP/IP 四层结构



TCP/IP 协议集

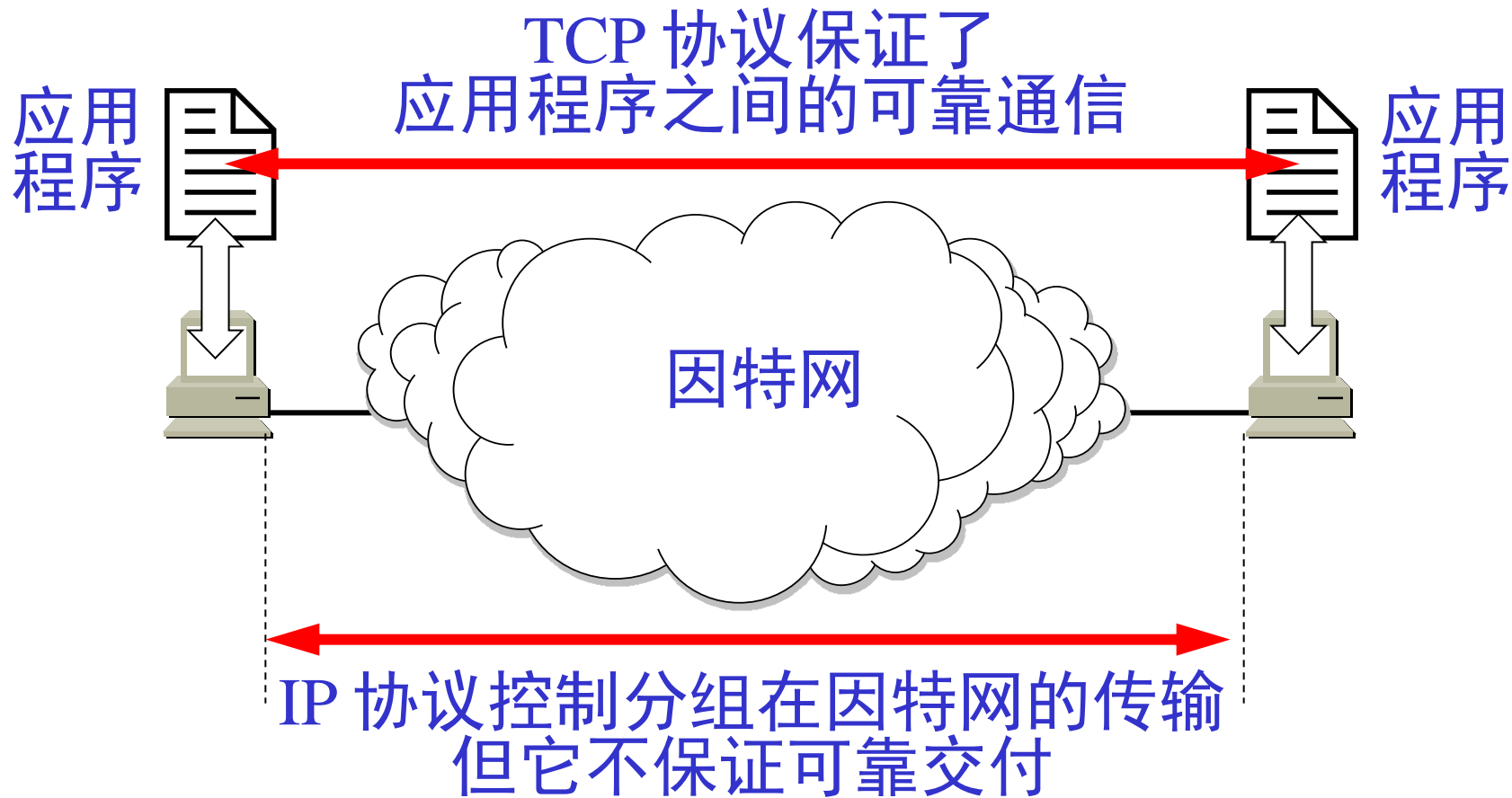


- **TCP/IP** 协议集的主要协议是TCP和IP，除此之外还有许多其他协议。
- 也称为：

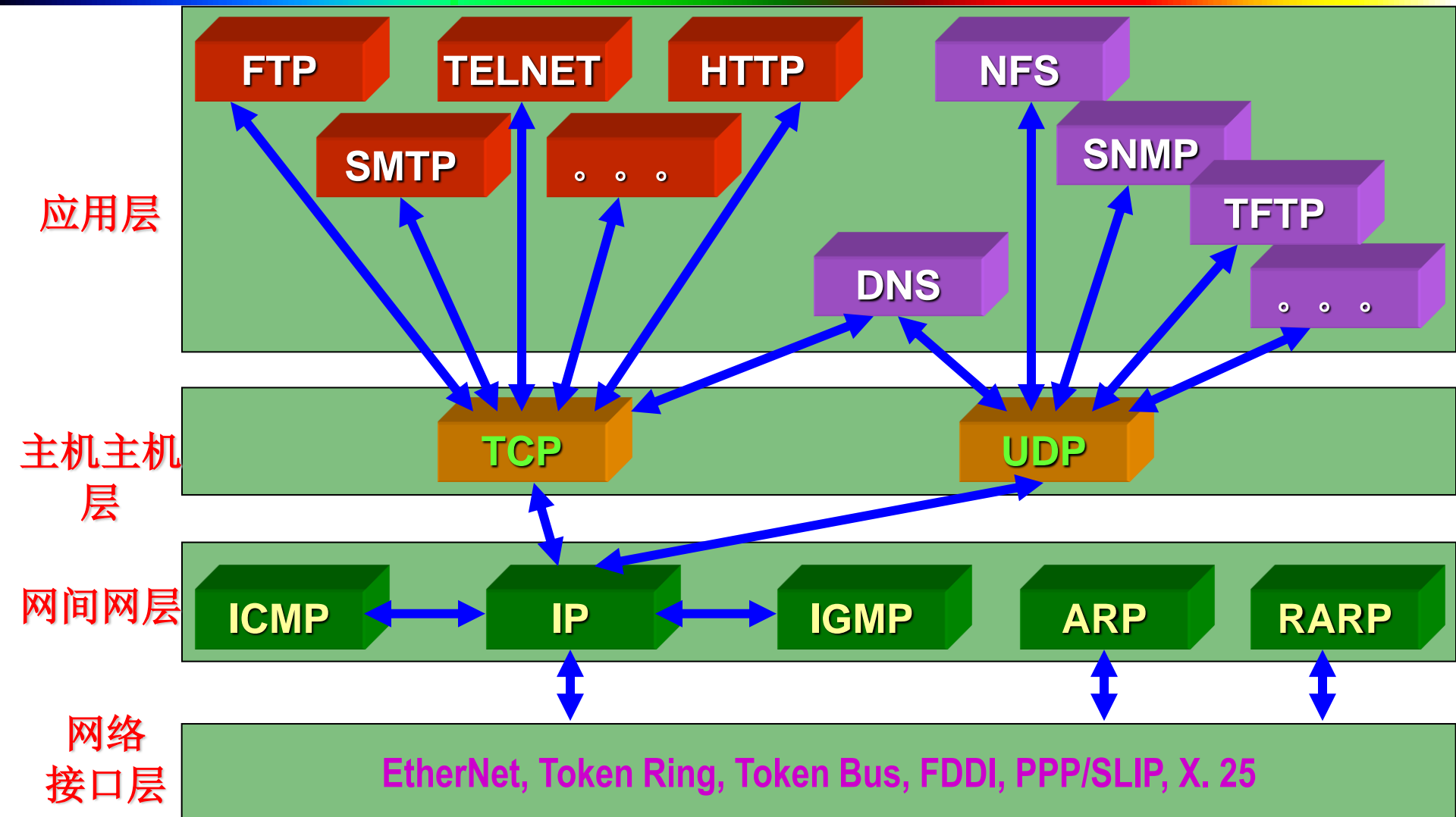
Internet Protocol Suite(Internet协议集)
U.S. Department of Defense (DoD)
Protocol Suite



最重要的两个协议：IP 和 TCP

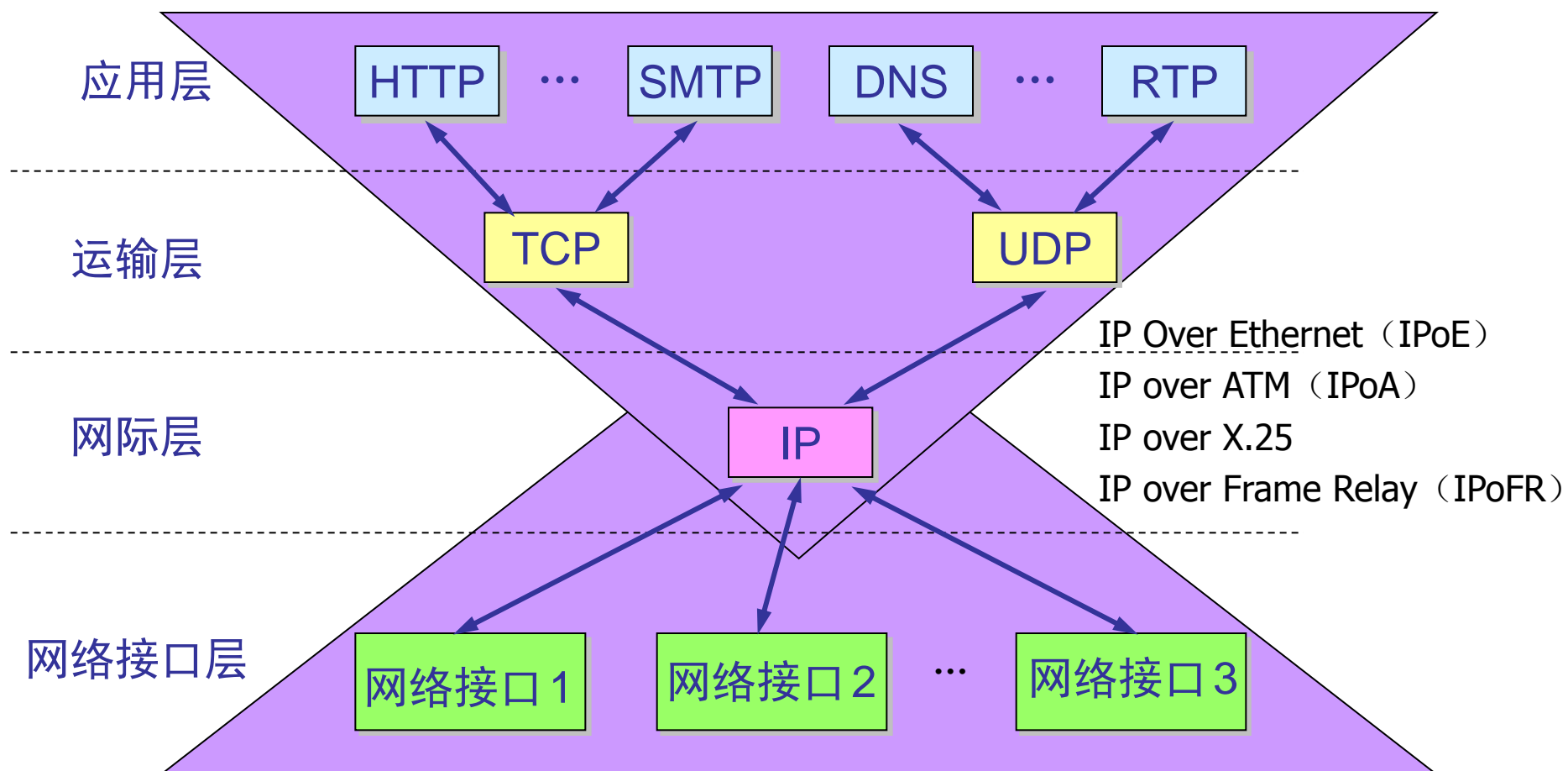


TCP/IP协议族中的应用层协议

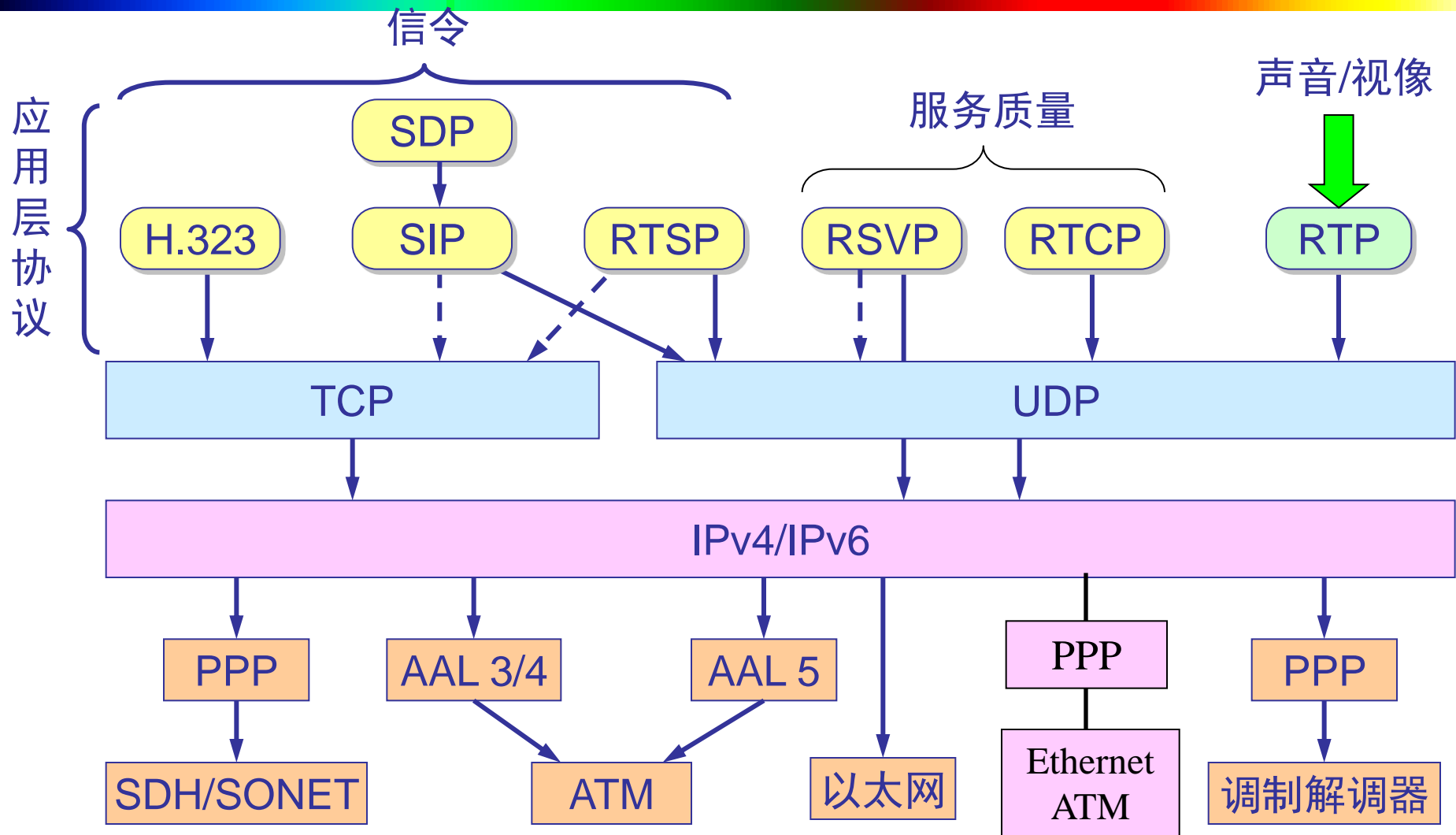


IP over Everything

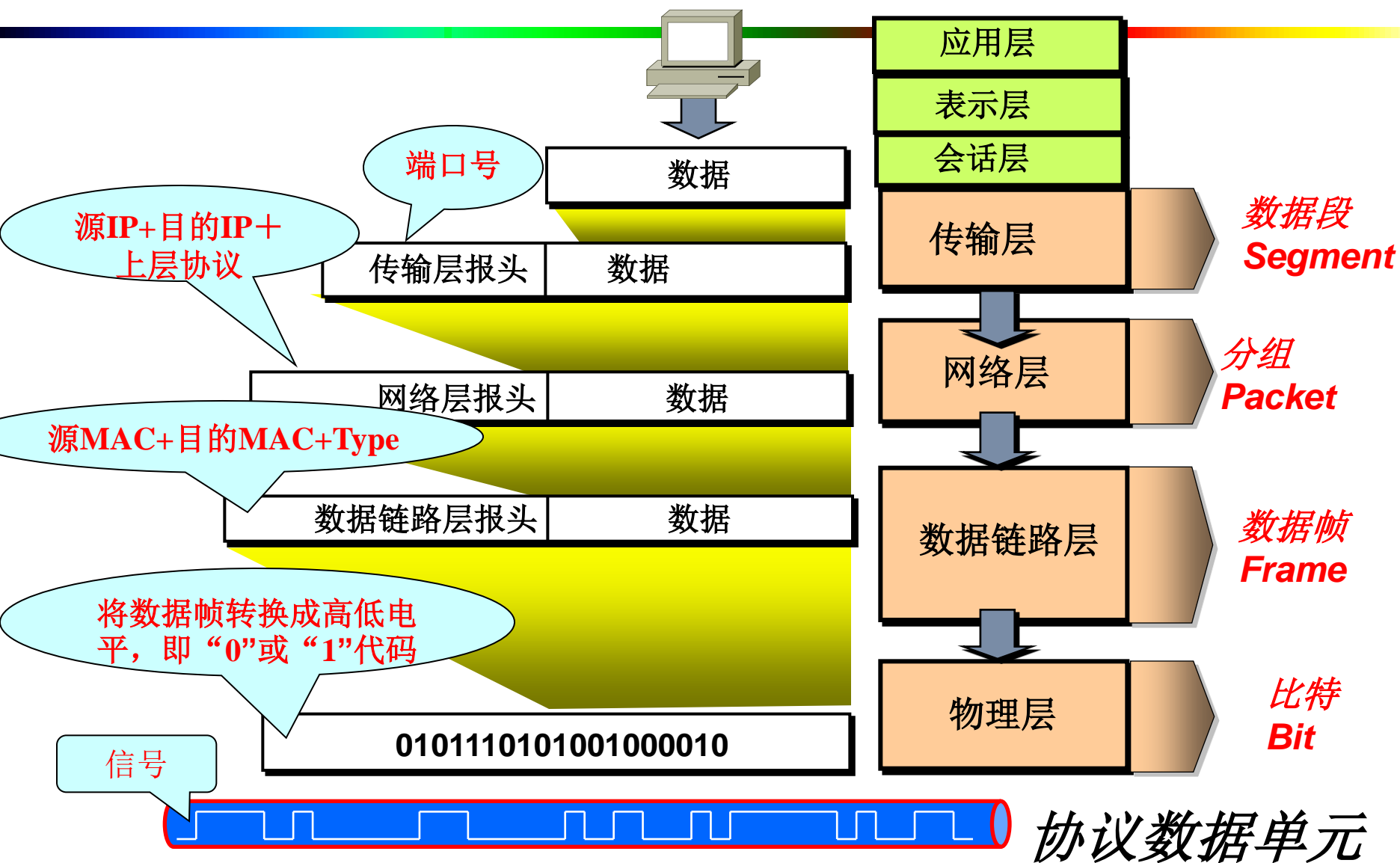
IP可应用到各式各样的网络上



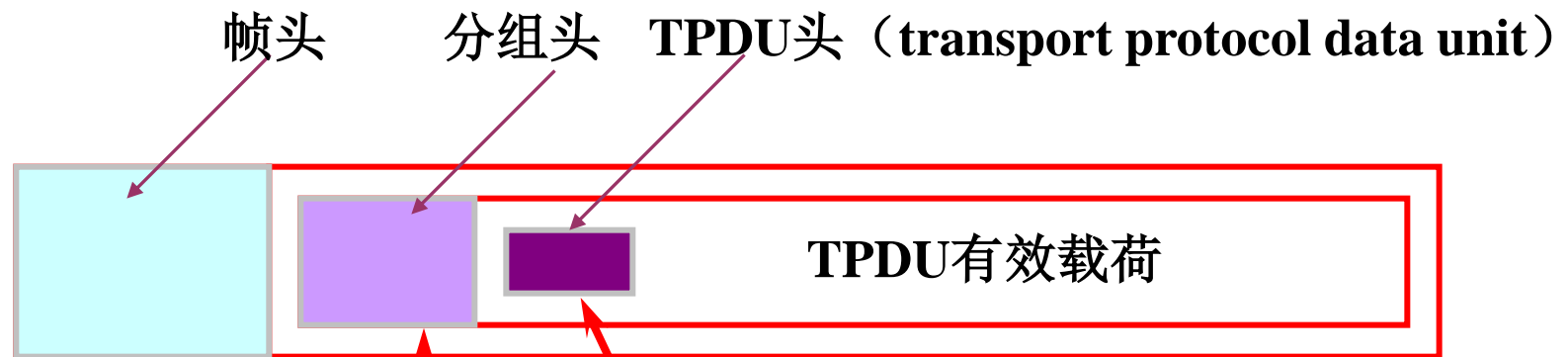
因特网的多媒体体系结构



TCP/IP数据封装--发送数据:数据封装



因特网数据封装—洋葱皮结构

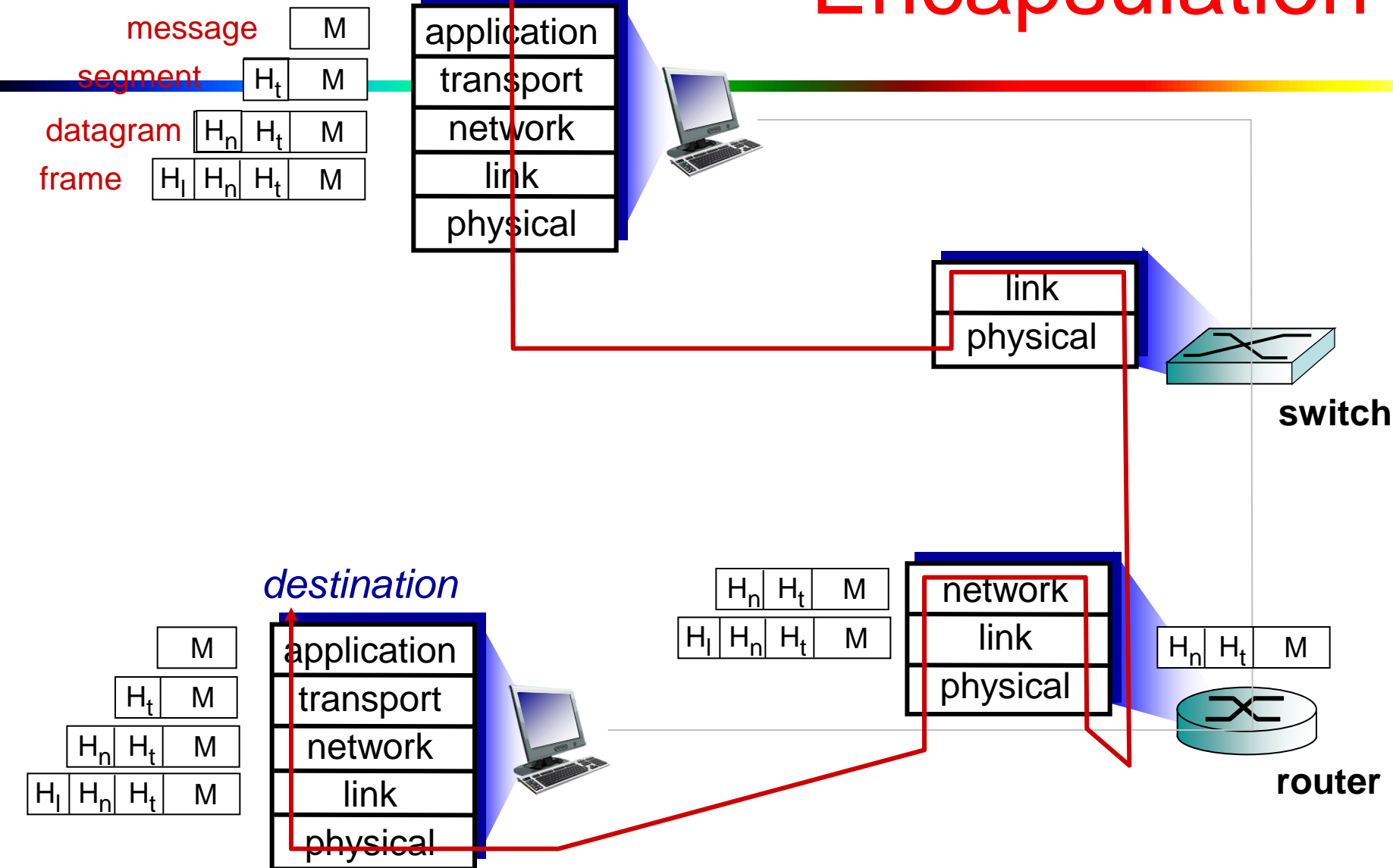


目的 (源) 端口号: 80 (WWW)

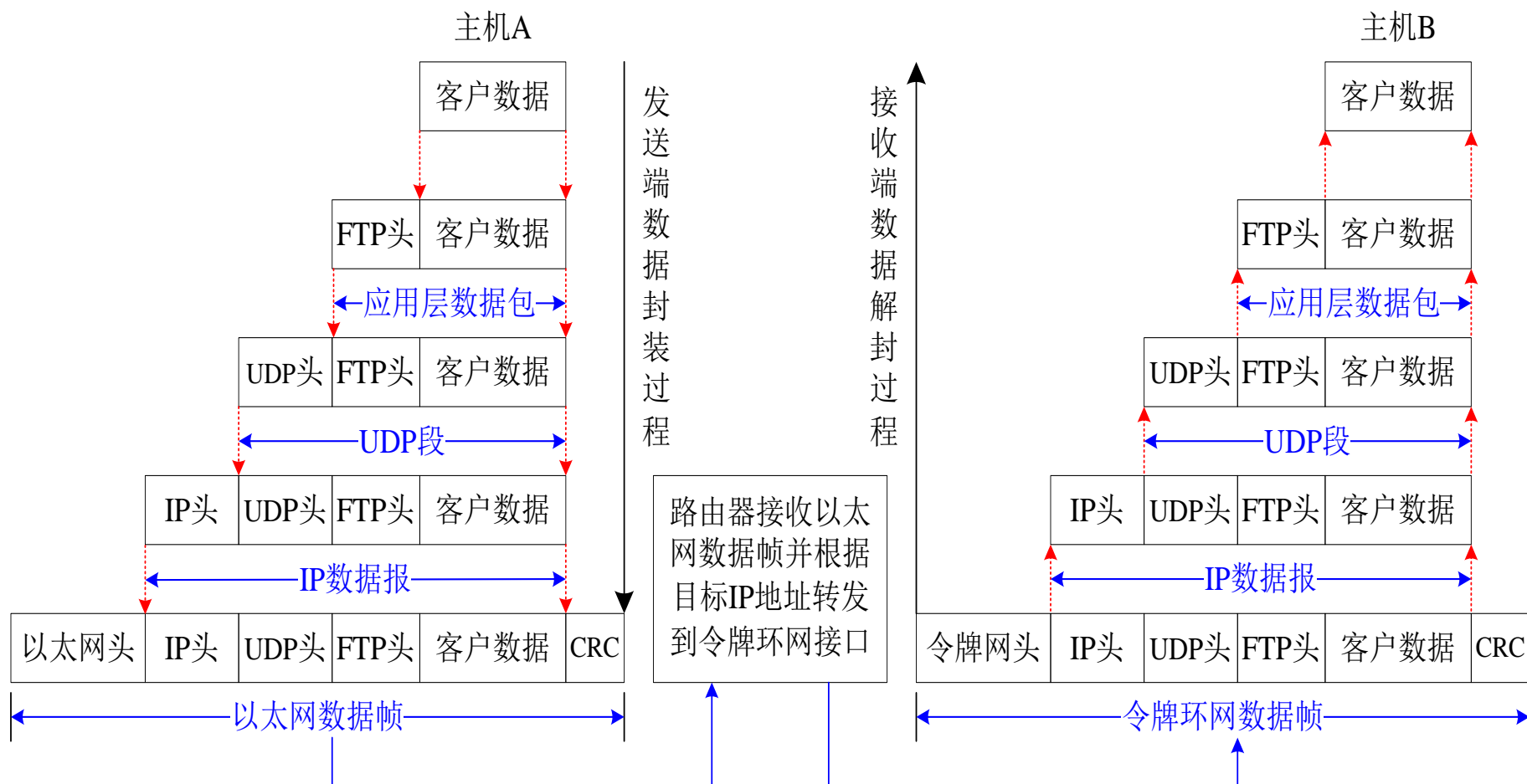
目的 (源) IP地址: 202.197.96.43

Physical Address... : 00-50-BA-63-87-A4

Encapsulation



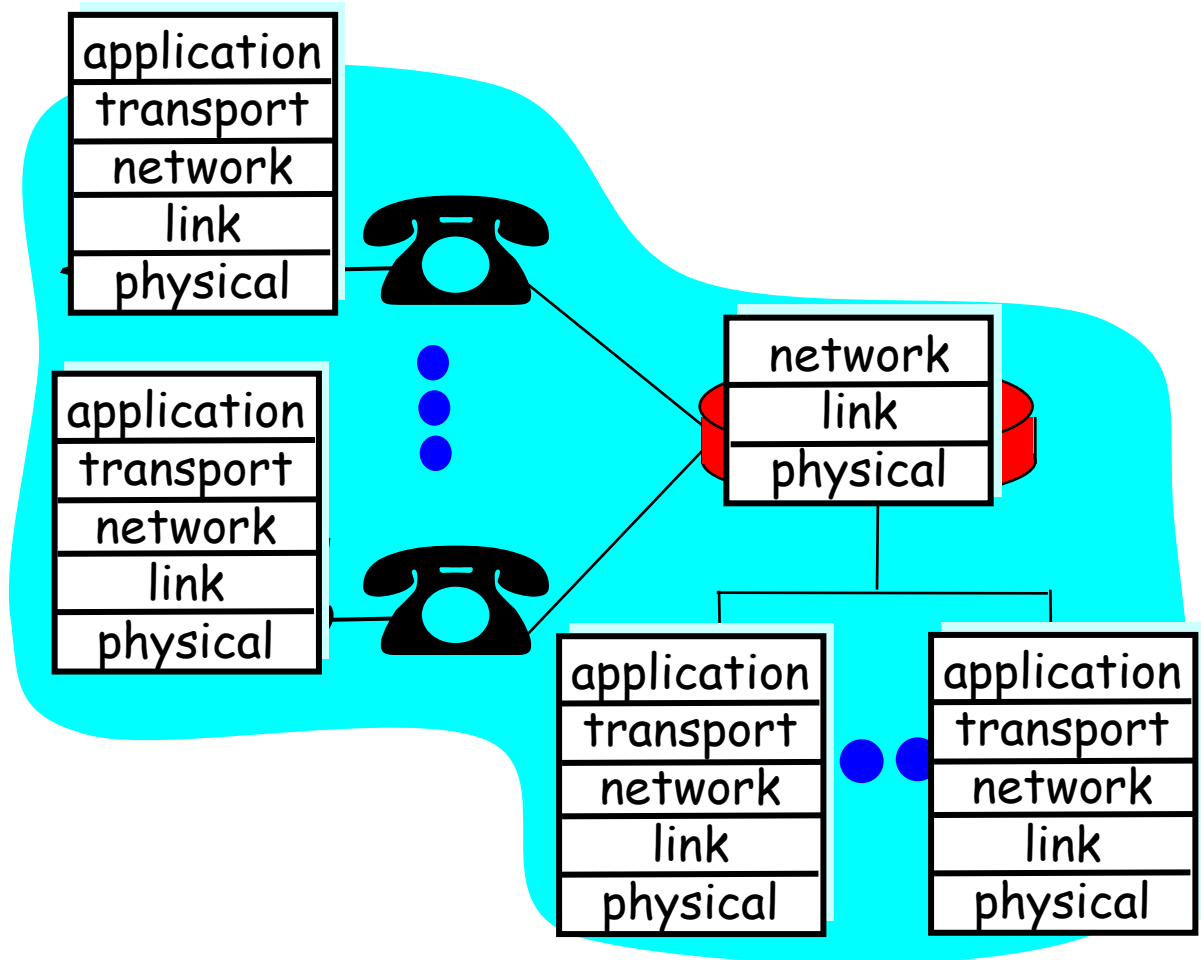
e.g. 多媒体数据的封装与传递过程



逻辑通信

每一层：

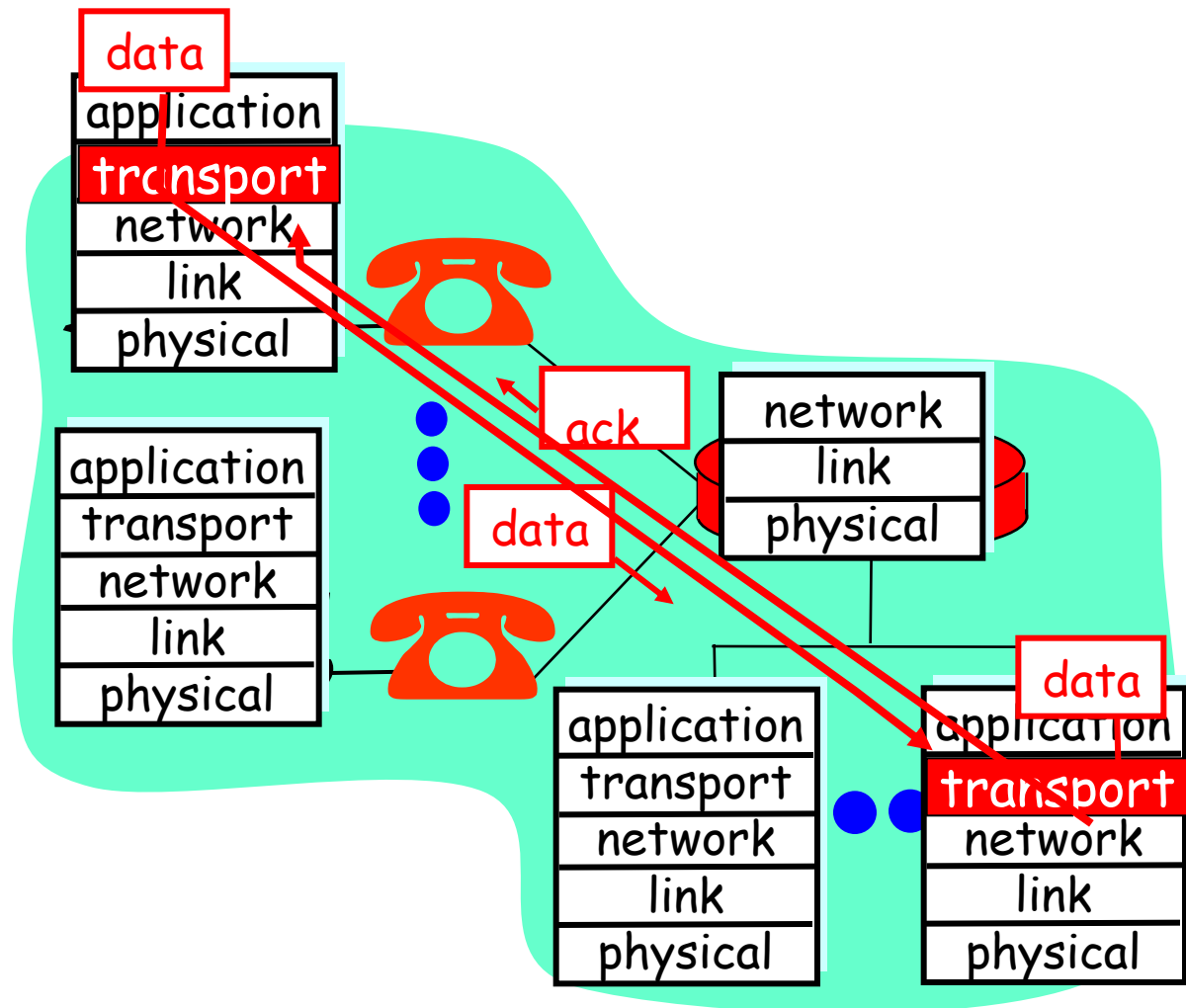
- 分布“实体”
- 在各节点实现各层功能
- 对等实体之间相互通信，交换报文



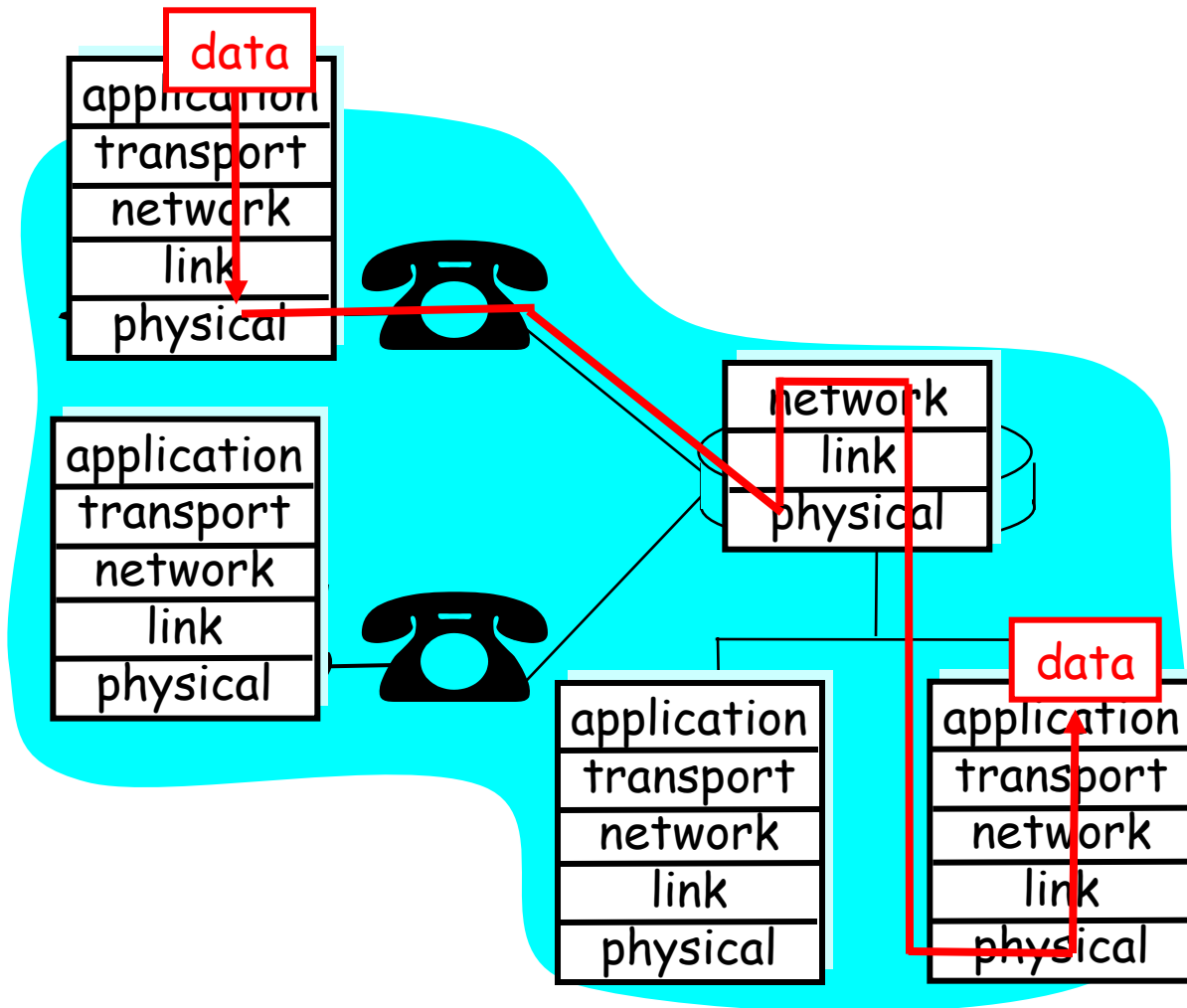
逻辑通信

以传输层对等通信为例:

- 从应用层接受数据
- 加上地址、校验码等信息，构成“数据报”
- 发送数据报到对等层
- 等待接收对等层的应答。



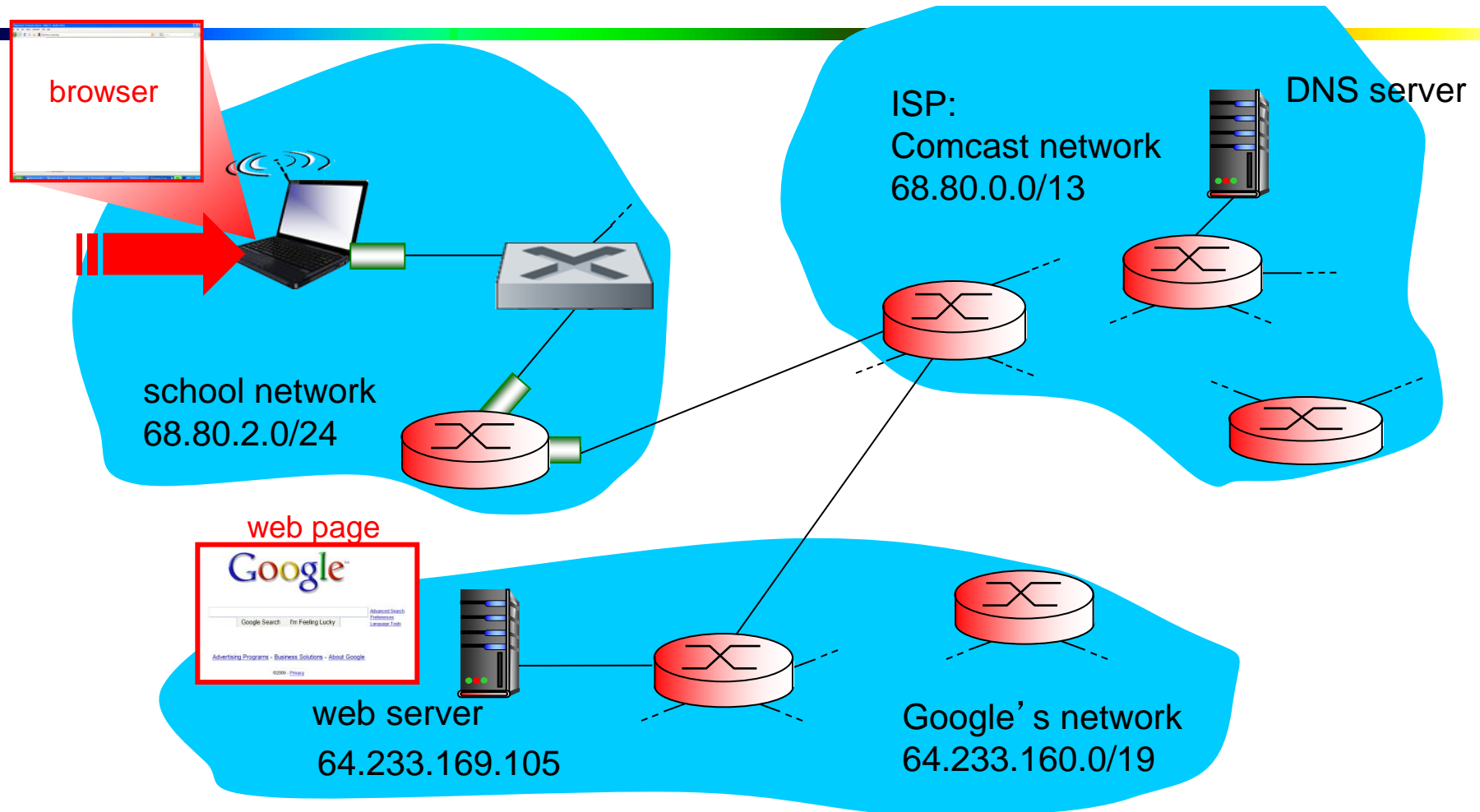
物理通信



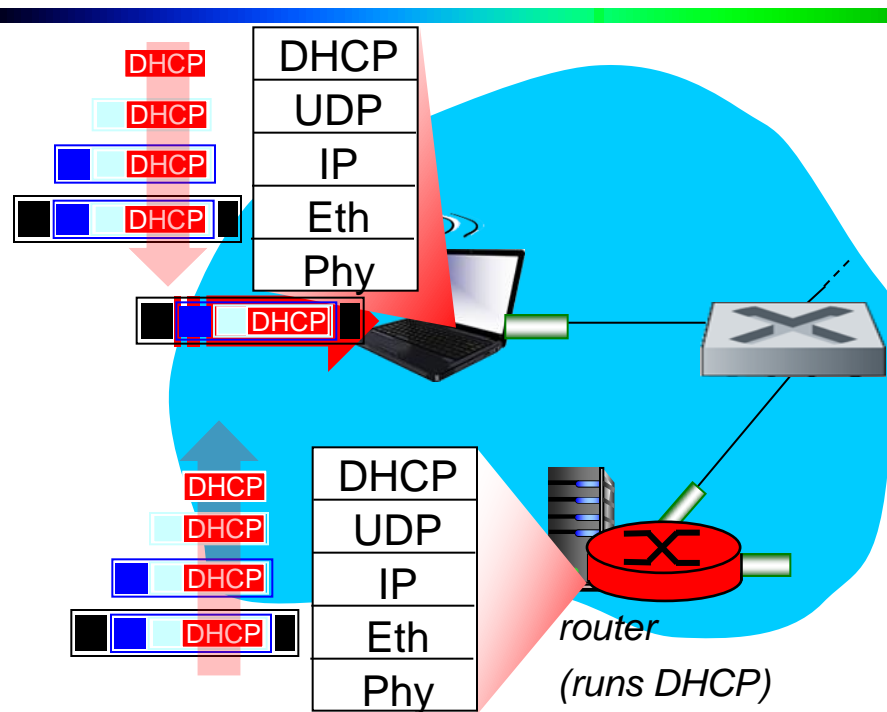
Synthesis: a day in the life of a web request

- journey down protocol stack complete!
 - application, transport, network, link
- putting-it-all-together: synthesis!
 - *goal*: identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
 - *scenario*: student attaches laptop to campus network, requests/receives `www.google.com`

A day in the life: scenario

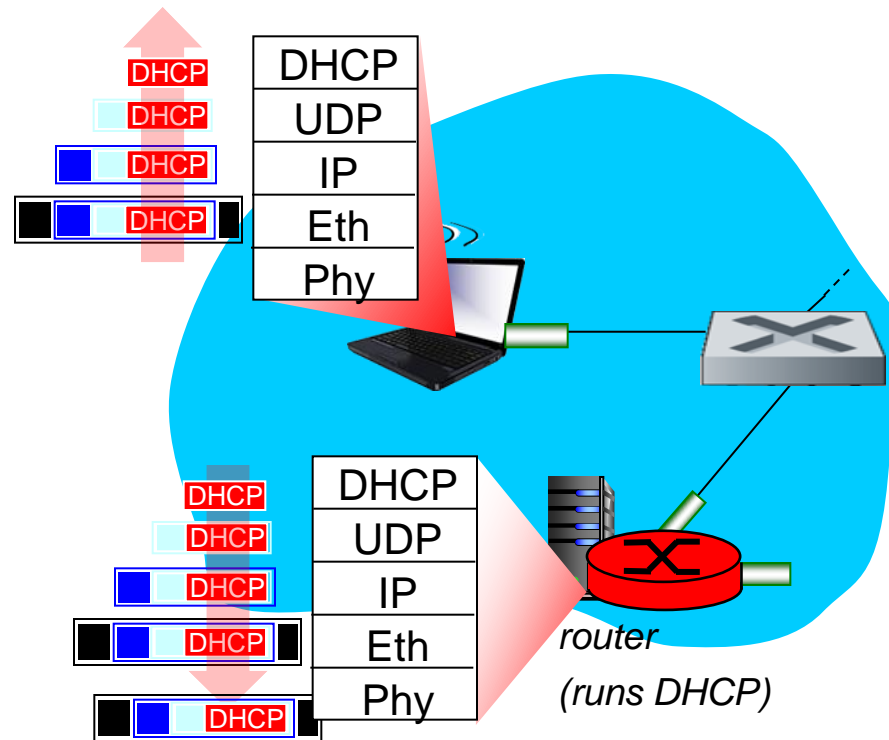


A day in the life... connecting to the Internet



- connecting laptop needs to get its own IP address, addr of first-hop router(default gateway), addr of DNS server: use *DHCP*
- ❖ DHCP request *encapsulated* in *UDP*, encapsulated in *IP*, encapsulated in *802.3* Ethernet
- ❖ Ethernet frame *broadcast* (dest: FFFFFFFFFFFFFFFF) on LAN, received at router running *DHCP* server
- ❖ Ethernet *demuxed* to IP demuxed, UDP demuxed to DHCP

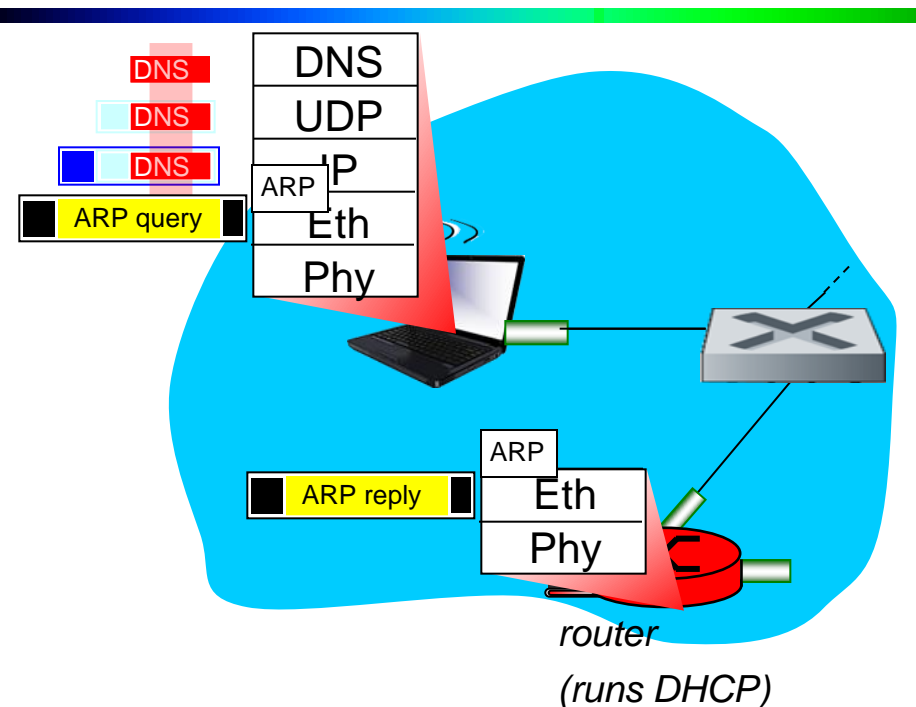
A day in the life... connecting to the Internet



- DHCP server formulates *DHCP ACK* containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- ❖ encapsulation at DHCP server, frame forwarded (*switch learning*) through LAN, demultiplexing at client
- ❖ DHCP client receives DHCP ACK reply

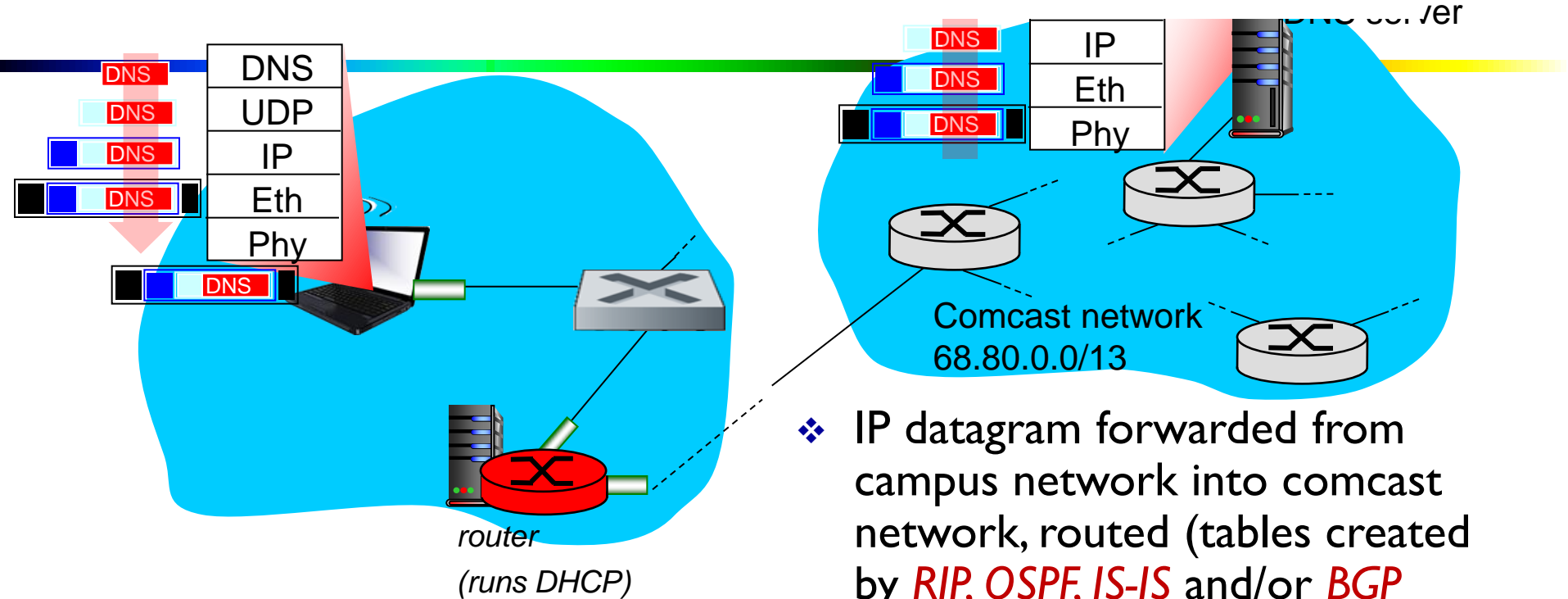
Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router(default gateway)

A day in the life... ARP (before DNS, before HTTP)



- before sending *HTTP* request, need IP address of `www.google.com`: *DNS*
- ❖ DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. To send frame to router, need MAC address of router interface: *ARP*
- ❖ *ARP query* broadcast, received by router, which replies with *ARP reply* giving MAC address of router interface
- ❖ client now knows MAC address of first hop router, so can now send frame containing DNS query

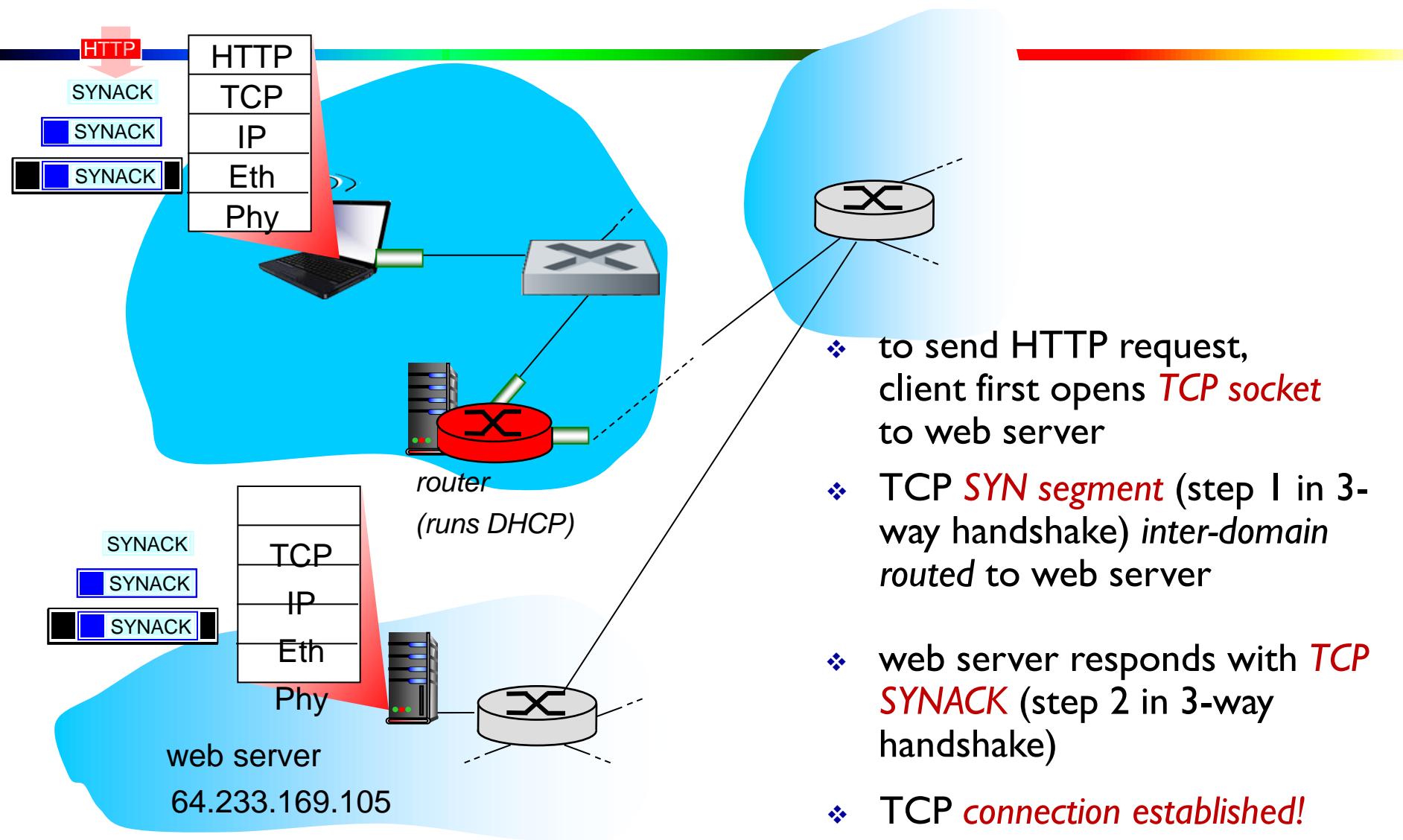
A day in the life... using DNS



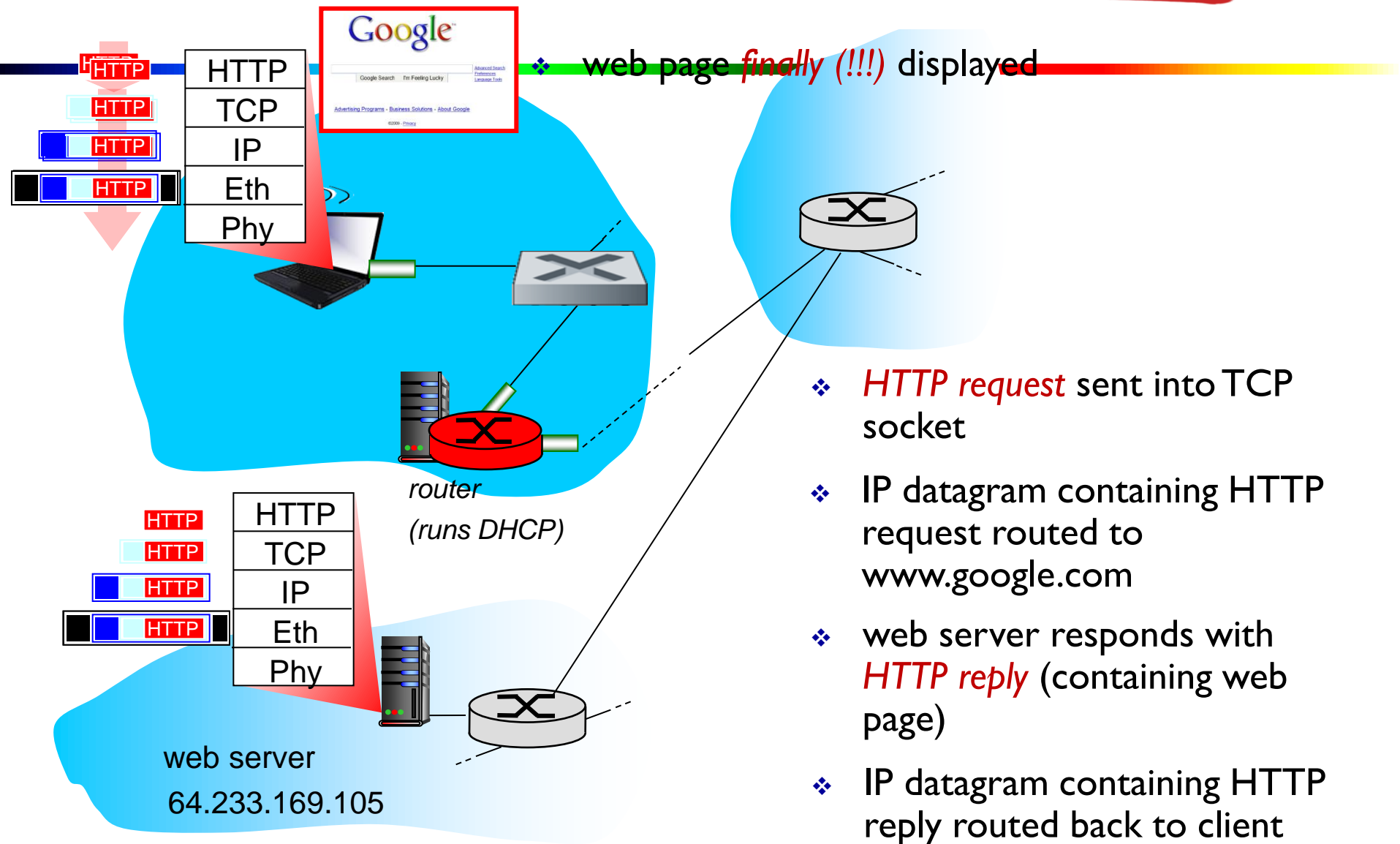
- ❖ IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router

- ❖ IP datagram forwarded from campus network into comcast network, routed (tables created by *RIP*, *OSPF*, *IS-IS* and/or *BGP* routing protocols) to DNS server
- ❖ demux'ed to DNS server
- ❖ DNS server replies to client with IP address of www.google.com

A day in the life...TCP connection carrying HTTP



A day in the life... HTTP request/reply



OSI与TCP/IP的比较

- OSI模型有3个主要概念：**服务、接口、协议**；TCP/IP参考模型最初没有明确区分服务、接口和协议。因此，OSI模型中的协议比TCP/IP参考模型的协议具有更好的隐藏性。
- OSI模型先有模型而无任何参照协议，因此一些功能最初不知道应该放入到哪一层；但TCP/IP模型先有协议后有模型，协议与模型匹配较好。
- **OSI模型有7层**，而**TCP/IP模型只有4层**。
- OSI模型在**网络层**支持无连接和面向连接的通信，但在传输层仅有面向连接的通信，TCP/IP模型在**网络层**网络层只有无连接服务，但传输层支持无连接和面向连接的通信。

OSI模型的优缺点

OSI模型的优点： 结构严谨、科学、完备，是计算机网络的**法律上的(*de jure*)**国际标准

OSI模型的缺点： 在市场化方面 OSI 却失败了

- 糟糕的提出时机
 - 在TCP/IP协议出现商机之后提出，无发展空间。
 - OSI 制定周期太长，使得按 OSI 标准生产的设备无法及时进入市场；
- 糟糕的技术
 - 表示层和会话层没有实现，数据链路层和网络层出现新的子层；
 - 效率低下，低层过分强调可靠性，忽略了高效的无连接服务；
 - 一些功能特性无法确定层次归属，有些功能在多个层次中重复出现；
 - 概念模型由通信专家制定不适合计算机和软件工作方式。
- 糟糕的实现：OSI 的协议实现起来过分复杂，且运行效率很低；
- 糟糕的策略：OSI 的专家们在完成 OSI 标准时没有商业驱动力；

两种国际标准

- 法律上的(*de jure*)国际标准 OSI 并没有得到市场的认可。
- 而非国际标准的TCP/IP 却获得了最广泛的应用。
 - TCP/IP 常被称为事实上的(*de facto*) 国际标准。

TCP/IP的优缺点

TCP/IP参考模型的优点：从体系结构上看，Internet的4层结构比OSI/RM的7层结构简单，也没有OSI/RM中复杂的“服务”等定义，制订的时机合适，在实践中它明显地占了上风——**既成事实的网络标准**(*de facto*)

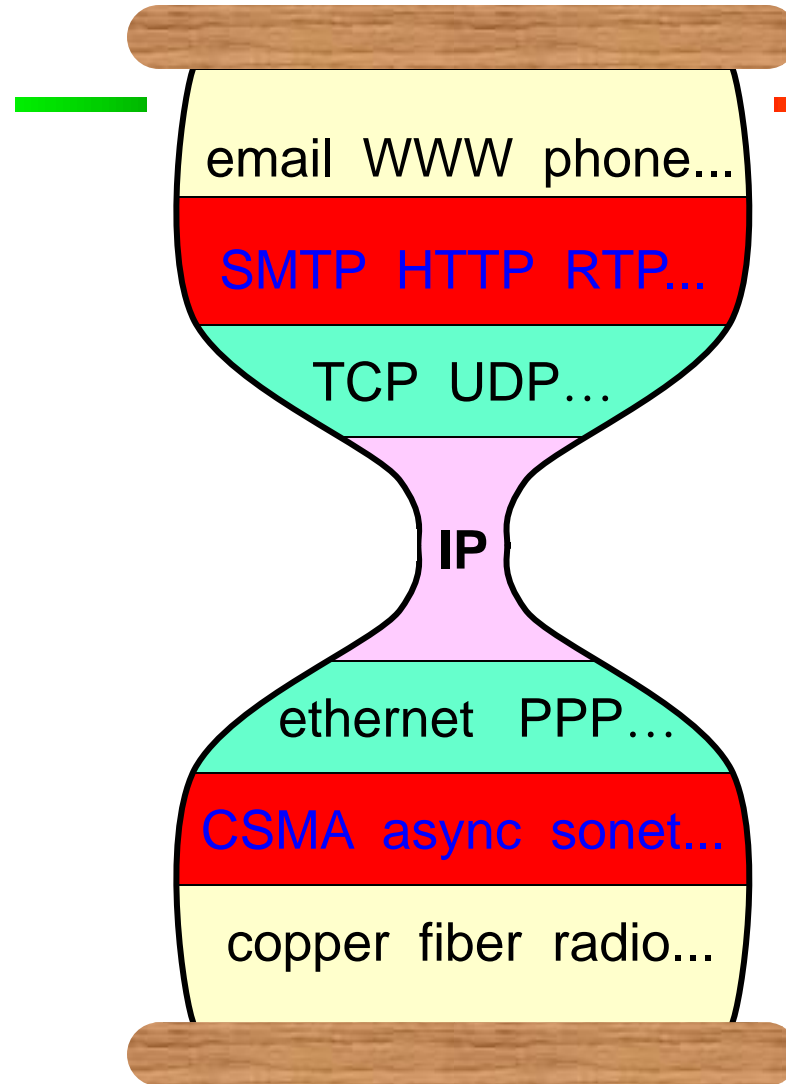
TCP/IP参考模型的缺点：结构不严谨、不科学、不完善。

在实践中不断发现问题，不断完善——**打补丁**。

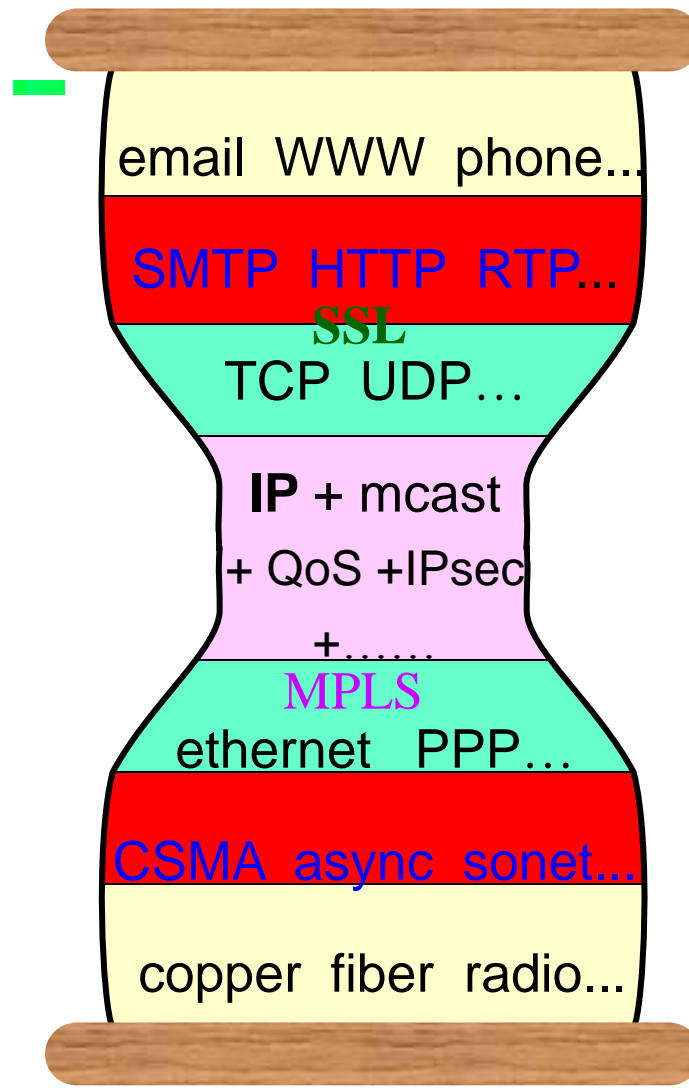
TCP/IP的补丁

- 不支持快速转发——MPLS (2.5层)
- IP地址空间不够——IPv6
- 网络层不安全——IPSec (3.5层)
- 没有数据加密、压缩等功能——增加SSL (增加会话层和表示层) (4.5层)
- 应用层：电子邮件不安全——安全电子邮件协议 (SMIME、PGP、PEM等)
- 不支持QoS——制订一组多媒体传送协议 (RSVP、RTP/RTCP、RSTP等)
- DNSSec: 在现有DNS协议上增加加密和认证功能
-

The Internet Protocol Hourglass (Deering)

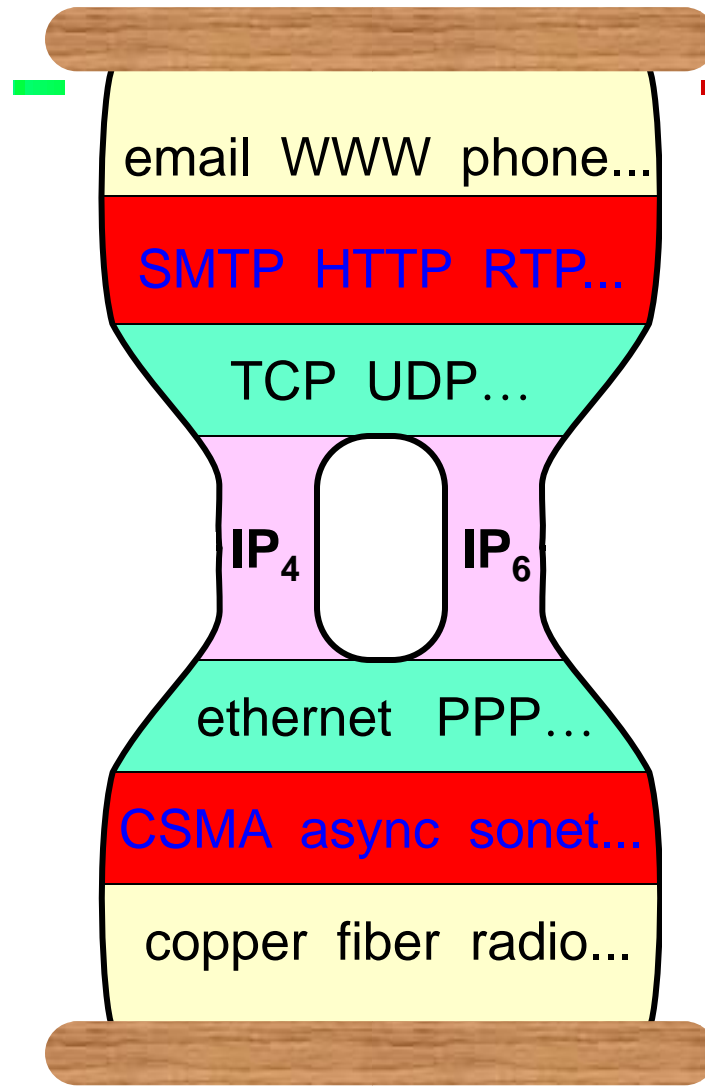


Putting on Weight



- requires more functionality from underlying networks

Mid-Life Crisis



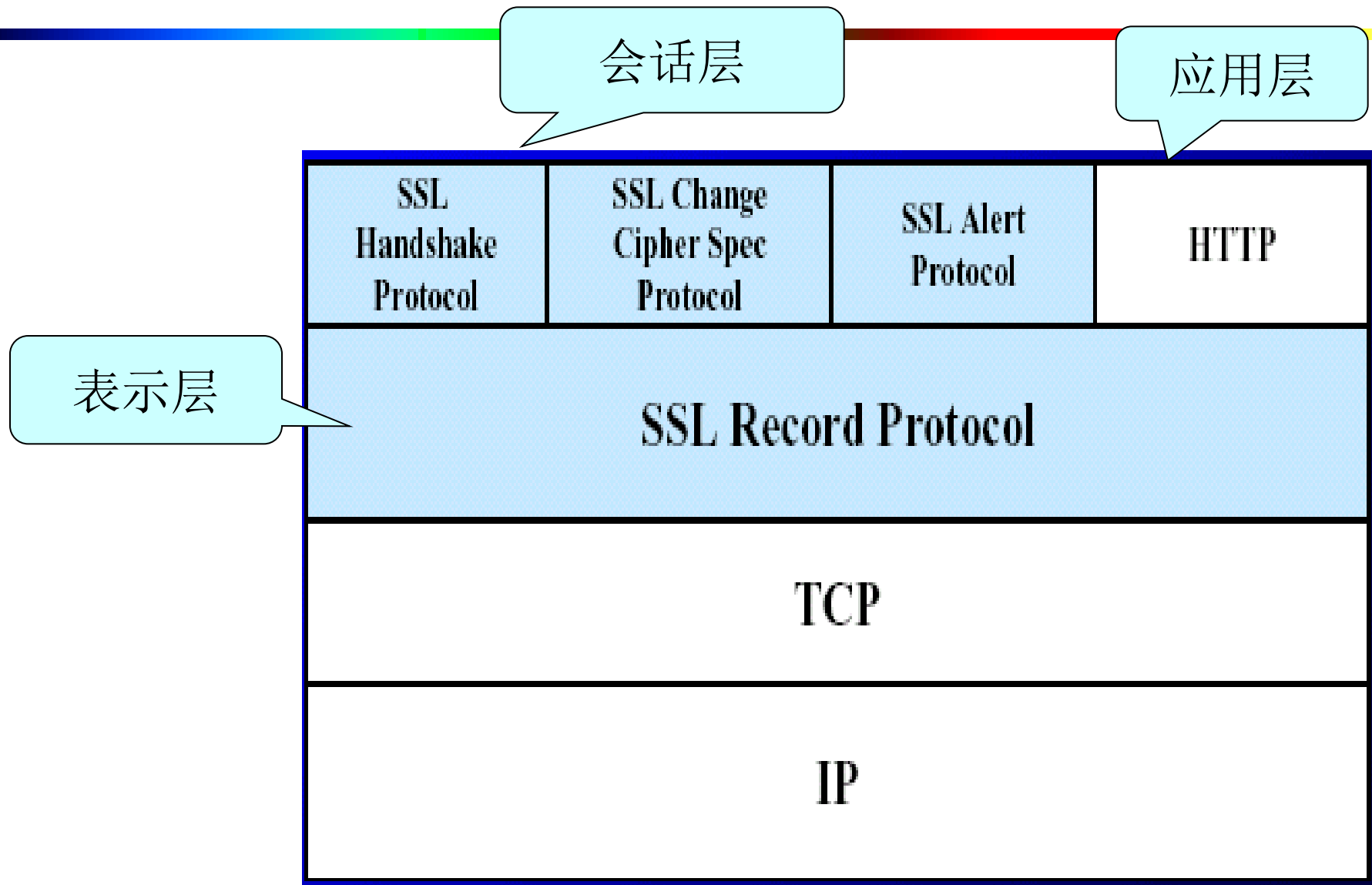
- doubles number of service interfaces
- requires changes above & below
- major interoperability issues

TCP/IP的补丁之一——SSL

SSL协议主要包括两个方面:

- ✓ SSL Handshake Protocol (会话管理, 协商算法, 产生密钥, 身份认证等)
----会话层功能
- ✓ SSL Record Protocol(对发送的数据进行加密、压缩等)
----表示层功能

TCP/IP 的补丁之一——SSL



完善之后的TCP/IP协议

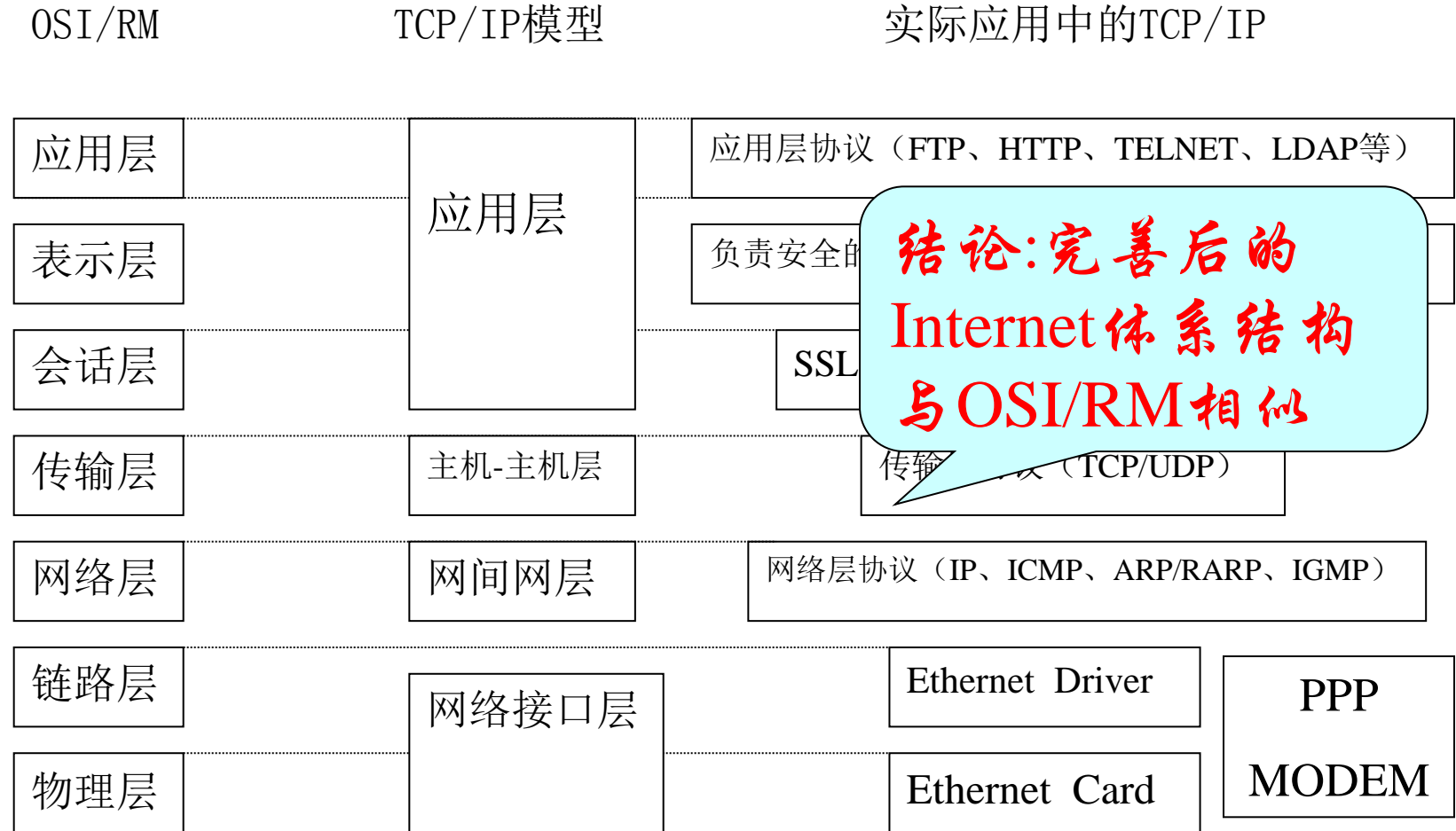
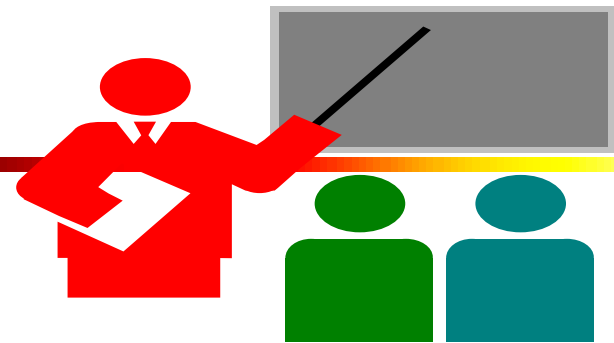


图 OSI/RM、TCP/IP模型与实际中应用的TCP/IP之间的关系

其它网络体系结构

ISO/OSI	ARPA	SNA		DNA	TCP/IP
应用层	用户层	端用户		端用户	应用
表示层	文件传输协议 远程通信协议	功能管理服务		网络应用	
会话层		数据流控制		会话控制	
传输层	HOST-HOST	传输控制		网络服务	TCP/UDP
网络层	源IMP-目的IMP	通路 控制	路由控制	传输控制	IP
数据链路层	IMP-IMP		传输组	数据链路控制	数据链路
		数据链路控制			
物理层	物理层	物理层		物理层	物理层

Summary



- ☐ 计算机网络的基本概念 ✓
- ☐ 计算机网络体系结构 ✓
- ☐ OSI/RM ✓
- ☐ TCP/IP ✓

Quit 2

1. 对具有 N 层结构的网络，设用户交给网络数据长度为 M 字节，每层协议头为 H 字节，试计算协议头在底层数据流中所占比例。

Thanks!

