



chapter 2 计算机网络的层次化体系结构

2.1 OSI参考模型

2.1.1 七层模型

2.1.2 层，协议，及服务

2.2 TCP/IP参考模型

2.2.1 TCP/IP 协议栈结构

2.2.2 TCP/IP 协议各层的功能演示

2.2.3 协议简介

2.3 ATM体系结构



网络功能的复杂性和模块化设计的必要性

网络需要为应用完成的工作：

- 编址、路由
- 在用户之间分配或共享带宽，并充分利用网络资源
- 传输用户任意长度的信息
 - 无错传输
 - 安全传输
 - 延时保障
- 支持尽可能多的用户
- 链路状态维护
- 等等

网络处理的事情过于复杂，人们发现有必要把不同的功能模块化，然后像搭积木一样，形成一个完整的网络



网络建模

- 网络建模是对以上问题的抽象
- 建模后可提供可视的、模块化的表示，并可清晰呈现相互之间的关系。这里的模块是层（协议）
- 有助于工程人员定位、描述问题所在



协议和分层

协议和分层是把网络功能分块的有效方法

- ✓ 每一层协议都认为，自己在用本层协议与通信对端的同一层“直接”通信。
- ✓ 每一层协议在使用低层协议提供的服务基础上，实现自己的功能，同时可为高层提供新服务。
- ✓ 系统中使用的协议集合，称为协议栈



服务和接口

- 服务: 每一层通过接口提供给高层的功能
- 接口: 位于层与层之间, 用于传递服务请求和服务结果
- 每一层通过接口调用低层 (≠底层) 提供的服务, 不必关心低层内部关于服务的实现细节
 - 实现层与层之间的“脱耦”
 - 只要保证接口不变, 任一层的更改不影响其它层



协议栈示例

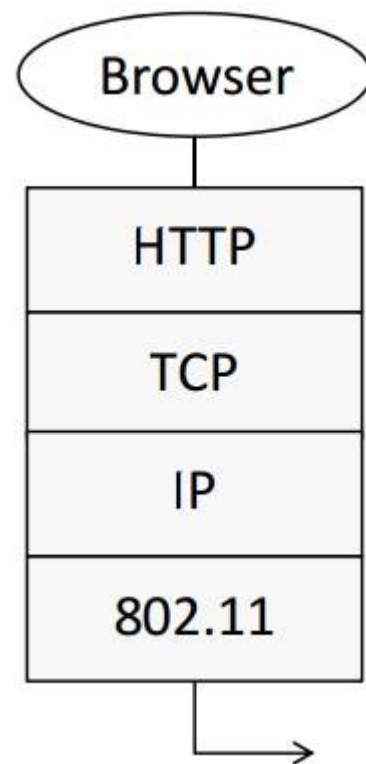
- 可能听说或者熟悉以下协议:

TCP, IP, 802.11, HTTP, ...

- 它们可能组成如右图的协议栈

右图是通过wifi接入互联网，利用浏览器访问web页面的一个协议栈结构

这体现这几个协议之间的关系





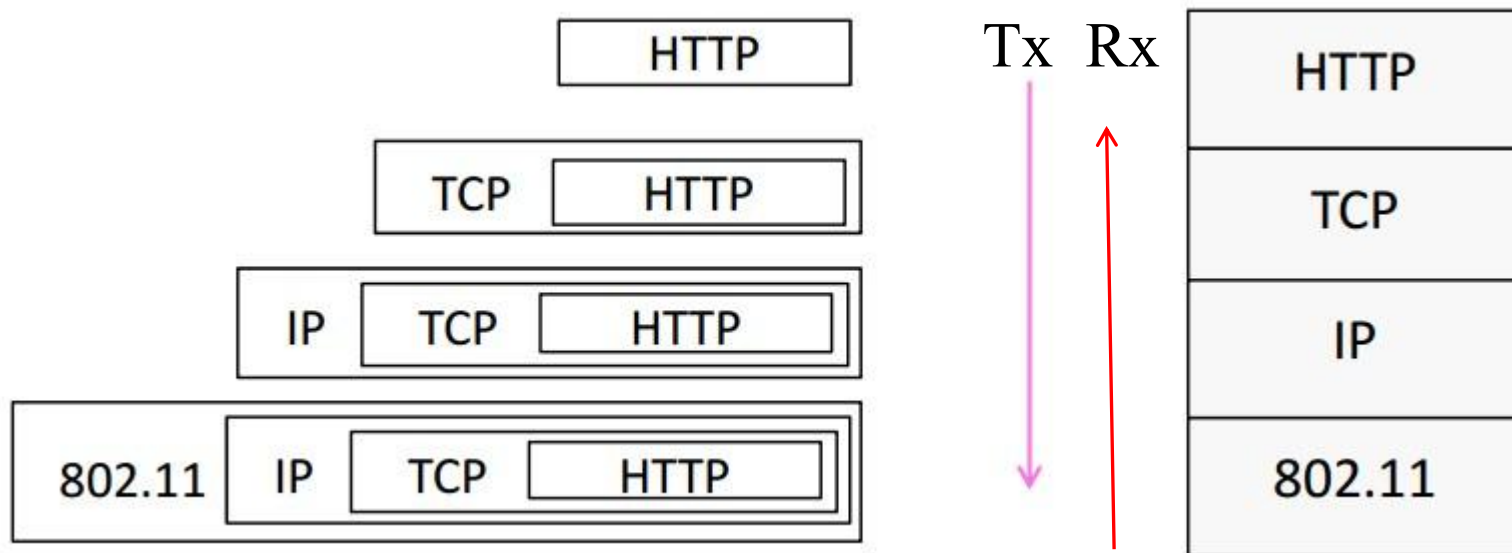
封装 Encapsulation

- 封装是用于实现协议分层的机制
 - 低层封装高层的内容，并加入本层必要的信息（通常加在原有内容的前面），生成用于传输的新消息
 - 在此过程中，低层并不查看高层递交了什么具体内容，低层只负责完整地转交，不必理解内容是什么
 - 日常举例：邮局寄信



封装示例

- 线路中传递的消息，被层层包裹，如同洋葱一样，最底层处于最外围





分层的优点

- 通过把整体功能分隔为若干部分，分层简化了网络的设计、开发和测试等过程
- 每一层的协议可以独立于其它层进行设计. 从而可隐藏内部实现细节，实现层间脱耦
- 可以在不涉及低层的情况下，对各层协议进行修改、升级，分层带来了灵活性！
- 平面状的非分层结构开发和维护的代价昂贵，也不灵活，是过时的技术
 - 特殊场景？



协议分层的缺点

- 额外开销增加

随着带宽增加，这些额外开销越来越不是问题；

对于长消息来说，增加的百分比有限！

- 隐藏了其它层可能需要的信息

举例？



2.1 OSI参考模型

ISO: International Organization of Standardization

国际标准化组织

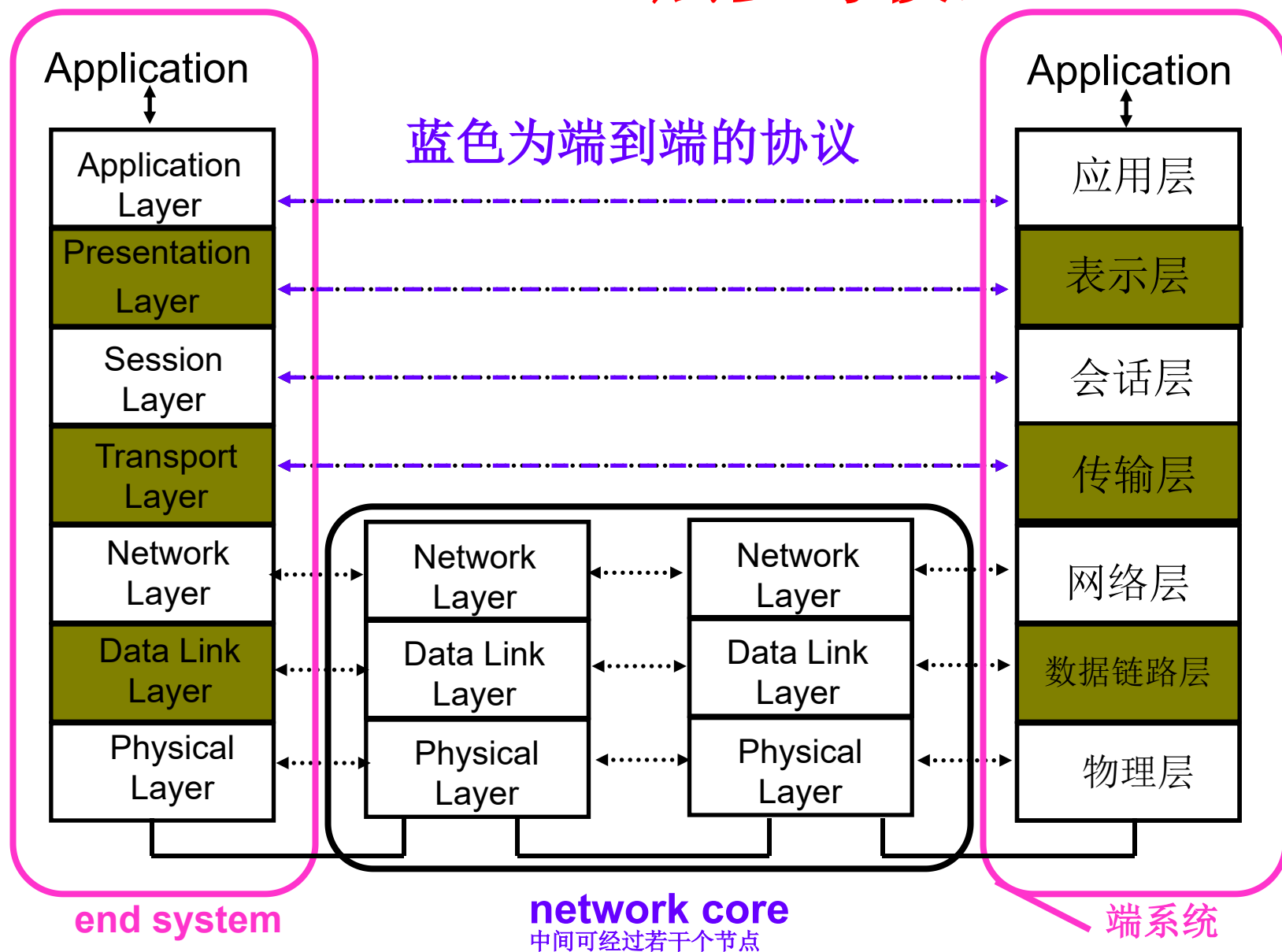
OSI: Open System Interconnection 开放系统互连

协议Protocol: 相互通信的双方关于如何通信的约定

协议栈Protocol stack: 特定系统使用的一组协议，每层至少一个, 称为协议栈



2.1.1 OSI七层参考模型





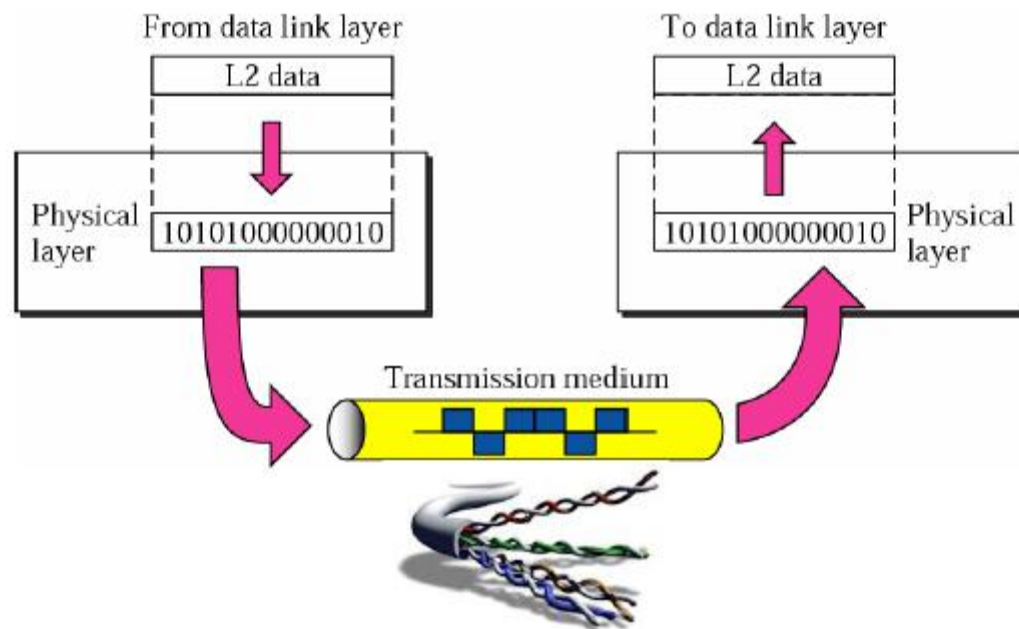
物理层 Physical Layer



- 物理层的任务是在一条链路（信道）上透明地传输比特流
- 物理层定义了通信链路的物理参数
 - 机械特性Mechanical: 线缆类型, 插接件规格, 引脚参数...
 - （光）电特性: 调制方法, 信号强度, 电平, 比特时间等, ...
 - 功能或规程特性: 如线路的激活、状态的维护等
- 常见的物理层技术: Ethernet, DSL, modems...
- 在OSI参考模型中, 物理介质并不包括在物理层中
- 常用的物理层介质: 双绞线Twisted-pair cable, 同轴电缆 coaxial cable, 光纤optical fiber, 无线radio...



物理层的工作模型

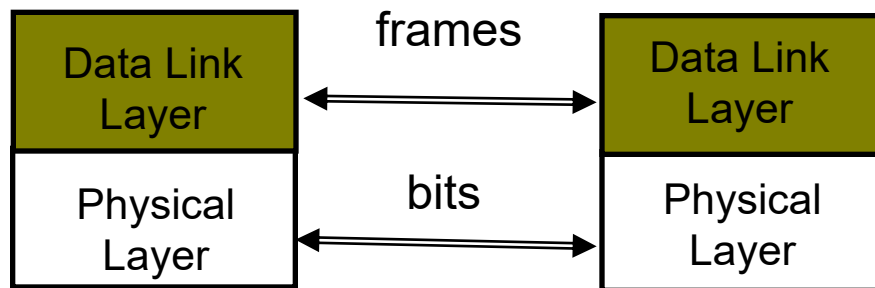


- Physical media **is not included** in physical layer



数据链路层Data Link Layer

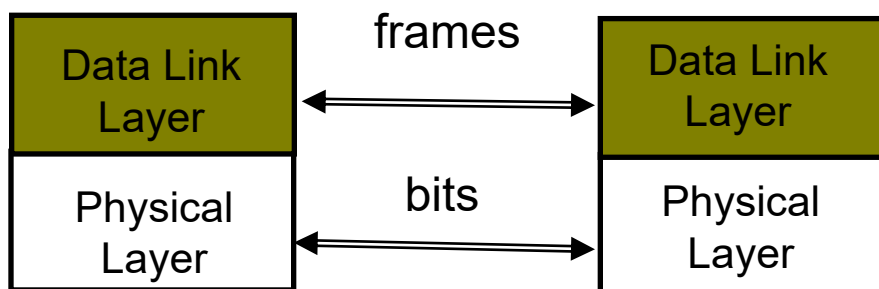
- 在直接相连的链路上，传输以帧(*frames*)为单位的数据
 - 引入了帧同步的问题：如何在物理层递交的比特流中确定帧的边界？
- 链路上存在多用户时，需要区分不同用户
 - 引入了编址的问题：称为物理层地址，或者MAC地址
 - 此时，地址字段将作为本层信息的一部分





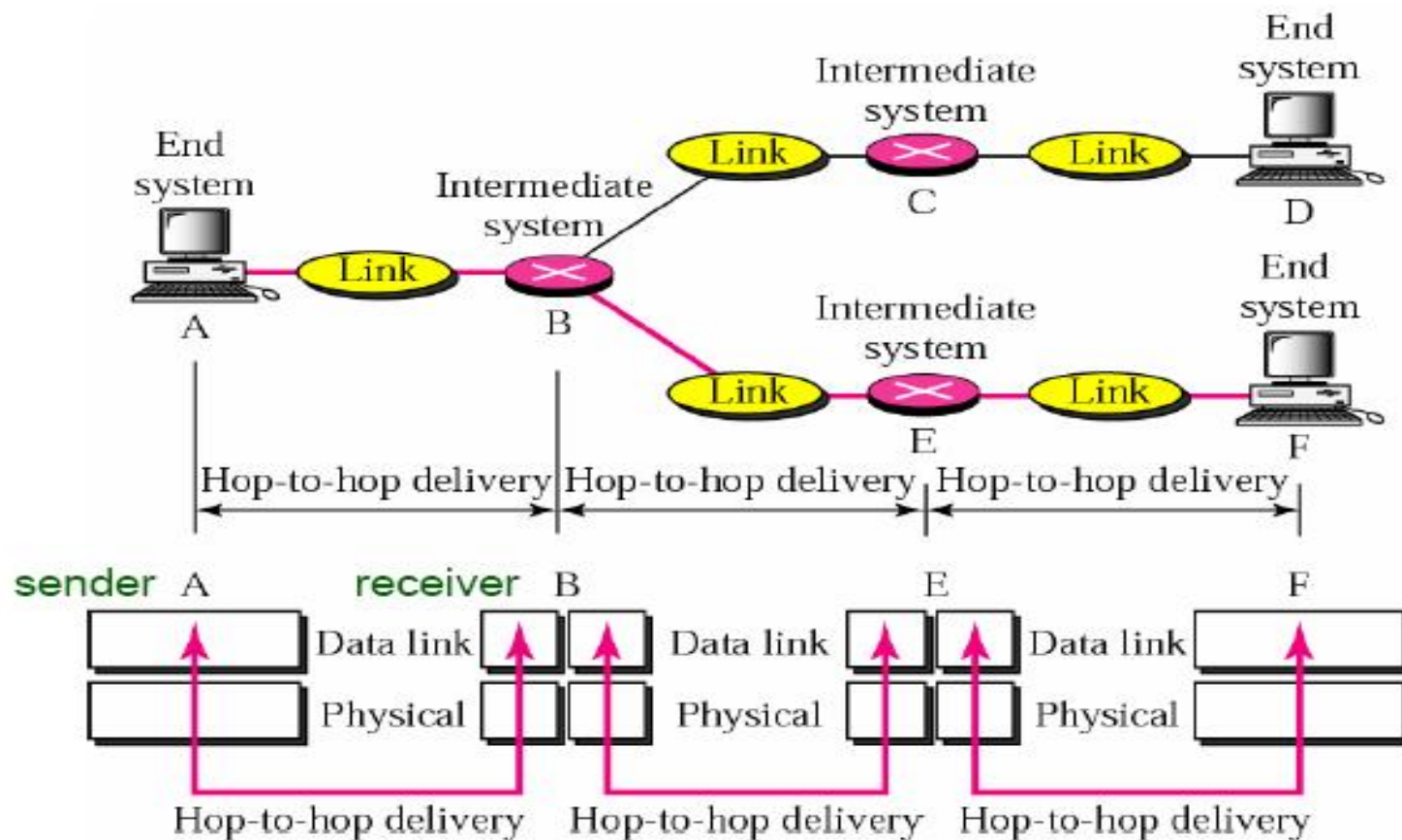
数据链路层常见功能

- 差错控制：发现差错；解决差错
- 媒体接入控制：
- 流量控制 Flow control：如果信息传输速率超过了接收者的处理速率...
- 连接管理：链路激活、维护，释放等操作





数据链路层（二层）互联的网络示意图





网络层Network Layer

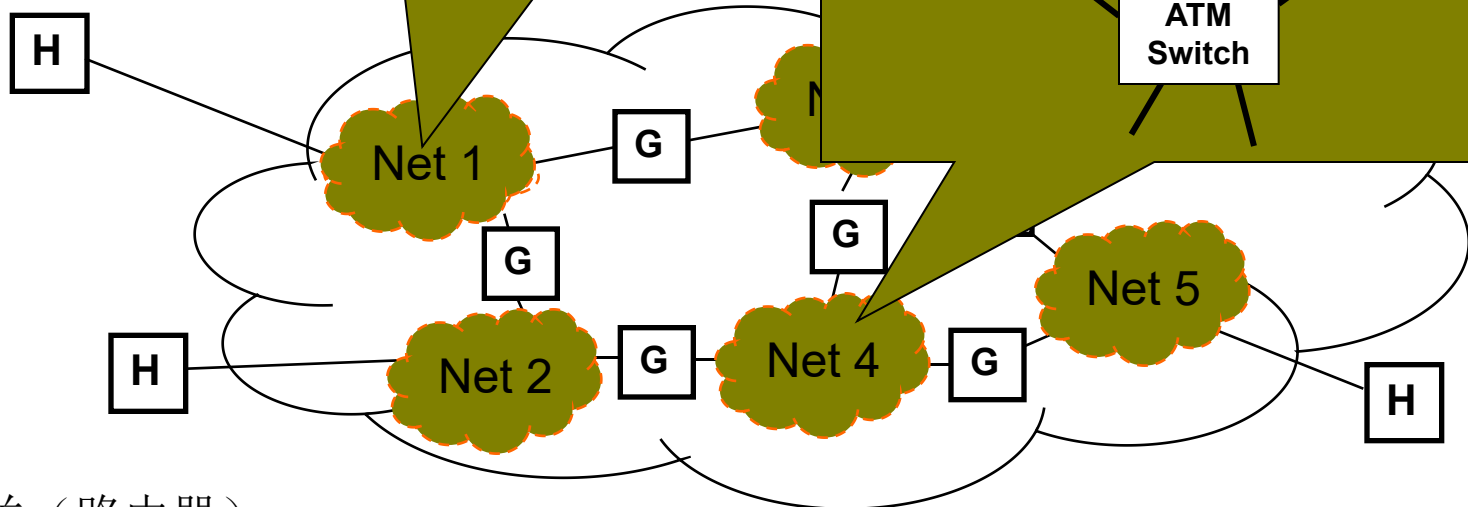
网络层的任务是在不同的链路上转发分组，主要功能是路由和转发：

- 编址
- 建立路由表：选择到目标节点最佳的下一跳节点
- 根据路由表，将分组转发给对应的下一跳节点
 - 可能涉及到分片和重组
- 拥塞控制Congestion control
- 网络层的连接管理（when connection-based）



网络互联

网络互联是网络层的工作之一，利用路由器（或网关）在多个可能不同结构的网络之间转发分组；转发的依据首先是逻辑地址（不同于MAC地址）；每个中间节点必须具有低三层的功能



G = 网关（路由器）

H = 主机（终端）

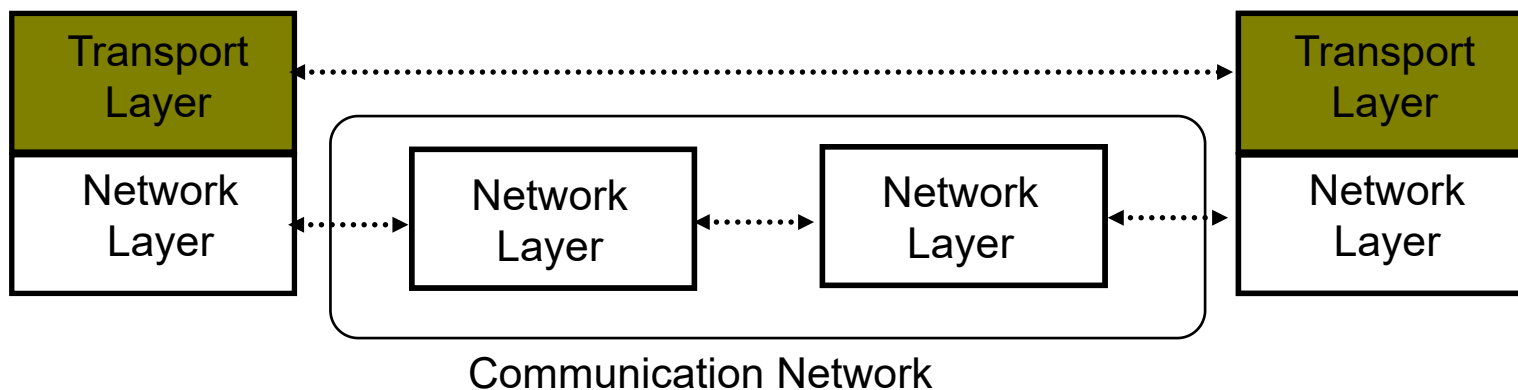


传输层Transport Layer

在终端的两个进程间传输数据：

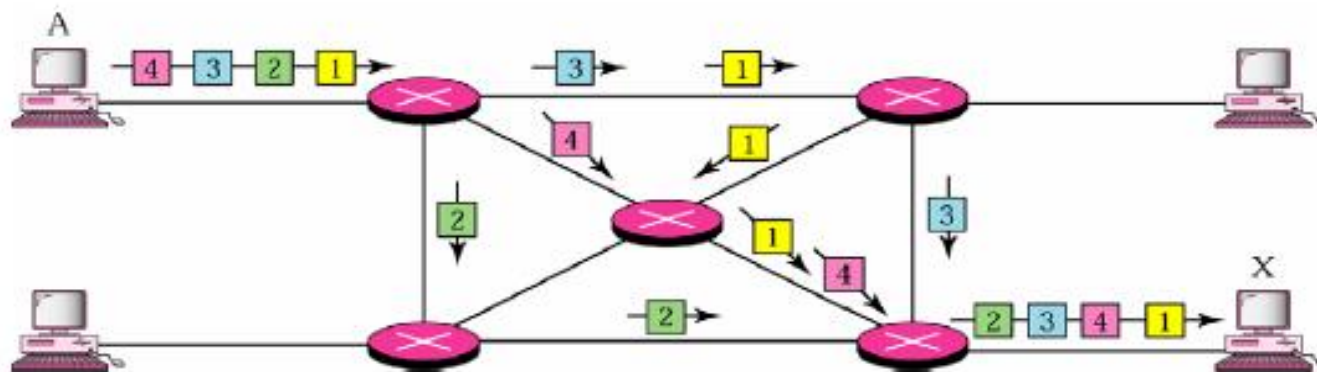
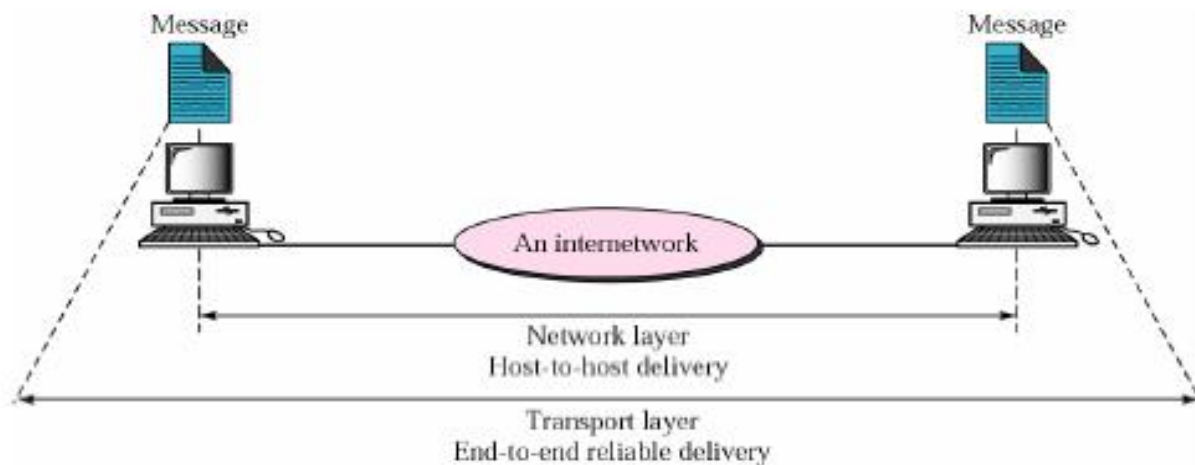
常见两种类型：可靠的传输（一般是面向连接的）和不可靠的传输（一般是面向无连接的）

传输层通过不同的端口为不同的应用提供服务





传输层负责端系统上不同进程间的消息传递. 而网络层负责传递到正确的端系统上

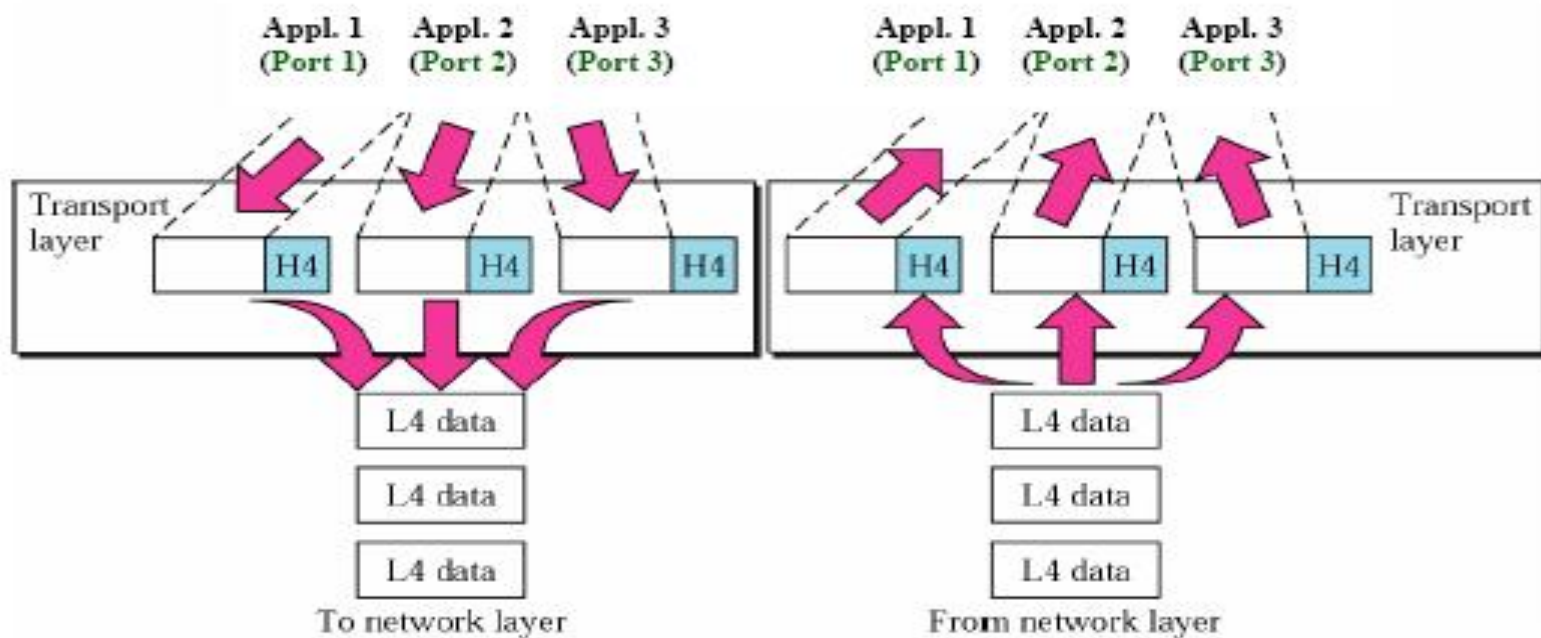




传输层的主要功能

基于端口的寻址：端系统可能同时运行多个进程

流量控制和差错控制：注意与二层的区别





其它端到端的层:

- 会话层Session layer:

数据的交换方式(如 半双工half-duplex, 全双工full duplex等), 数据同步控制点等等

- 表示层Presentation layer:

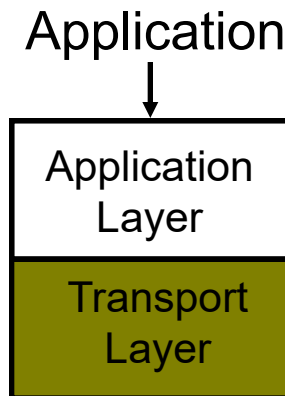
数据的表示问题, 通常解决的是机器相关的表达与机器无关的表达之间的转换

- 应用层Application layer:

提供应用层所需的服务. 如WWW业务, 浏览器使用HTTP应用层协议获取文档.



表示层、会话层和应用层的合并

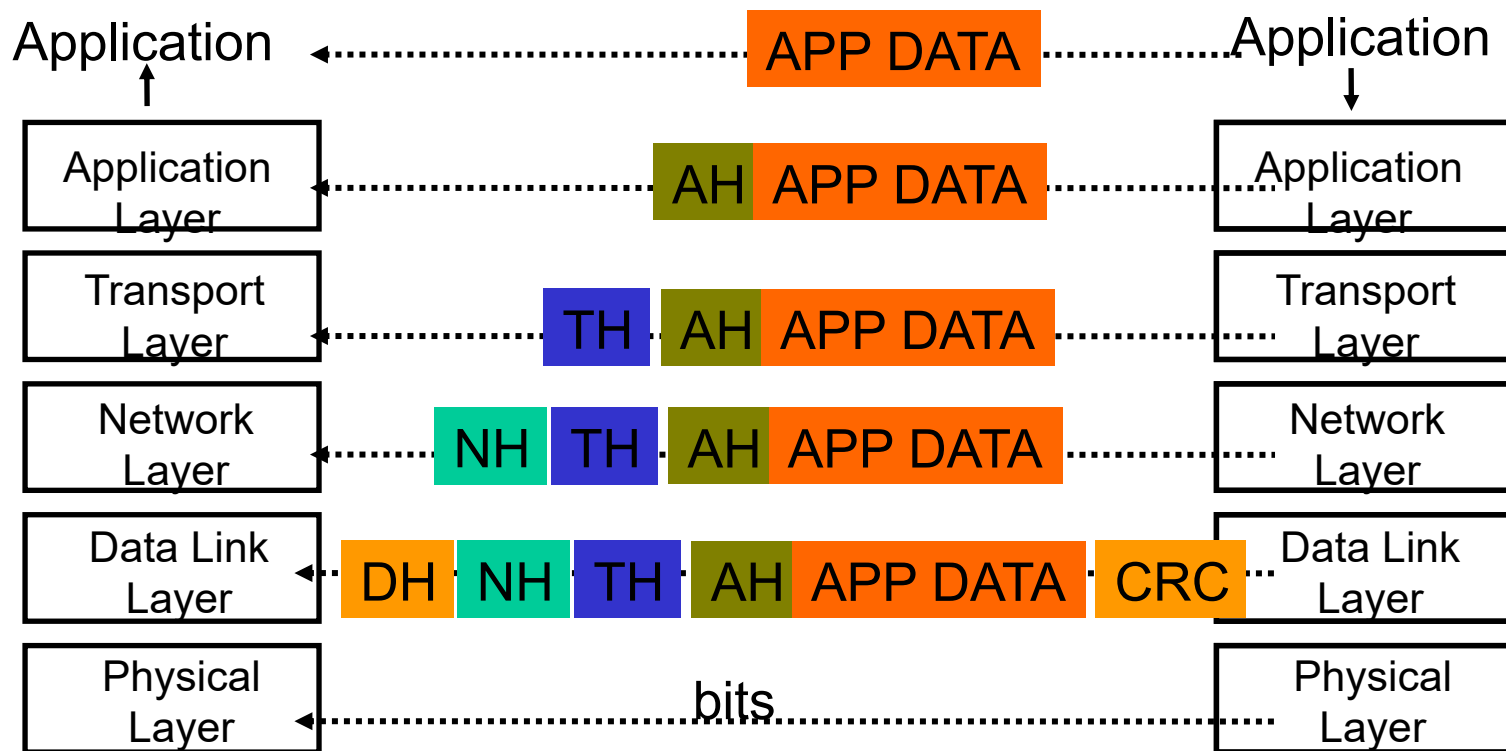


**Incorporated into Application Layer in Internet.
But...**



协议的头尾字段

- 每个协议都会在高层的数据前，增加自己的协议控制消息，如标志位、地址字段等等；
- 对于二层来说，除此之外还会在高层的数据尾部，增加一个校验字段，通常是CRC校验





2.1.2 分层、协议、服务

SAP: service access point, 服务访问点, 通常是一个软件或硬件接口

SDU: service data unit, 业务数据单元

Layer n entities: n层实体

PDU: protocol data unit 协议数据单元, 各层与对等层交换的数据单元

封装: $n+1$'s PDU = n 's SDU n 's PDU = n 's PCI + n 's SDU

在高层PDU的基础上增加本层的控制消息, 变成本层的PDU



2.1.2 分层、协议、服务

Service: 服务是n层实体提供给n+1层实体的功能, 服务可以是面向连接的或者面向无连接的, 可以是可靠的也可以是不可靠的. 如: $n \rightarrow n+1$, (e.g. $n+1$: HTTP, n : TCP; $n+1$: DNS, n : UDP).

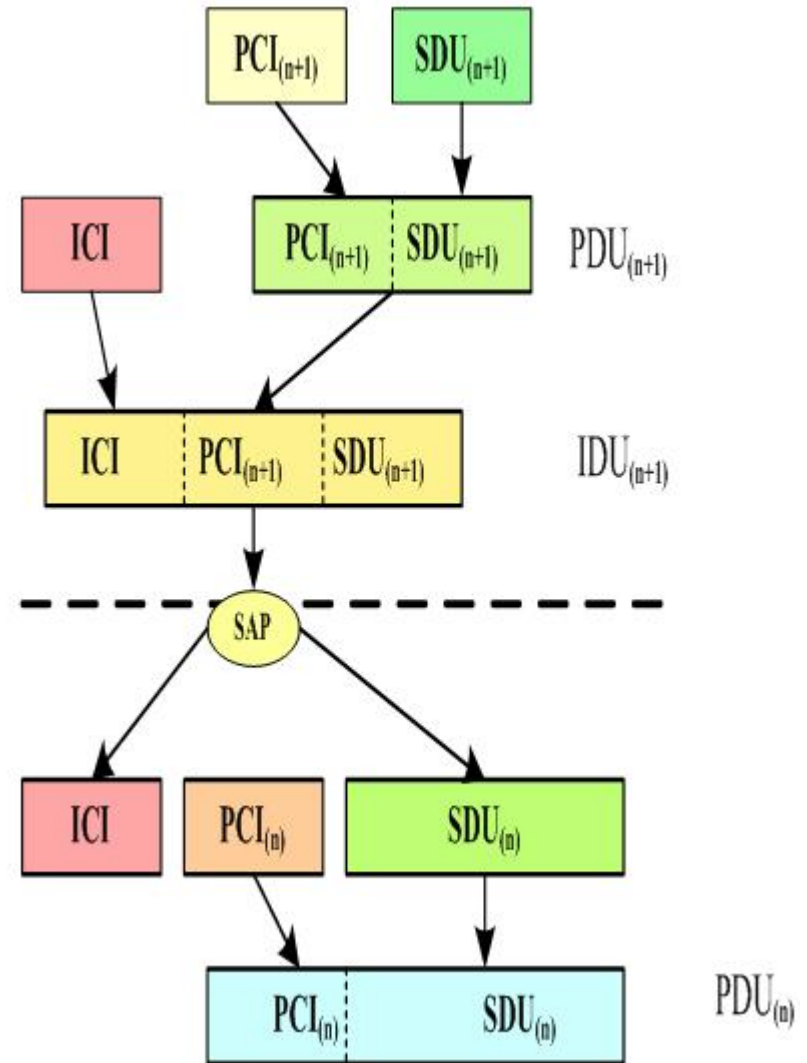
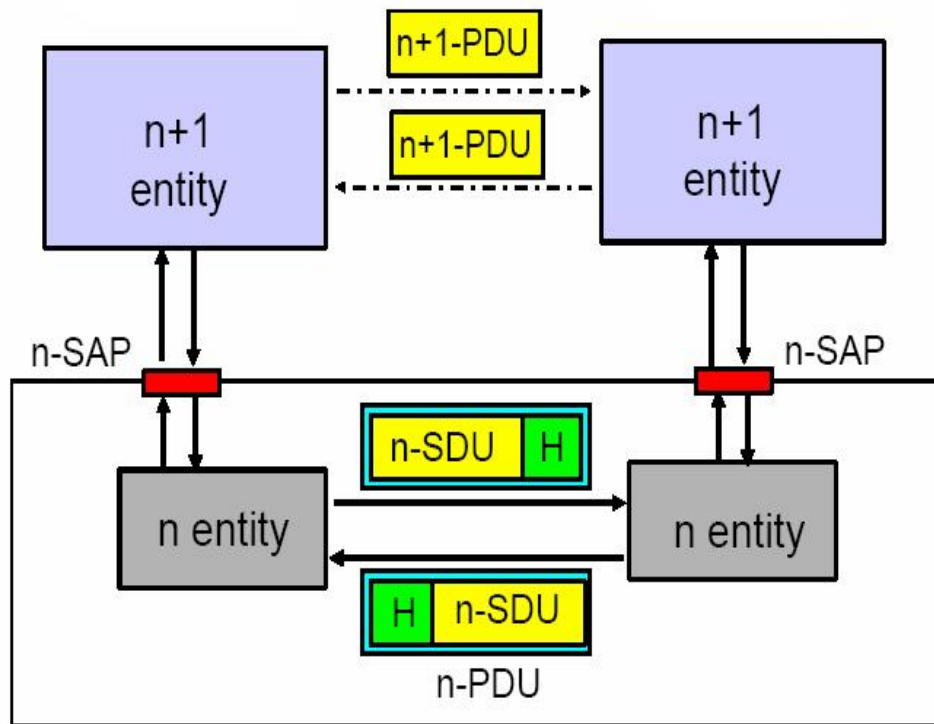
IDU: Interface Data Unit 接口数据单元, 指相邻层经过SAP点的数据

ICI: Interface Control Information 接口控制信息, 指IDU中的控制信息部分

PCI : Protocol Control Information 协议控制信息, PDU中的控制信息部分, 体现了协议的控制消息



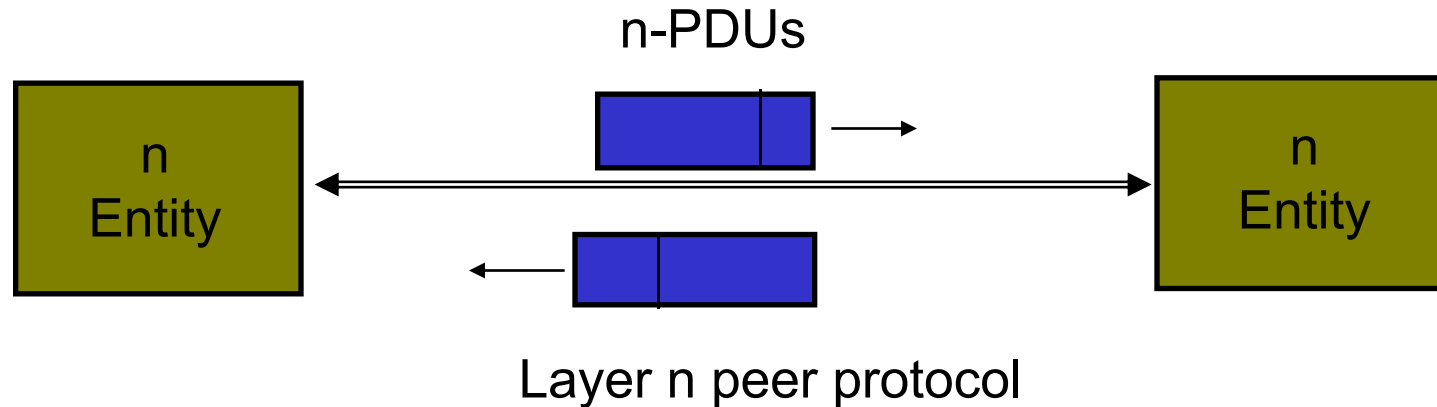
协议分层后的数据关系





ISO/OSI下的协议

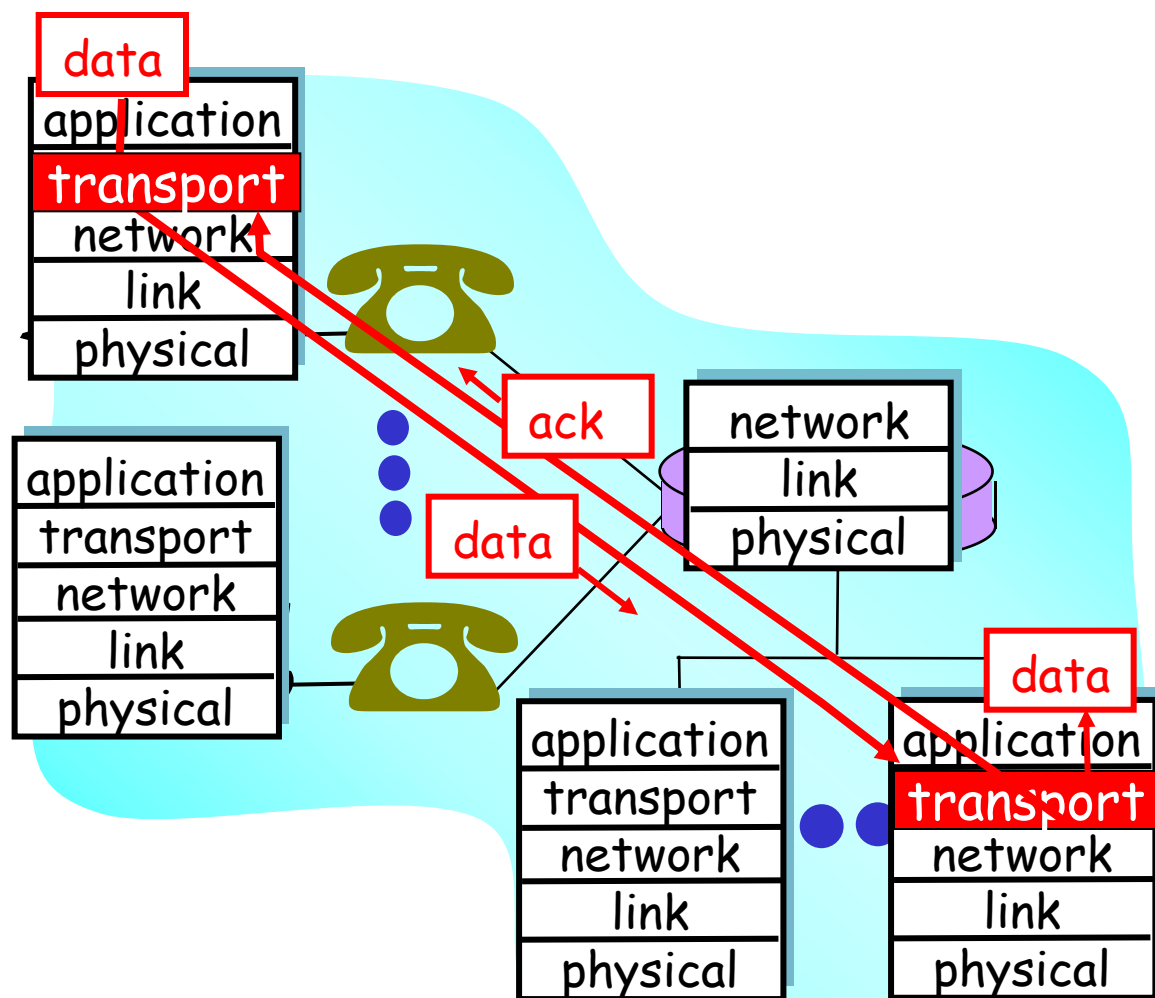
- n层协议与通信对端的n层协议交互，并为n+1层提供服务
- 对应层的实体称为对等进程
- n层对等进程交互所使用的规则，称为n层协议，它们交互的内容称为n层的协议数据单元n-PDU





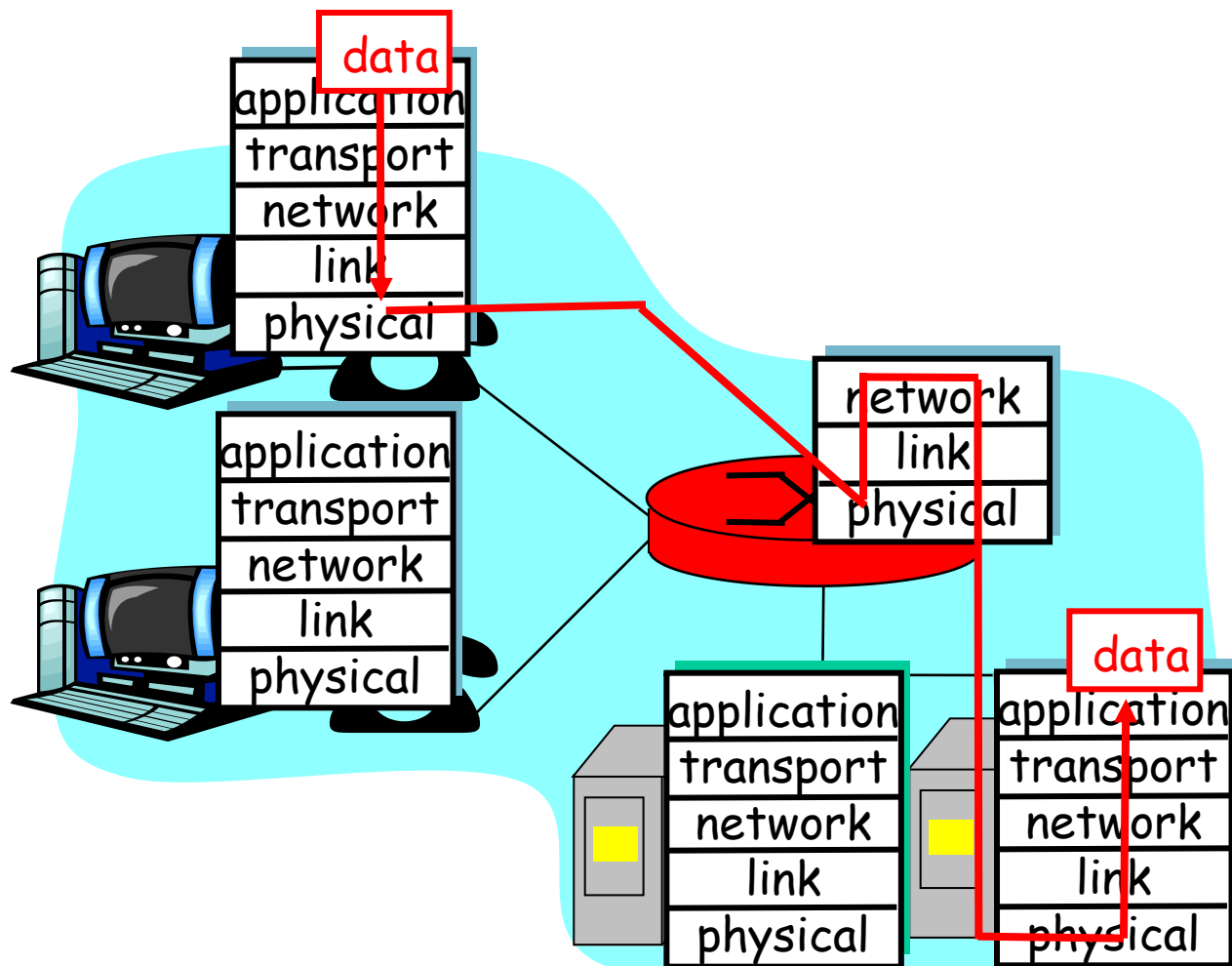
协议分层：协议角度的通信过程（逻辑过程）

以传输层为例：
从高层取数据，可靠传输给对方传输层，对方传输层递交对方高层





协议分层：实际的通信过程（物理过程）



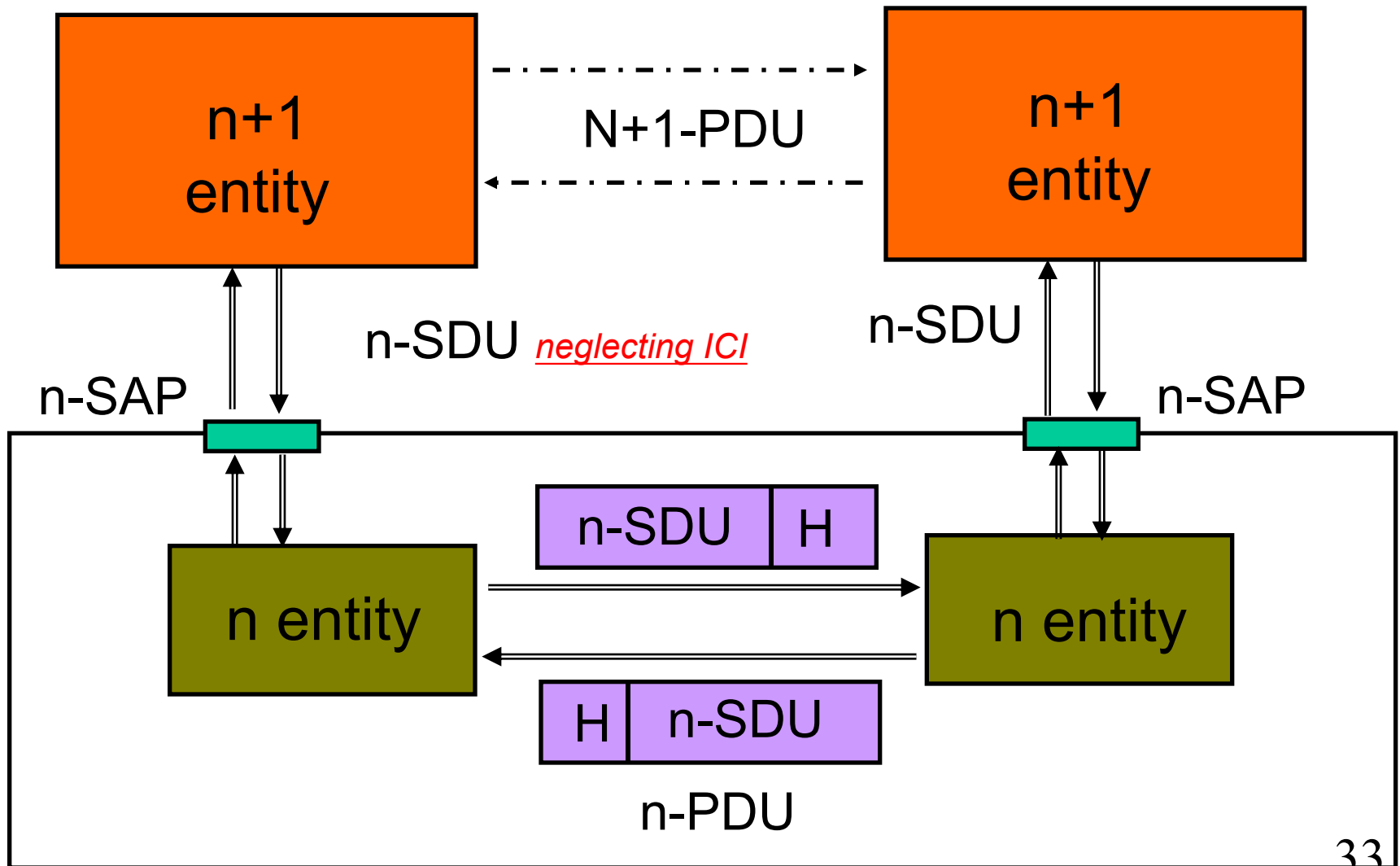


ISO/OSI下的服务

- n 层协议之所以能将通信过程视作直接与对等层的交互，是因为它调用了 $n-1$ 层的服务
- 服务通过服务访问点 (*Service Access Points* SAP) 获得，服务访问点可能是一个软件的函数调用
- 每一层将自身的数据和控制消息及高层的PDU一起，传递给下一层，直到在物理层进行真正的传输过程
- 高一层的PDU即自身的SDU
 - 自身的PDU呢？

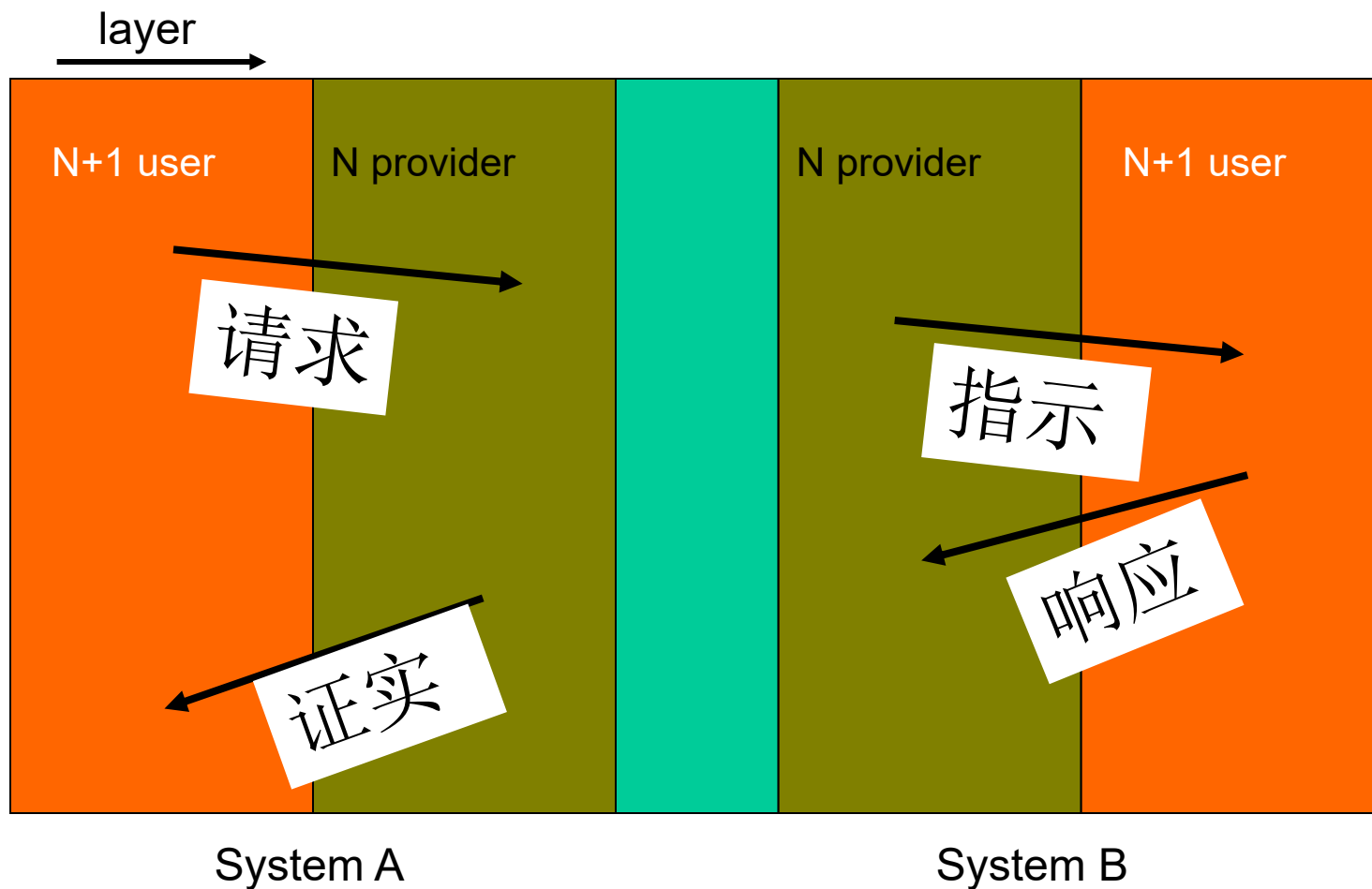


层，服务，和协议



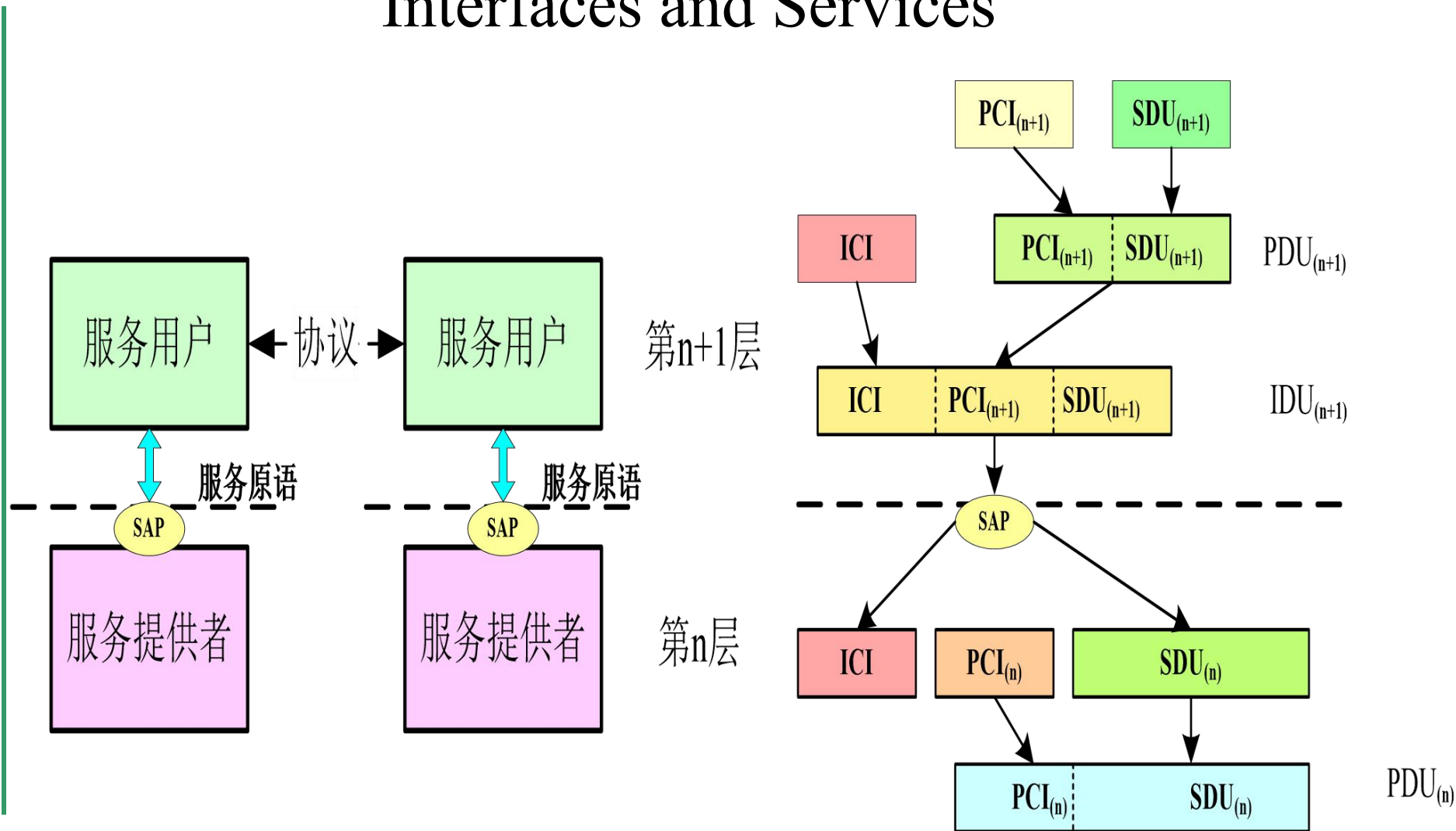


层间交互：服务原语





Interfaces and Services





服务类型：面向连接的和无连接的服务

- 无连接的特点
 - 直接进行SDU的传输
 - 没有连接建立、维护、拆链的过程
 - 服务的典型特征：无序、尽最大努力服务 (Best-effort)
 - 例如. UDP, IP



服务类型：面向连接的和无连接的服务

- 面向连接的：
 - 连接建立的三个阶段：
 - 1.待建立连接的SAP之间启动连接，并初始化连接状态
 - 2.传输SDU
 - 3.拆除连接
 - 服务的典型特征：有序、可靠
 - E.g. TCP, ATM



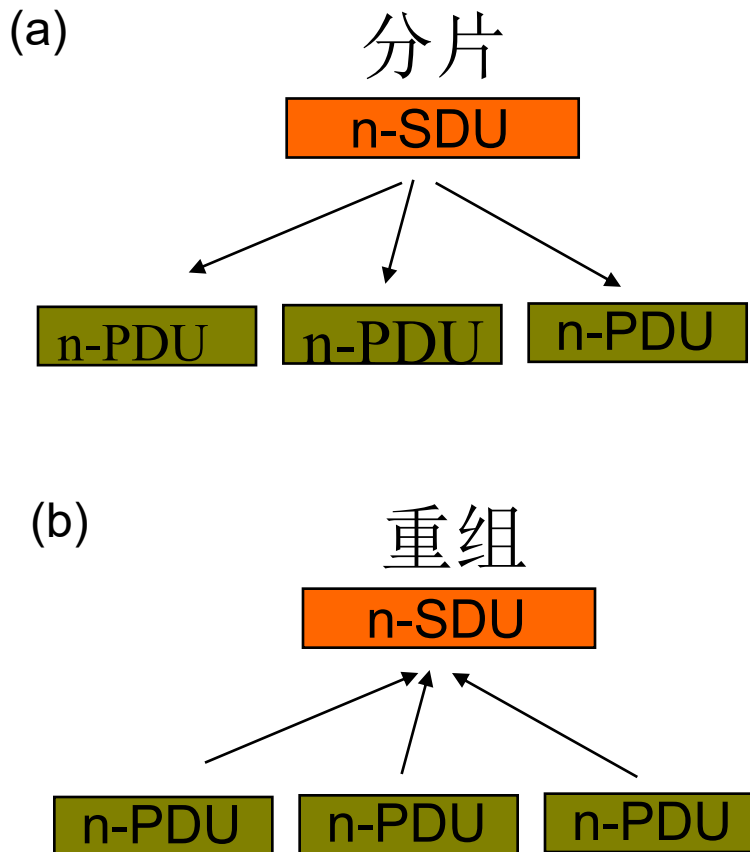
服务类型：面向连接的和无连接的服务

- 不要求所有层，全是连接或者全是非连接的，不同层可以不一样，如IP是非连接的，使用IP服务的TCP是面向连接的
- 各层提供的服务可以是确认的或者非确认的 (*confirmed or unconfirmed*)，一般来说面向连接提供的是确认的服务，无连接可以是确认的或者非确认的



分片和重组

- 每层可以限定单次传输的数据块大小，各层的大小没有强制性要求，即不要求低层必须装下一个高层的PDU
- 当高层的数据量太大时，发送时将之拆为若干个分片，接收时重组恢复会原始大小
- 缺点？



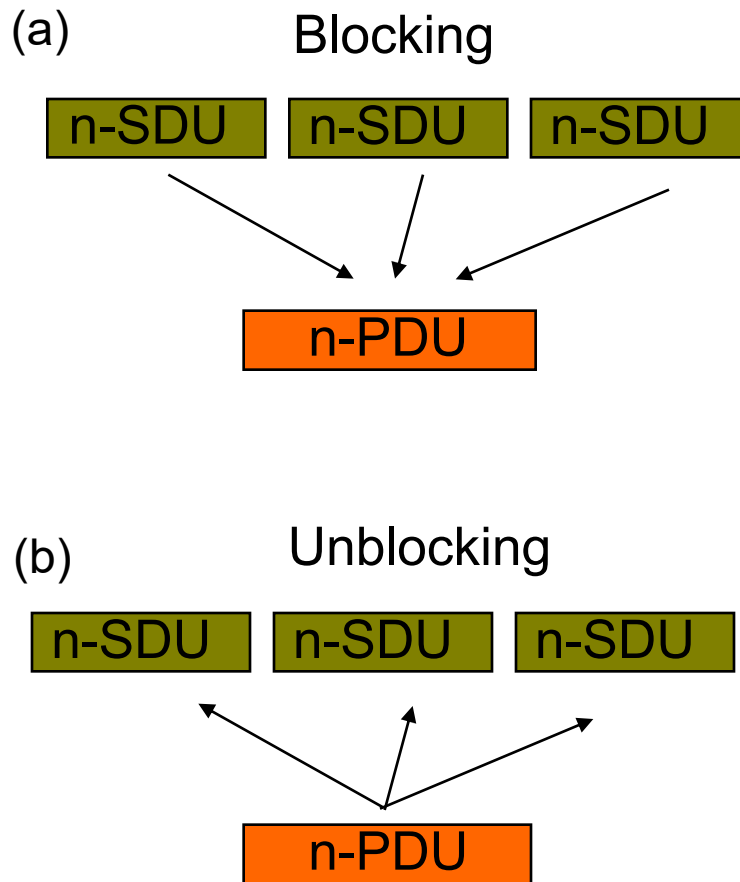


组块与分块 Blocking & Unblocking

- 高层的数据量太小，以至于影响到低层的效率

- why

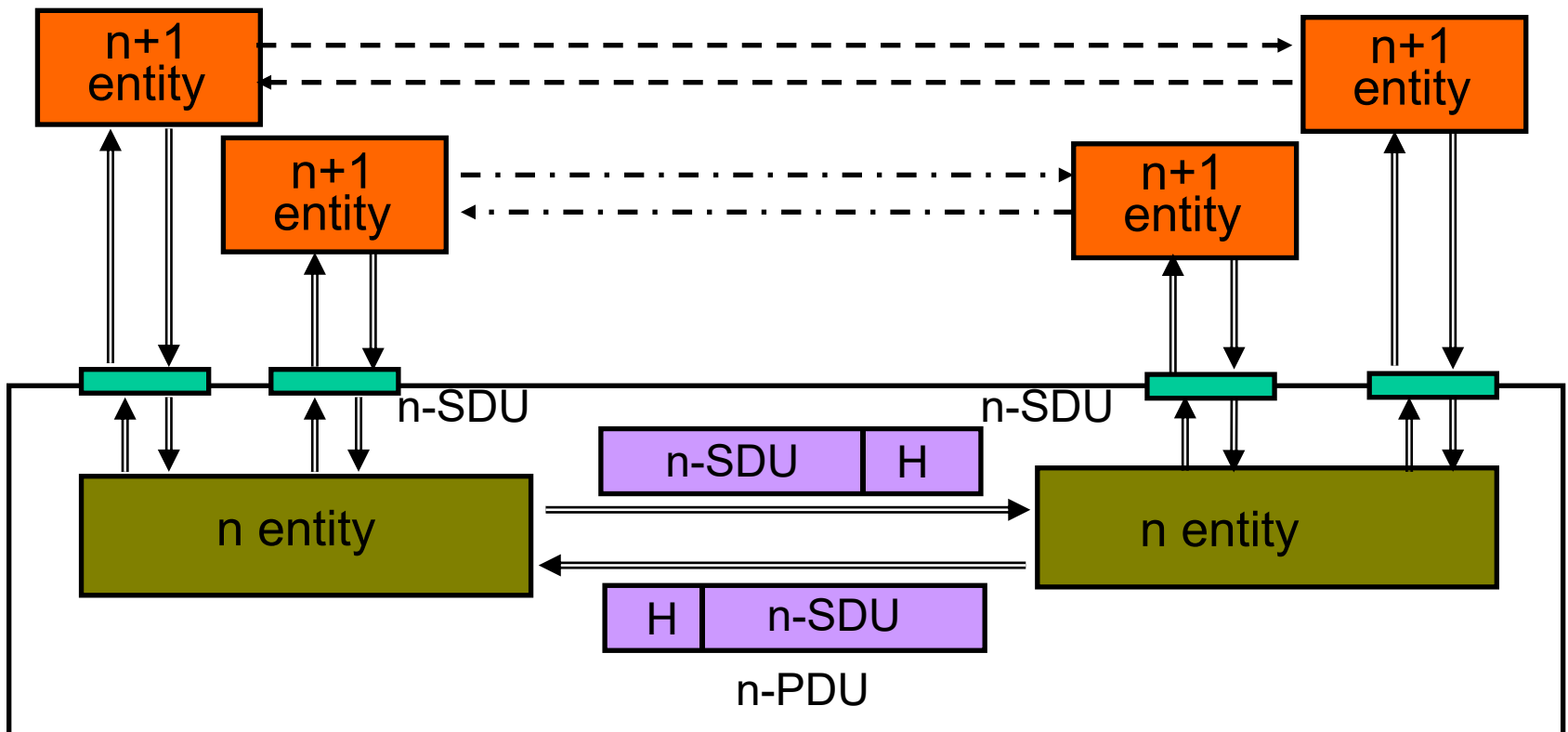
- 发送时，将多个高层 PDU（即本层SDU）组合为一个数据块整体发送；接收时，将该数据块恢复为原始的若干小数据块





复用与解复用 Multiplexing & Demultiplexing

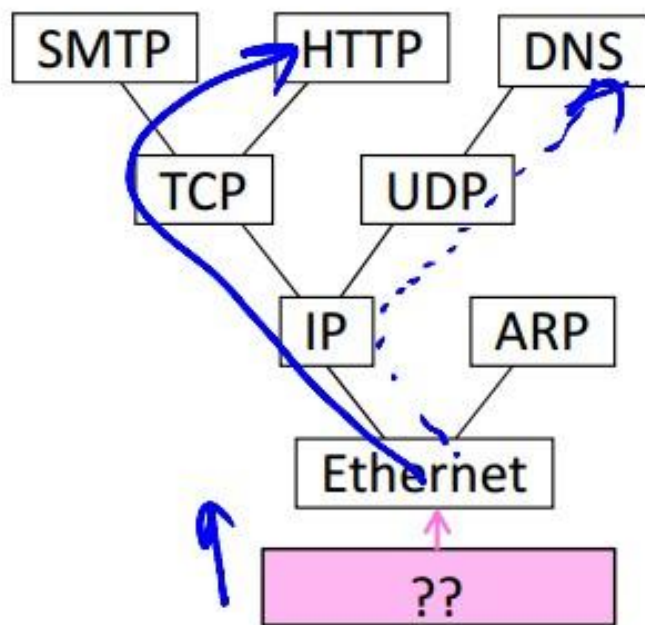
- 多个高层协议可以共用同一个低层的服务
- 复用后需要考虑解复用时如何区分不同的高层





解复用示例

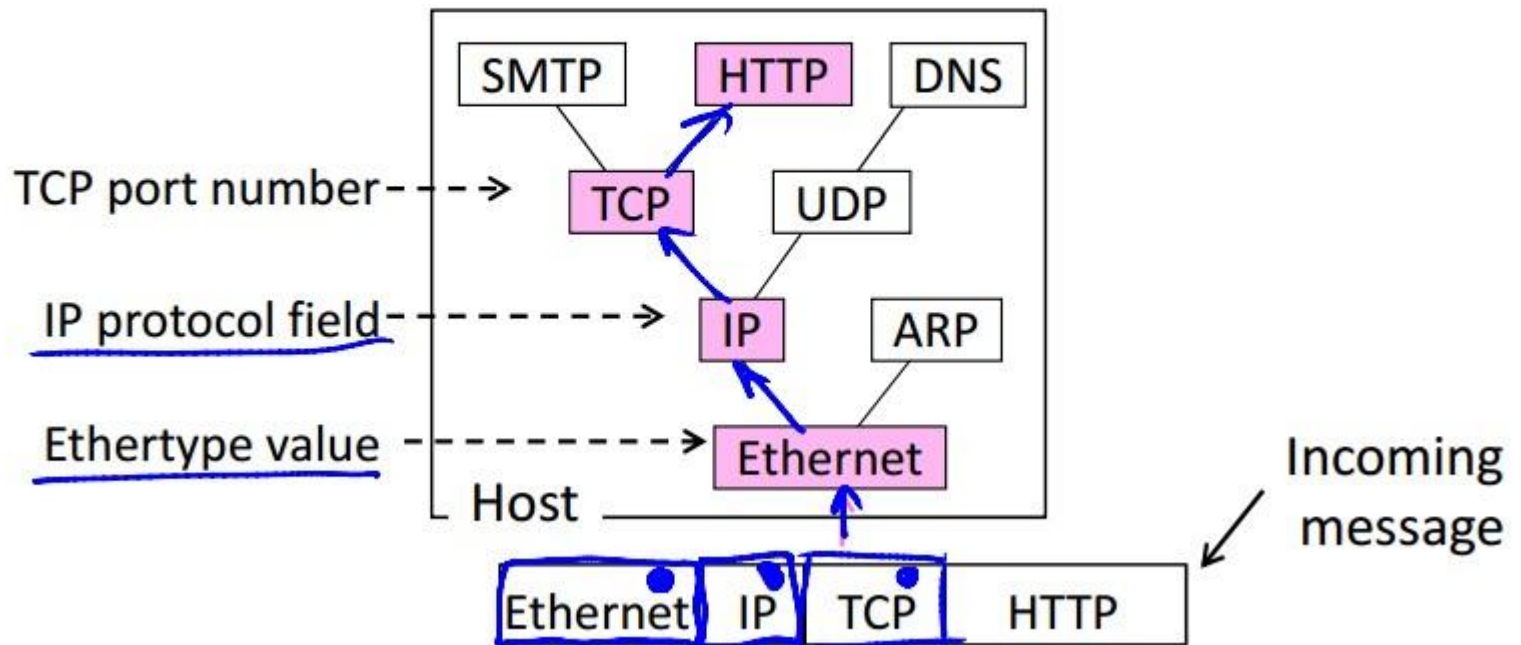
- 从接收端的角度看，任何数据必须正确地交给其原先的高层协议处理
- 由于存在协议复用的情况，如右图所示，当以太网收到一个帧时，它是如何判断是IP数据还是ARP数据？





解复用Demultiplexing

Done with demultiplexing keys in the headers





分裂/整合 Splitting/Recombining

- 分裂/整合与复用解复用相反，它是若干个低层同时为一个高层提供服务
- 应用场景
 - 低层不可靠时
 - 单个低层速度不够时
 - Etc



ISO/OSI参考模型总结

- 协议分层：
 - 分为七层，依次是...
 - 每一层有其约定的功能
 - 从协议栈角度看，协议是水平的（对等层）
- 服务：是低层提供给高层的功能
 - 服务的种类
 - 服务是垂直的，体现为相邻层之间的关系
- 除了最底层之外，其它各层都建立在低层的服务之上。
如HTTP建立在TCP之上，TCP/UDP建立在IP之上等
- 分片/重组等一些列层间关系的概念



OSI参考模型失败的原因

(1) 过于复杂，真正出现的时机不对

- OSI始于1977年，完成于1984年，但直到1989年最终版本仍未完成。而同期TCP/IP协议已经作为开放系统互联的替代方案出现
- 且，TCP/IP协议采取了免费的策略，并开源

(2) 模型过于复杂，影响了效率

- 模型设计时，经验欠缺。很多标准未预先得到有效性和必要性的证明
- 虽然七层模型比较完备，但并不是所有的系统都需要这么齐全的协议分层，从而影响了系统效率



各层对应的专业名词

● For devices in the network:

Layer	Device Name	中文名
Physical Layer	Repeater or Hub	转发器、中继器/ 集线器
Data Link Layer	Switch or bridge	交换器、网桥
Network Layer	Router	路由器*
Other Layer	Gateway	网关*

* 很多场合并不严格区分路由器和网关这两个名词



各层对应的专业名词

- For units of data:

Layer	Unit of Data	中文
Application	Message	消息
Transport	Segment	消息段（或分片）
Network	Packet	分组
Link	Frame	帧
Physical	Bit	比特

有时用数据包统称三层及以上的数据



涉及的标准组织

三大官方国际标准组织

- ITU-T: 国际电信联盟
- ISO: 国际标准组织
- IEC: 国际电工委员会



三大民间国际标准组织

- IETF: 互联网工程任务组
- IEEE: 电气与电子工程师协会 (Ethernet、WLAN)
- 3GPP: 第三代合作伙伴计划 (无线IP)



三大区域性标准组织

- CCSA: 中国通信标准协会
- ETSI: 欧洲电信标准化协会
- ANSI: 美国国家标准局





2.2 TCP/IP体系结构

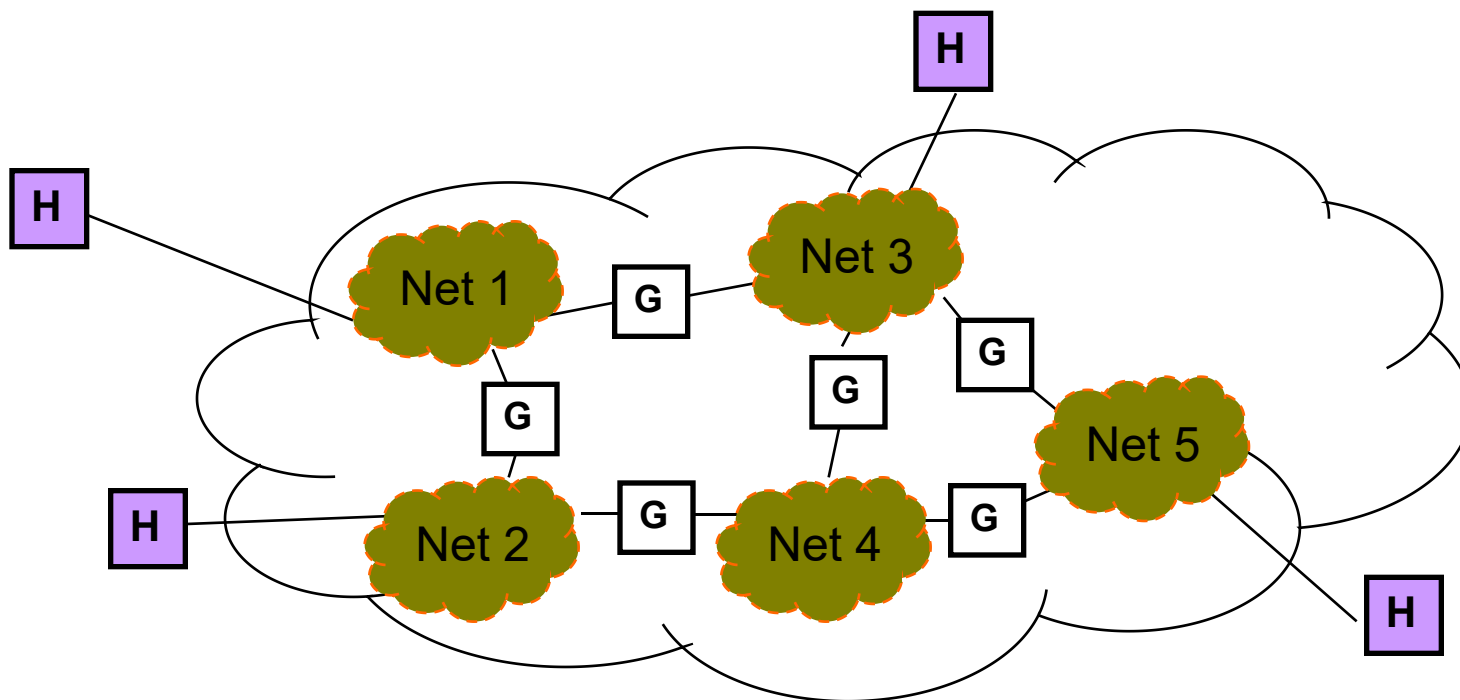
TCP/IP: 是目前网络互联事实上的标准。技术成熟，应用广泛，遍布世界每个角落



网络互联的意义

为了建立“网络的网络”，或者互联网

- 支持在不同的网络之间互联互通
- 通过IP实现无处不在的连接
- 实现经济上的规模效应

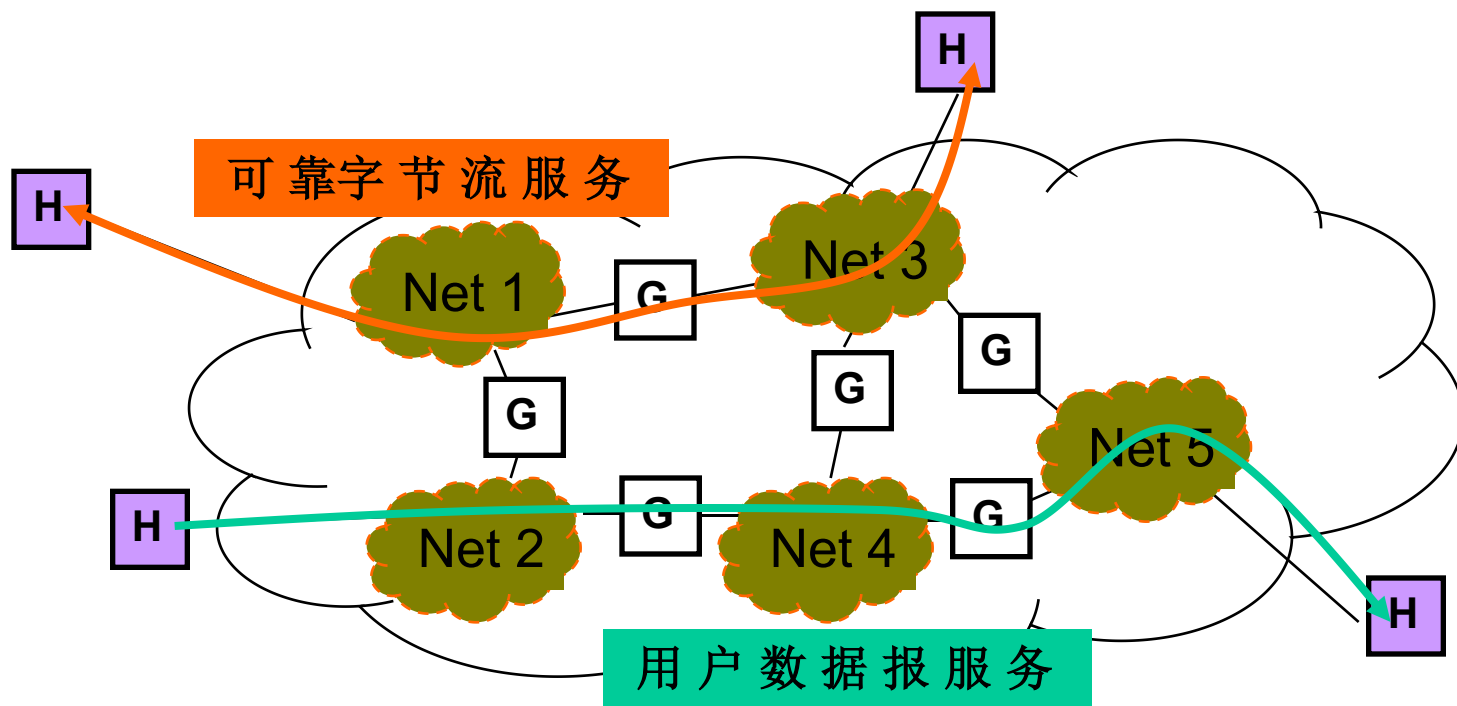




网络互联的意义

提供通用的通信服务

- 独立于底层网络技术
- 对应用程序编写，提供统一的接口





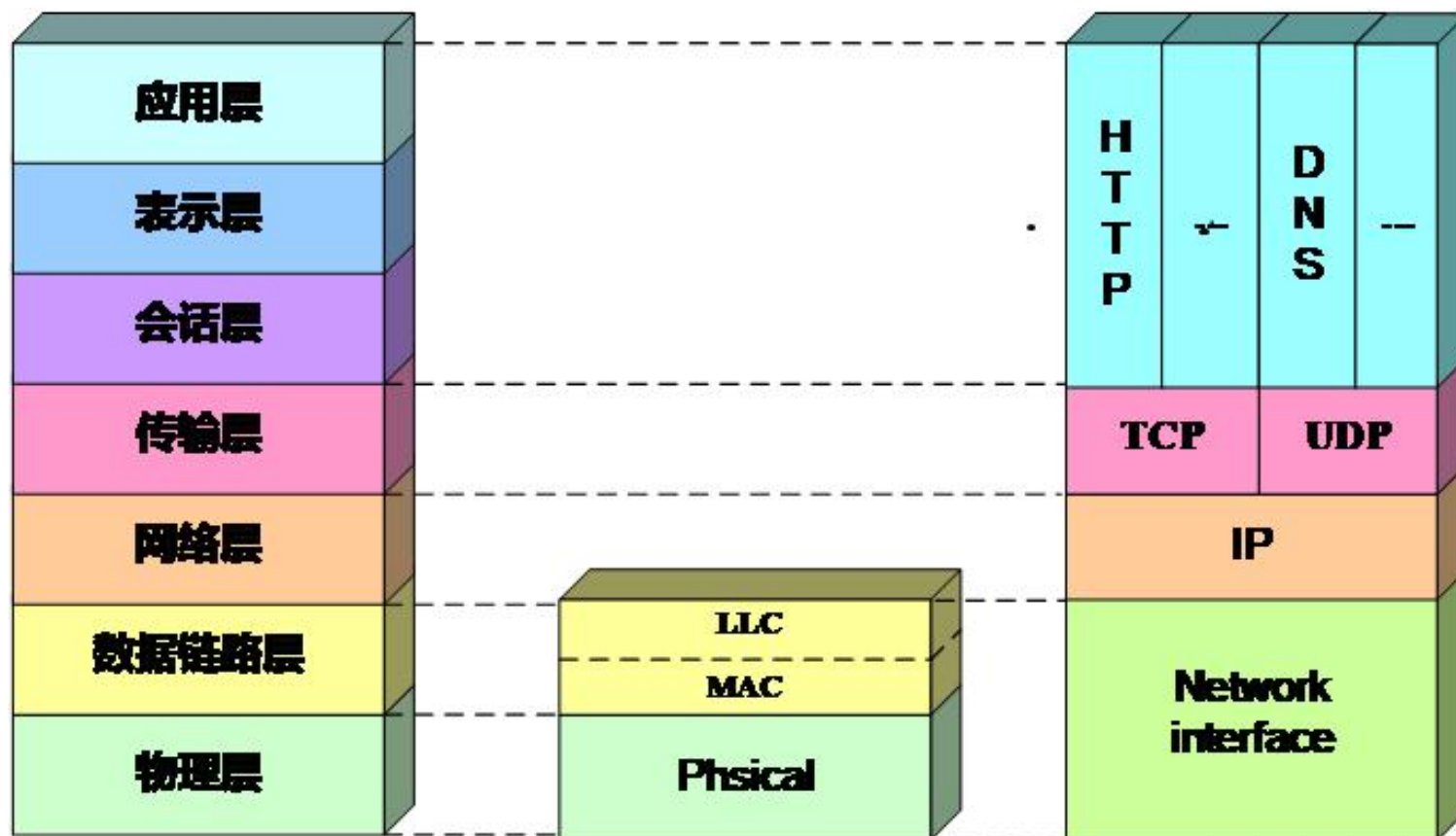
网络互联的意义

提供分布式应用功能

- 任何基于Internet服务标准设计的应用程序，
（无需修改）可在整个互联网上运行
- 新应用可迅速部署
 - Email, WWW, P2P（下载，理财），共享单车
- 应用程序的设计，独立于底层网络技术
 - 底层可引入新技术
 - 过时的网络技术可被淘汰



2.2.1 TCP/IP Architecture



OSI、IEEE802、TCP/IP分层对比

LLC: Logical Link Control, 逻辑链路控制子层



2.2.1 TCP/IP 体系结构

TCP/IP 应用层: 合并了OSI参考模型的高三层

TCP/IP 传输层: 提供两个基本服务类型

TCP—面向连接的、可靠的、字节流服务

UDP—面向无连接的、尽最大努力的服务



2.2.1 TCP/IP 体系结构

TCP/IP 互联网层: 在多个网络中传输信息，与OSI参考模型的网络层对应，提供尽最大努力的服务

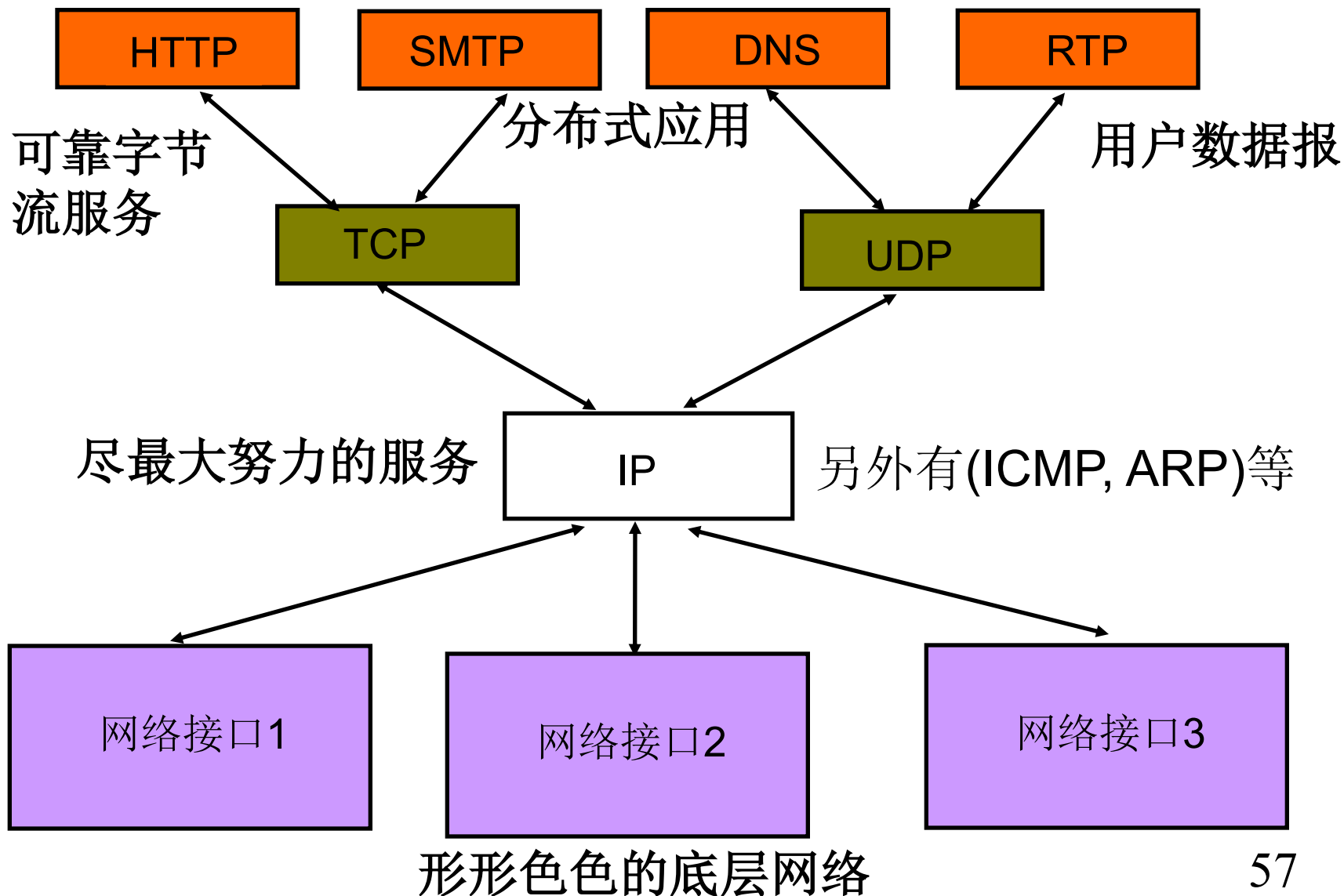
TCP/IP 网络接口层: 泛指提供IP分组传输能力的网络接口

TCP/IP协议栈将其它底层网络归纳为网络接口层，从而将互联网层与底层具体的网络技术分离

TCP/IP协议栈没有严格意义分层，例如BGP的功能提供了网络路由，从功能上看属于网络层；但BGP却运行在TCP之上，从协议栈角度是应用层协议



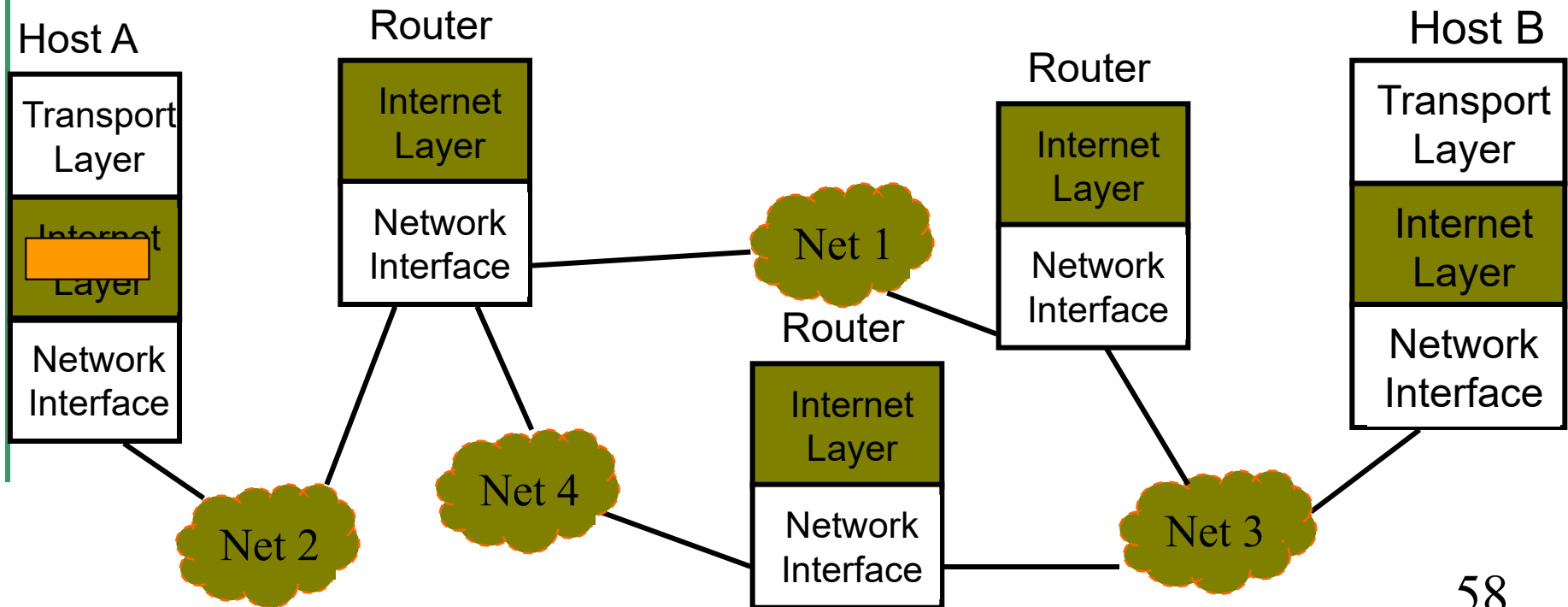
2.2.1 TCP/IP 体系结构





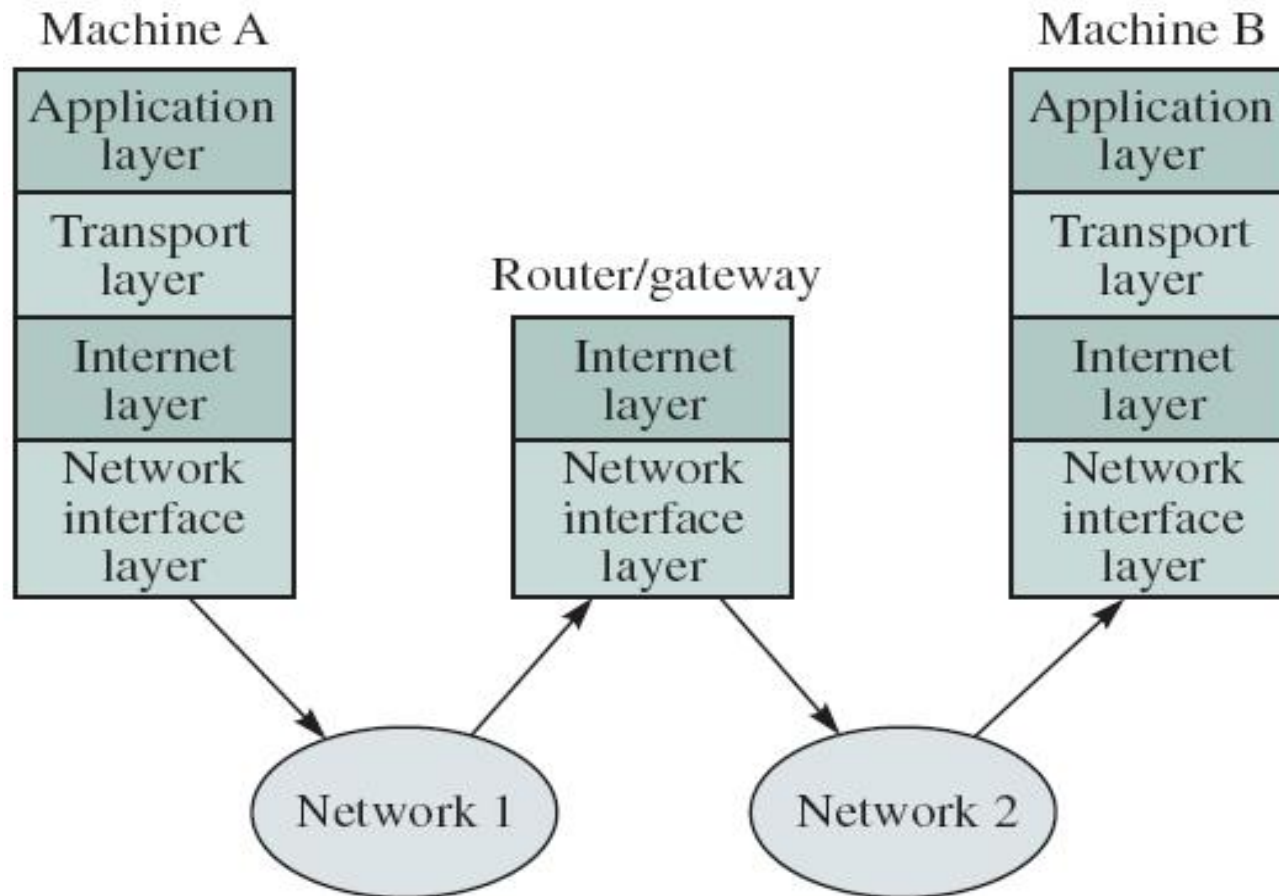
IP协议的工作过程

- IP packets transfer information across Internet
Host A IP → *router* 路由器 → *router*... → *router* → *Host B IP*
- 每个路由器的IP层负责查找路由并选择下一跳节点
- 网络接口层负责将IP分组转发到下一跳节点





互联网层和网络接口层





2.2.2 TCP/IP 协议为例: 各层的相互作用

举例:

- **设备:** 一台服务器, 一台工作站, 一个路由器, 一台位于远端的PC。
 - 虽然从应用的角度, 服务器的重要性要远高于PC或者工作站, 但是从网络的角度看, 它们都是端系统
- **通信链路:** 以太网和点到点 (PPP) 链路各一个



物理地址（MAC地址）

- 以太网允许许多用户接入，它使用了MAC地址区分不同的用户。
- 以太网的MAC地址共48位：
 - 地址是分配给网卡（network interface card，NIC)的
 - 地址在网卡出厂时写入（部分设备可通过软件修改）
 - 网卡的前24比特标记了生产厂家，后24比特对应序号
 - MAC地址的写法，用12个16进制数字，字节之间用冒号或短横分隔。如 00:90:27:96:68:07
 - 为简化描述，后文用小写字母表示MAC地址
 - 以太网的格式中，包括了源MAC地址和目的MAC地址
- 点到点链路，由于链路上“非你即我，我发出去的数据，一定是给你的”，不需要MAC地址区分，因此无须MAC地址



网络层地址（以IPv4地址为例）

- IPV4使用了32位地址，而且是层次编址，地址可以分为两部分：网络部分(netid)+主机部分(hosted)。网络部分又称网络号，是唯一的，IP网络认为网络号相同的终端，在同一个IP网络内，即依据网络号进行路由
- IP地址的表示：每个字节使用十进制数字表示，不同字节之间用点区分。如
byte1.byte2.byte3.byte4
128.100.10.13
- 基于IP的通信，需要源IP和目的IP地址

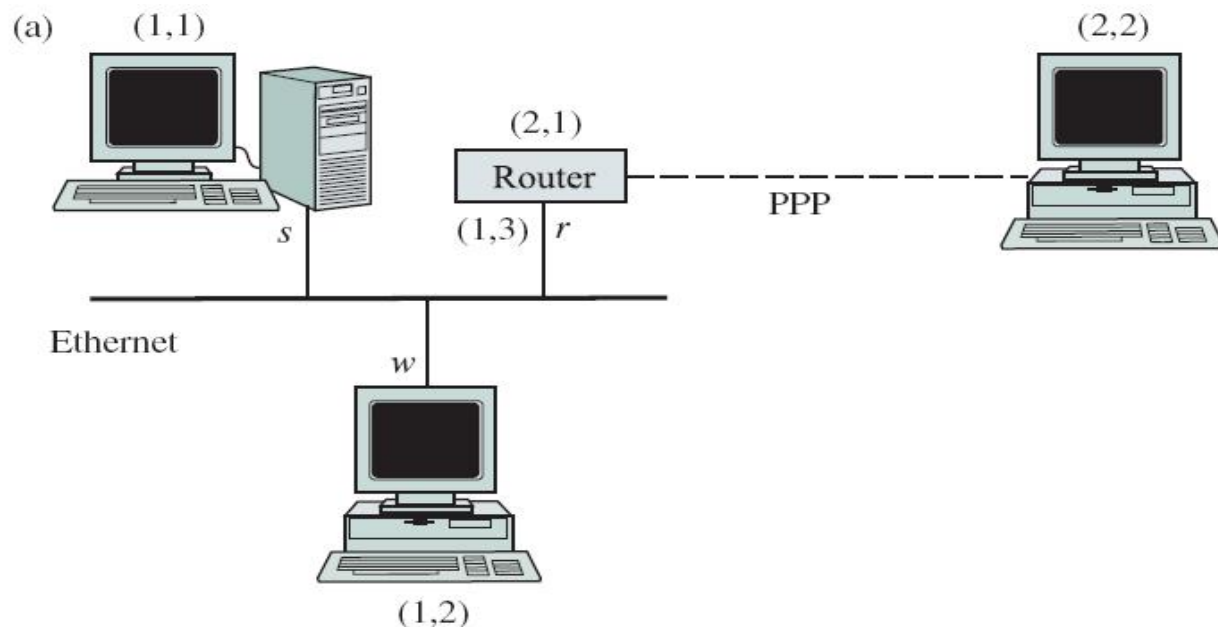


MAC地址 vs IP地址

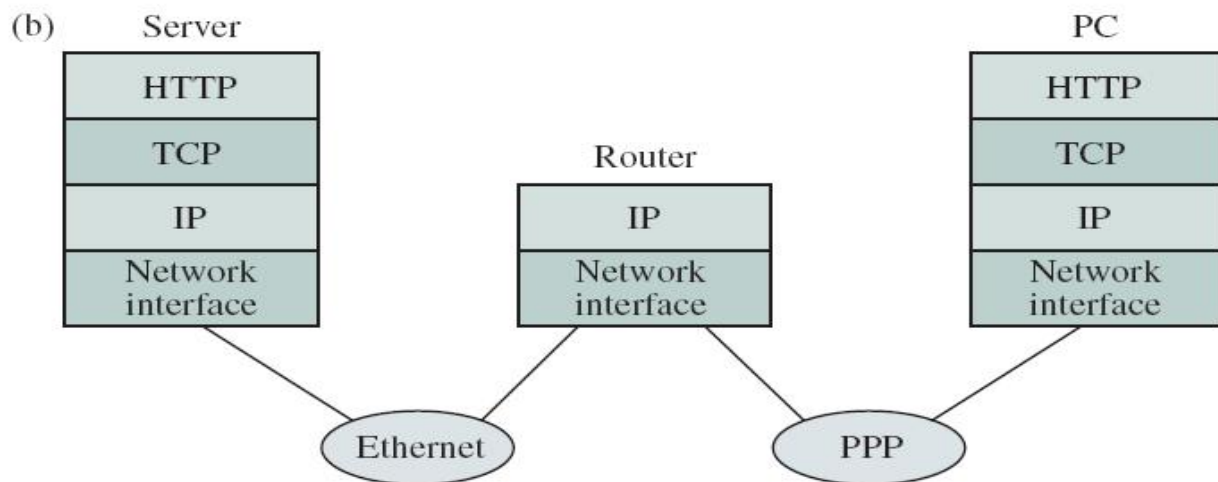
- 为什么不用MAC地址直接通信，还要增加一个IP地址
 - 二层地址格式不统一，也未统一分配
 - 甚至二层没有地址
 - 不同类型的网络，会存在通信障碍
- 若一个以太网内，两个主机需要通过IP地址进行通信
 - 从网络层的角度（三层），源和目的都是IP地址
 - 但此时以太网是它的网络接口层，以太网需要自己和对方的MAC地址
 - 已知对方的3层地址，如何去获取2层地址呢？
 - ARP
- 如果通信的对端不在本局域网呢？甚至对端没有MAC地址呢？



演示网络拓扑和协议栈结构



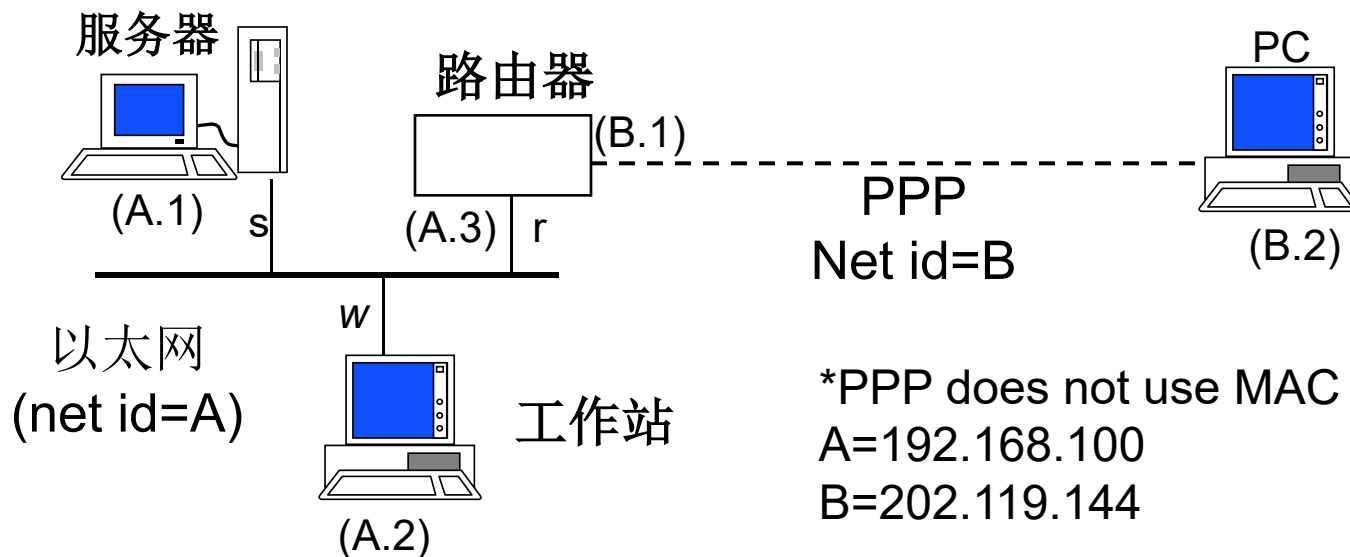
网络拓扑



协议栈



演示网络地址参数



	网络号	主机号	物理地址
服务器	A	1	s
工作站	A	2	w
路由器接口1	A	3	r
路由器接口2	B	1	-
PC	B	2	-

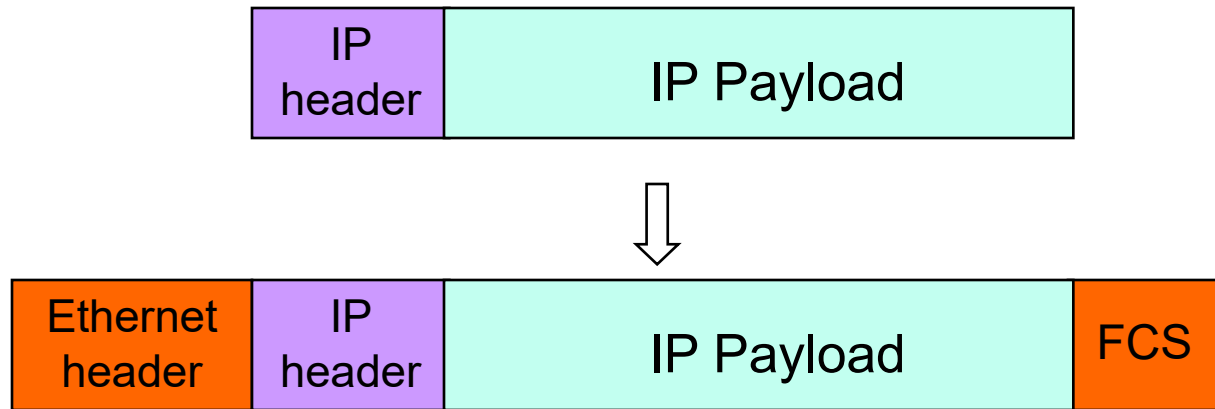


IP地址和MAC地址

- 互联网上IP地址是分配给接口（网卡）的，而不是分配给主机的，如电脑同时接网线和wifi，各有一个地址
 - MAC地址也是分配给接口的，用小写字母r,s,表示
- 上述路由器：有2个接口，分属于2个网段
- LAN的IP 地址: (A.1), (A,2), (A.3)，分别表示192.168.100.1~3
- 点到点链路IP地址: (B,1), (B,2) ，分别表示202.119.144.1~2



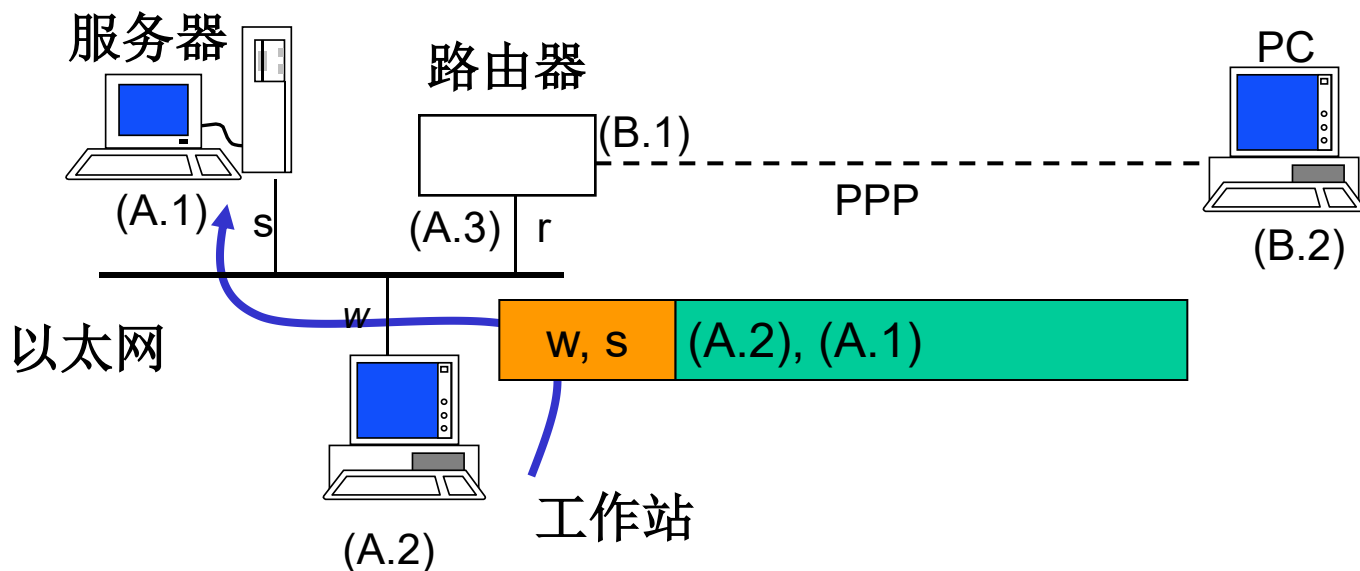
演示网络的封装



- IP 头: 源IP和目的IP地址, 其它字段暂时忽略
- 以太头:
 - 源MAC和目的MAC地址
 - 高层协议字段 (如0x0800表示 IP)



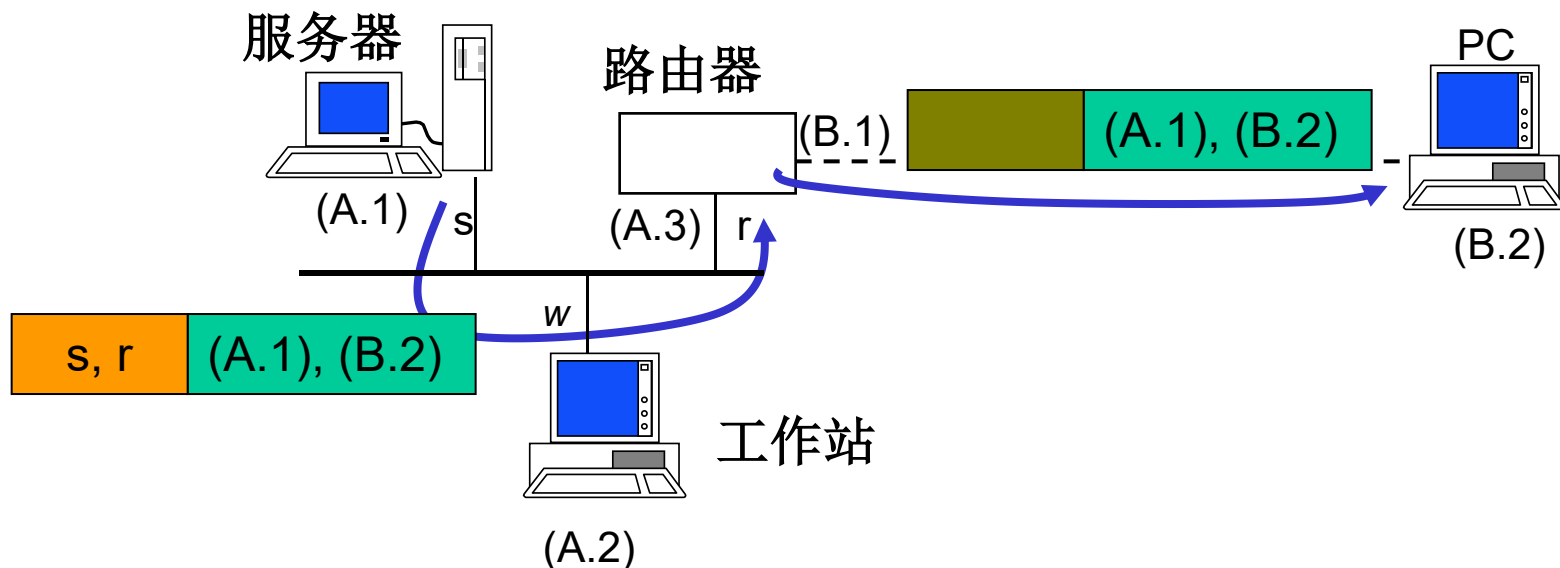
工作站(A.2)发给服务器(A.1)



- 1 该IP分组的源IP地址是A.2，目的地址A.1。A.2发现与A.1属于同一个网段IP（第八章讲解）。同一个网段之间需要直接交付，因此该IP分组封装到以太网帧中，源MAC为w，目的MAC为s，以太帧的类型指明这是一个IP包。
- 2 该以太帧从工作站的网卡发出，被服务器的网卡收取
- 3 服务器网卡收到后，根据以太帧类型，将其中的IP部分（以太帧的SDU）交给IP协议处理，从而完成工作站到服务器的IP分组收发过程



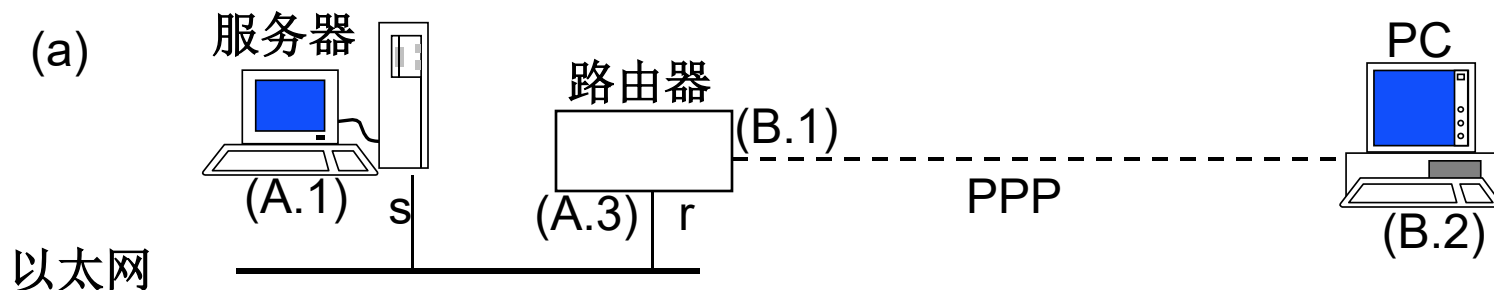
服务器(A.1)发给PC(B.2)



- 1 该IP分组的源IP地址是A.1，目的地址B.2。A.1发现与B.2不属于同一个网段IP ($A \neq B$)。此时该IP分组不能直接到达，服务器应该将该分组交给路由器处理，因此从二层的角度，以太网的目的地址是路由器在LAN上接口的地址（即 r），源地址是服务器自身的MAC地址
- 2 该以太网帧从服务器的网卡发出，被路由器的网卡收取
- 3 路由器网卡收取该以太网帧，由以太网帧类型将包含的IP分组交给IP协议处理
- 4 路由器网络层根据该IP分组的目的地址，查找下一跳路径，确定下一跳是B.2。路由器通过PPP直连B.2，可直接交付。
- 5 IP分组封装在PPP帧中，传输给PC (B.2)
- 6 PC上的PPP协议收到该帧，判断是IP分组，从而交给PC上的IP协议处理。完成从服务器到PC的IP分组收发过程

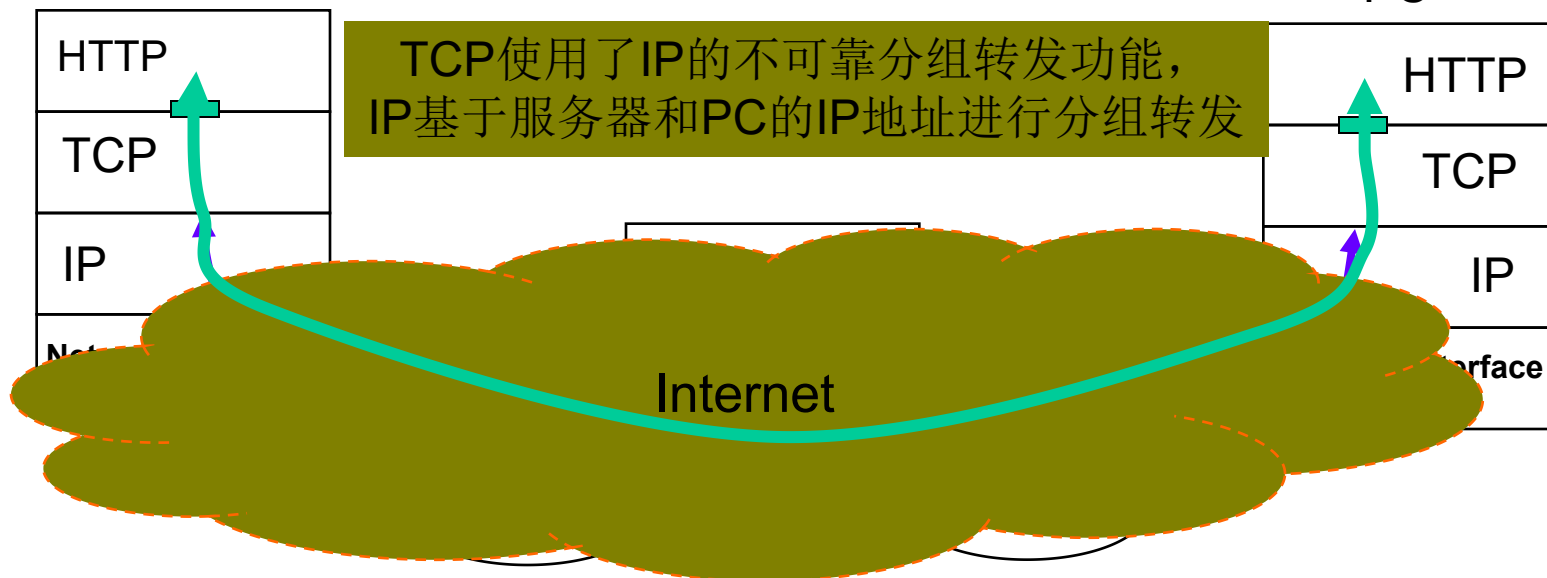


层间的通信协作



HTTP使用了TCP提供的可靠字节流服务，进行进程到进程的通信：
服务器套接字: (IP Address, 80)
PC套接字(IP Address, Eph. #)

(b) 服务器





进程间通信

- **套接字**: 套接字是应用程序与协议栈交互的手段，协议栈提供一系列函数调用（SAP点）。创建套接字时，会关联本地IP地址、协议类型（TCP/UDP），端口号。套接字与创建它的进程（程序）关联。
- 源套接字和目的套接字的组合，唯一确定了端系统上的两个进程

例如:

- 源套接字: (TCP, (B.2), C) //TCP是协议类型，C表示某端口
- 目的套接字: (TCP, (A.1), 80)
- 组合: (TCP, (B.2), C, (A.1), 80) ， 又称五元组



封装

TCP 头包含:

源端口和目的端口等

IP头包含源IP目的IP地址, 高层协议标识 (TCP)

以太头包含了源MAC和目的MAC地址; 高层协议标识 (IP)

HTTP Request



TCP header

HTTP Request



IP header

TCP header

HTTP Request



Ethernet header

IP header

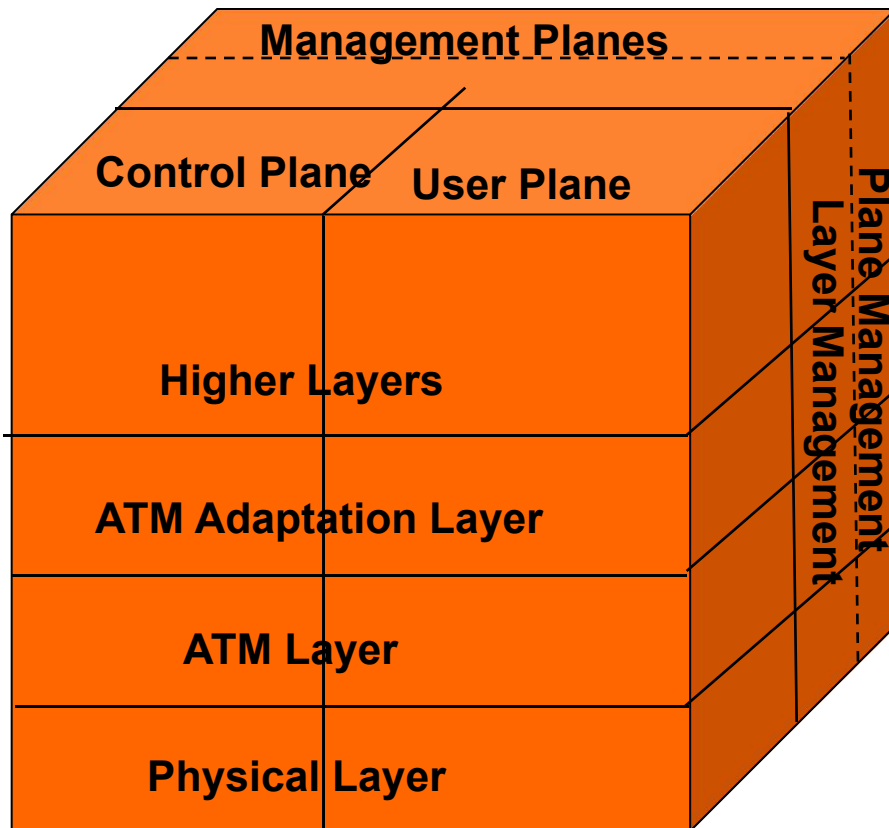
TCP header

HTTP Request

FCS



2.3 ATM Architecture



ATM—Asynchronous transfer mode

User Plane: transfer of user information; flow control; error recovery

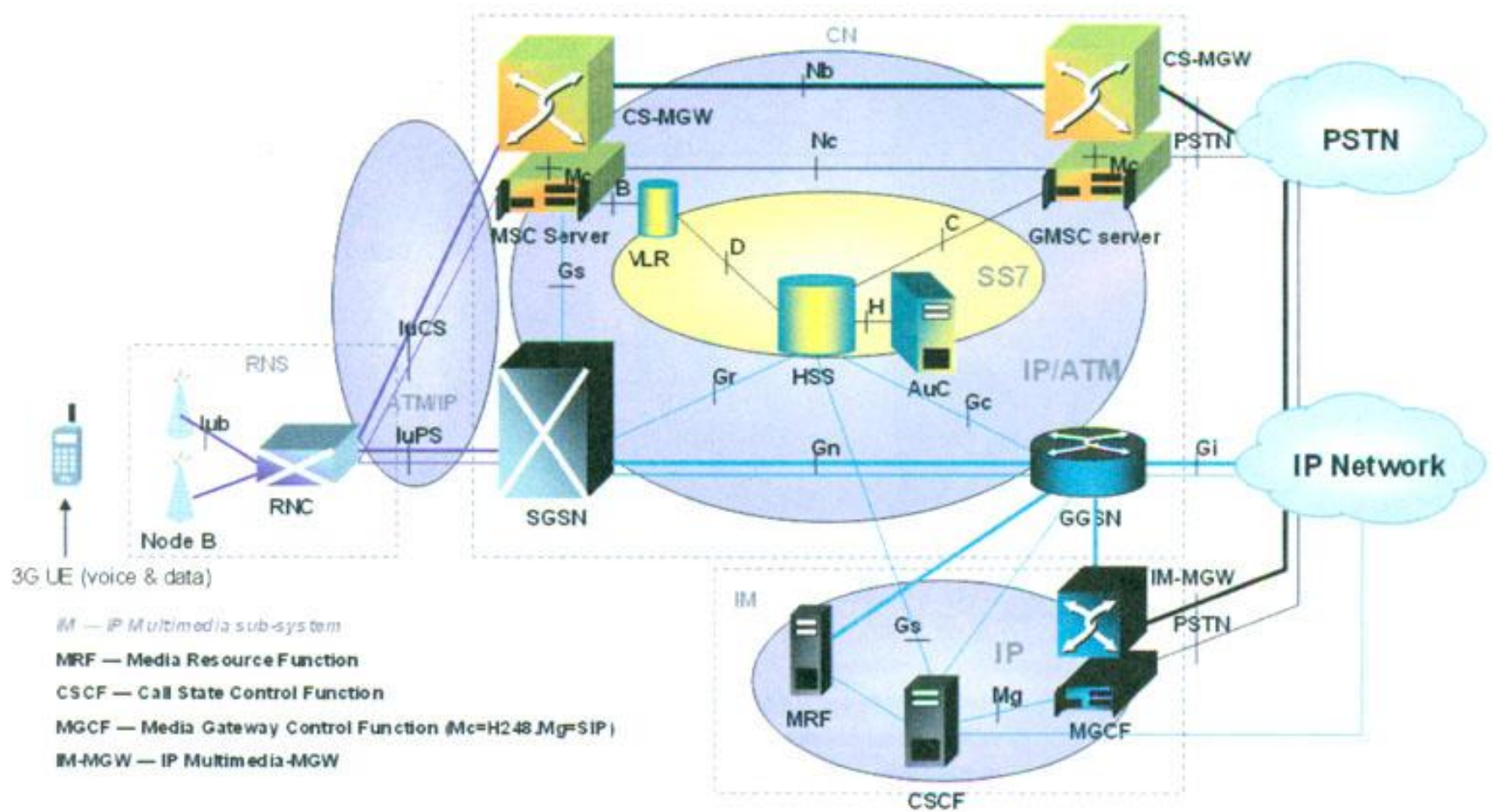
Control Plane: setting up, management, and release of connections

Layer Management Plane: management of layer entities & OAM

Plane Management: management of all the planes



WCDMA structure





Radio Interface protocol architecture

