

# 第二章 计算机网络体系结构

李风华

网络科学与网络空间研究院

# 上堂课小结

- 计算机网络的发展历史
  - 以互联网为例
  - ARPANET
- 基本概念
  - 计算机网络的定义
  - 计算机网络的组成
  - 分组交换的概念
  - 计算机网络的通信方式
  - 分组交换网络的特点

# 主要内容

- 计算机网络体系结构
  - ISO/OSI参考模型
  - TCP/IP参考模型
- TCP/IP网络应用实例



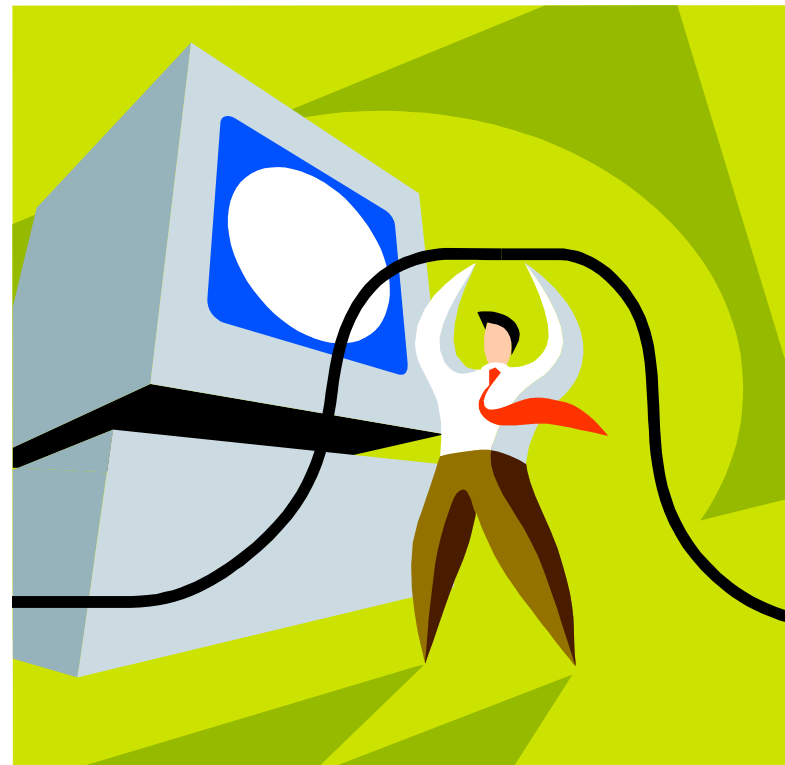
# 计算机网络体系结构

- 计算机网络的基本功能
  - 为地理位置不同的计算机用户之间提供访问通路
- 如何实现?
  - 连接性
  - 可标识
  - 可定位
  - 正确性
  - 其他



# 如何实现计算机之间的信息交互

- 计算机之间信息交互所必需的功能
  - 连接性
    - 连接源结点和目的结点的物理传输线路，可以经过中间结点
    - 每条线路两端的结点利用信号进行二进制通信
    - 多个用户共享一条物理线路
  - 可标识、可定位
    - 地址分配
    - 按照地址信息，进行路由选择
  - 正确性
    - 无差错的信息传送
    - 信息缓冲和流量控制
  - 其他
    - 会话控制
    - 满足各种用户、各种应用的访问要求
- 计算机网络的基本功能！



# 计算机网络体系结构

- 计算机网络体系结构
  - 对计算机网络及其部件所完成功能的比较精确的定义
  - 从功能的角度描述计算机网络的结构
  - 是层次和层间关系的集合
- 计算机网络体系结构仅仅定义了网络及其部件通过协议应完成的功能；不定义协议的实现细节和各层协议之间的接口关系。

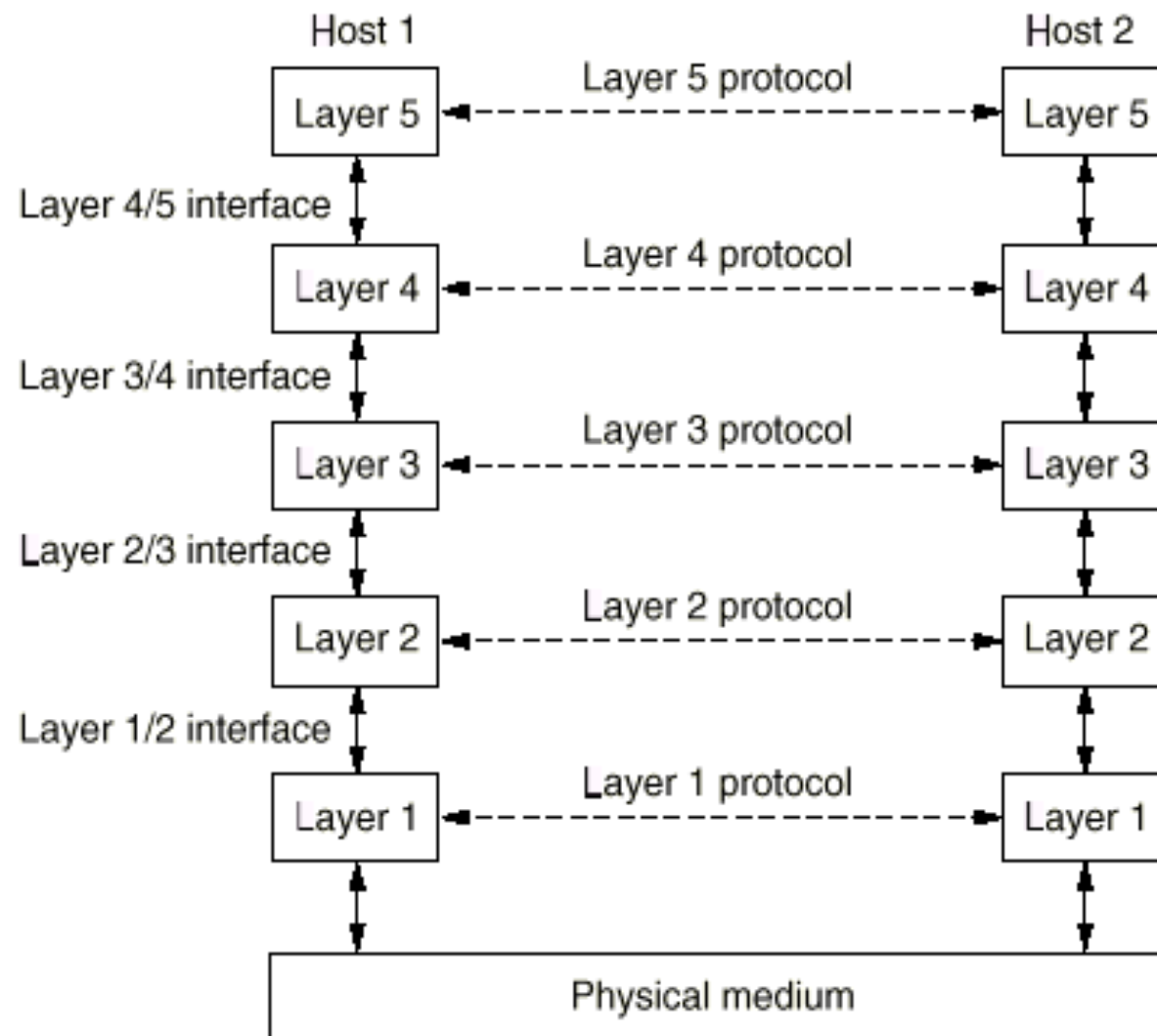
# 计算机网络体系结构（续）

- 计算机网络体系结构特点
  - 通信的计算机之间同时满足功能要求
  - 通信是双向的，通信功能是相互的
  - 这些功能分布在计算机（用户设备）和网络交换节点（网络设备）中
- 一般人们采用“层次结构”的方法来描述计算机网络：
  - 计算机网络中提供的功能是分成层次的

# 计算机网络体系结构（续）

- 计算机网络体系结构：协议与服务
  - 同层之间（层内功能）：协议
  - 不同层次之间（层间关系）：服务
- 协议
  - 计算机网络同等层次之间（对等实体），通信双方进行信息交换时必须遵守的规则
- 服务
  - 层间交换信息时必须遵守的规则
    - 上层：服务用户
    - 下层：服务提供者





**Fig. 1-9.** Layers, protocols, and interfaces.

# 计算机网络体系结构（续）

- 服务访问点SAP（Service Access Point）
  - 任何层间服务都是在接口上的SAP进行的
  - 每个SAP有唯一的识别地址
  - 每个层间接口可以有多个SAP
- 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

# 计算机网络体系结构（续）

- 协议数据单元PDU（Protocol Data Unit）
  - 第N层实体通过网络传送给它的对等实体的信息单元
  - 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
  - 分段和重组

# 计算机网络体系结构（续）

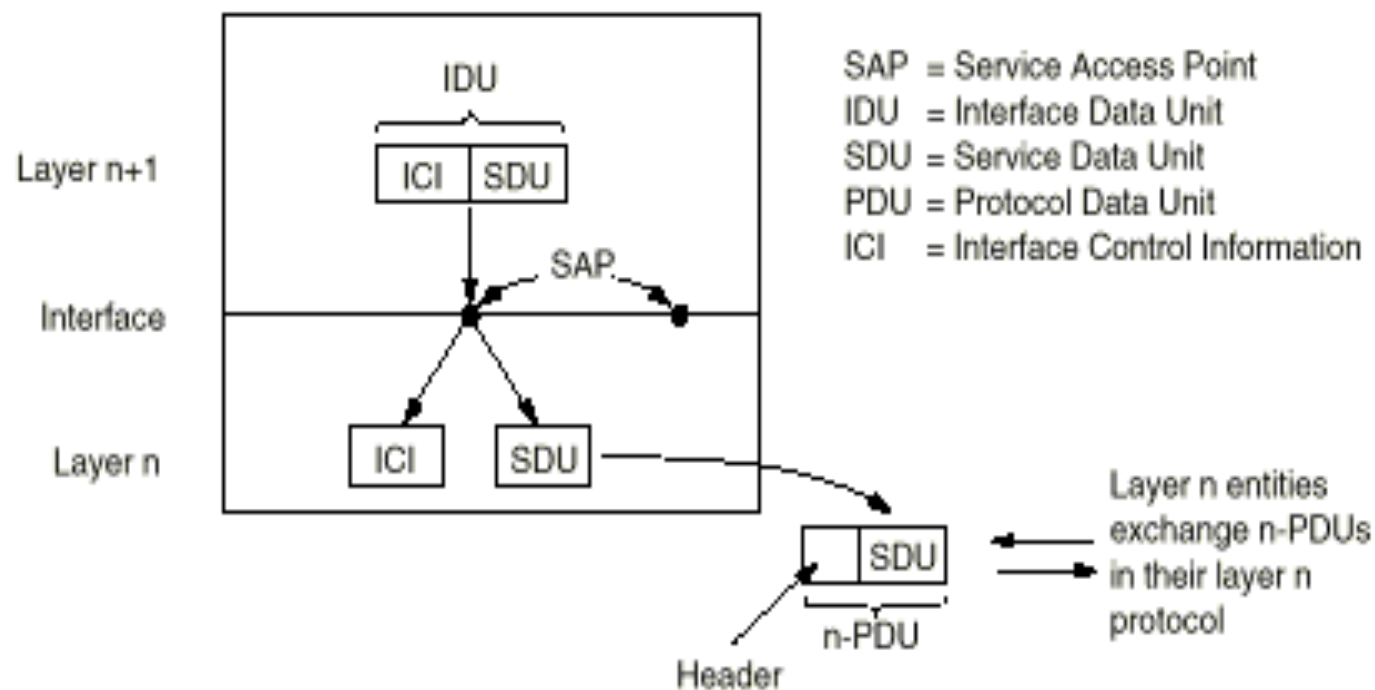
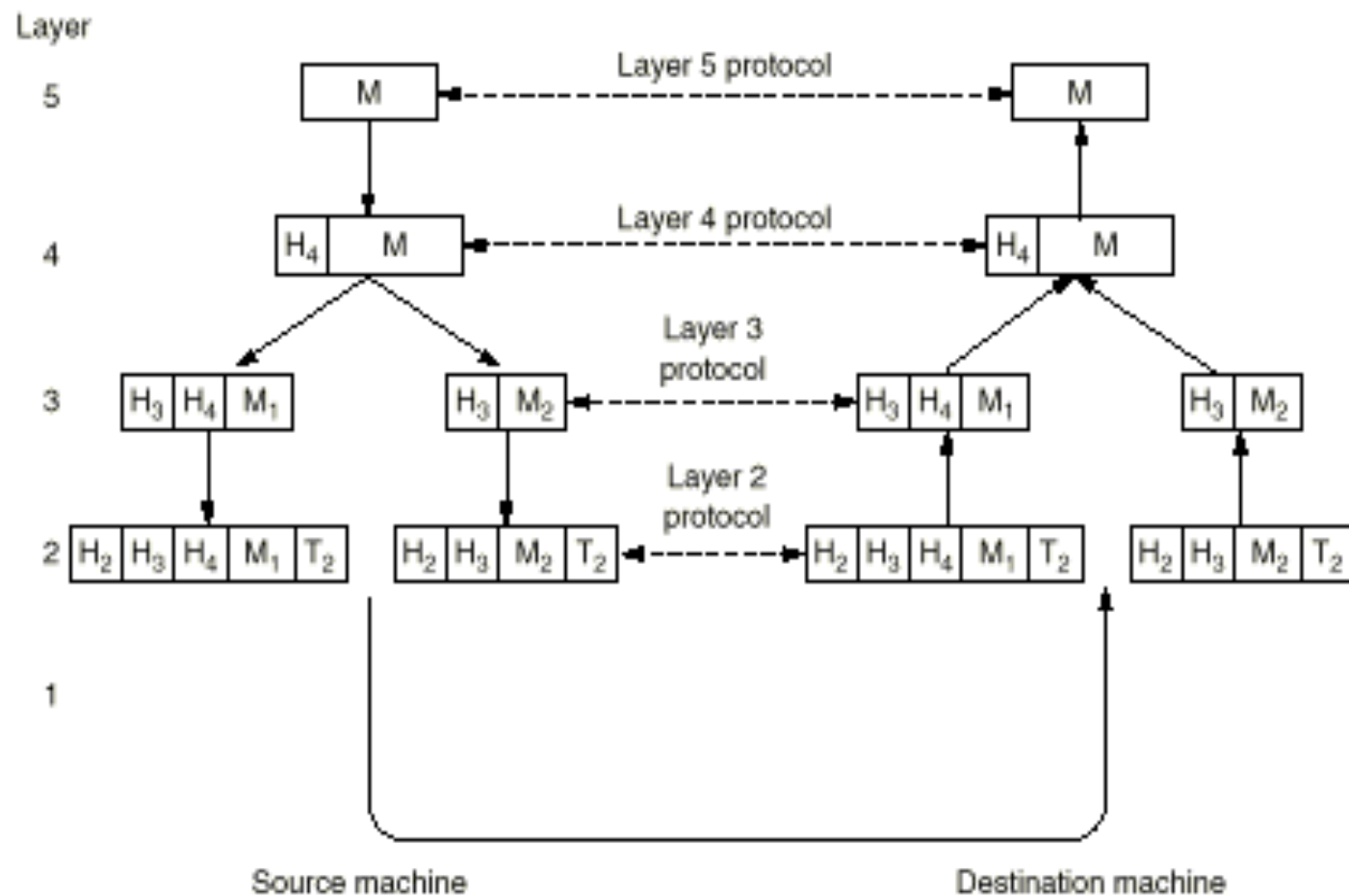


Fig. 1-12. Relation between layers at an interface.



**Fig. 1-11.** Example information flow supporting virtual communication in layer 5.

# 计算机网络体系结构（续）

- 服务分类和服务原语（Primitives）

- 基于连接的服务和无连接服务

- 基于连接的服务

- 当使用服务传送数据时，首先建立连接，然后使用该连接进行数据传送
      - 传送结束后，关闭连接
      - 特点：顺序性好

- 无连接服务

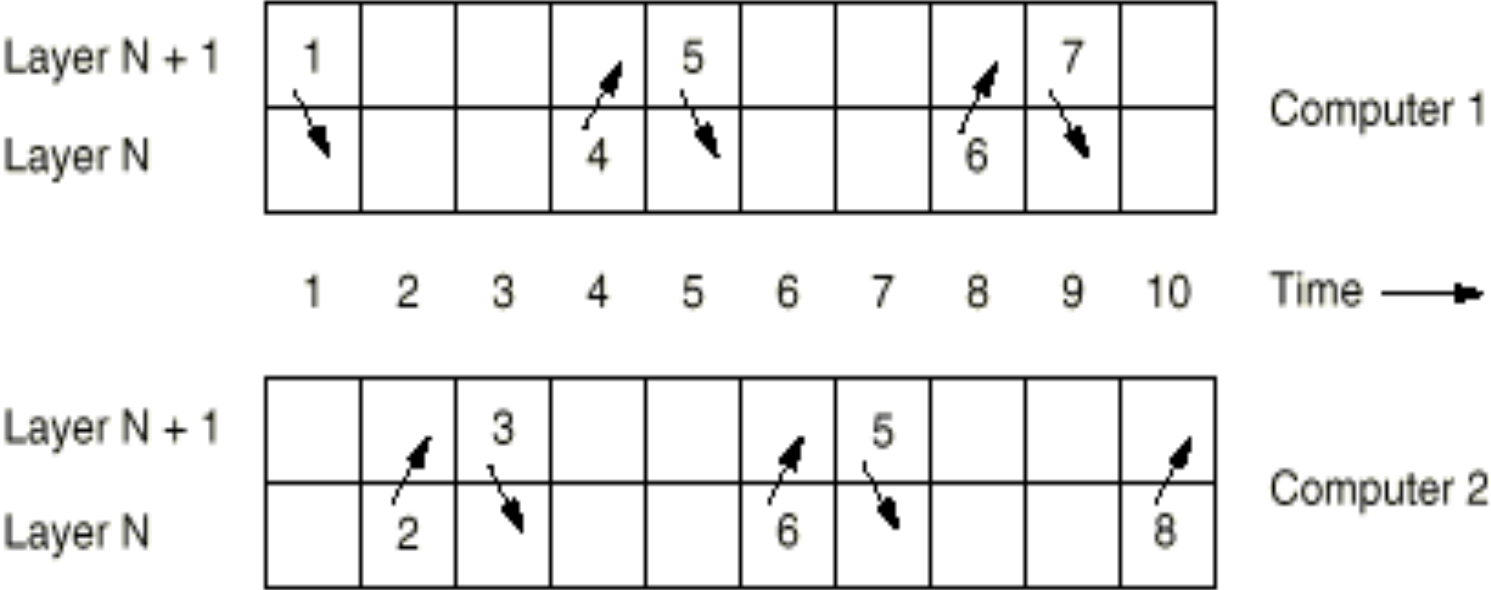
- 直接使用服务传送数据，每个分组独立进行路由选择
      - 特点：顺序性差

- 面向连接不等于传输的可靠性，可靠性要通过确认、重传等机制保证

# 计算机网络体系结构（续）

- 服务原语
  - 服务在形式上是由一组接口原语（操作或者命令）来描述的
    - 原语可以携带参数
    - 如目的主机地址、服务类型、最大分组长度...
  - 服务原语可分为四种类型
    - 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）：一个实体希望响应一个事件
    - 确认（Confirm）：返回对先前请求的响应
- 服务
  - 有证实
  - 无证实

# 计算机网络体系结构（续）

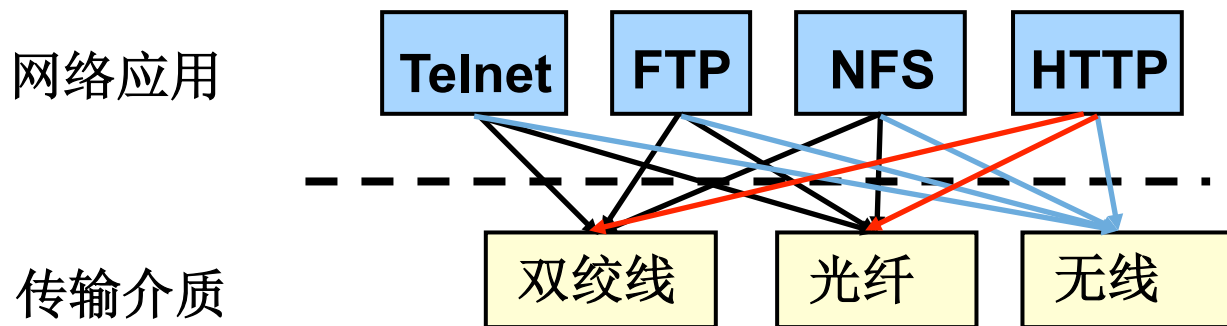


**Fig. 1-15.** How a computer would invite its Aunt Millie to tea. The numbers near the tail end of each arrow refer to the eight service primitives discussed in this section.



# 计算机网络体系结构（续）

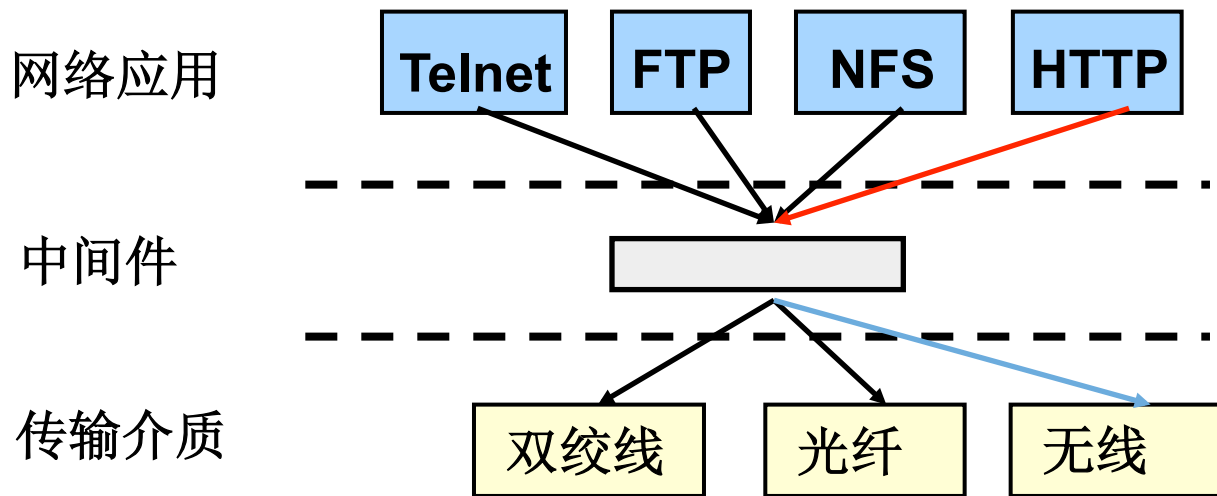
- 为什么要分层?



- 如果没有分层：不同的应用在不同的传输介质上都要开发独立的程序

# 计算机网络体系结构（续）

- 解决办法：开发一个中间件，屏蔽所有不同网络介质的特点，给网络应用提供一个独立的接口方式



# 计算机网络体系结构（续）

- 分层的优点：

- 模块化：易于开发、管理和维护
  - 降低复杂性
- 独立性：层次内部的变化不会影响其他层次
  - 易于技术进步
- 复用性：高层模块可以复用底层模块的功能
- 标准化：为不同厂家的设备互联提供条件
  - 提高互操作性

- 分层的缺点：

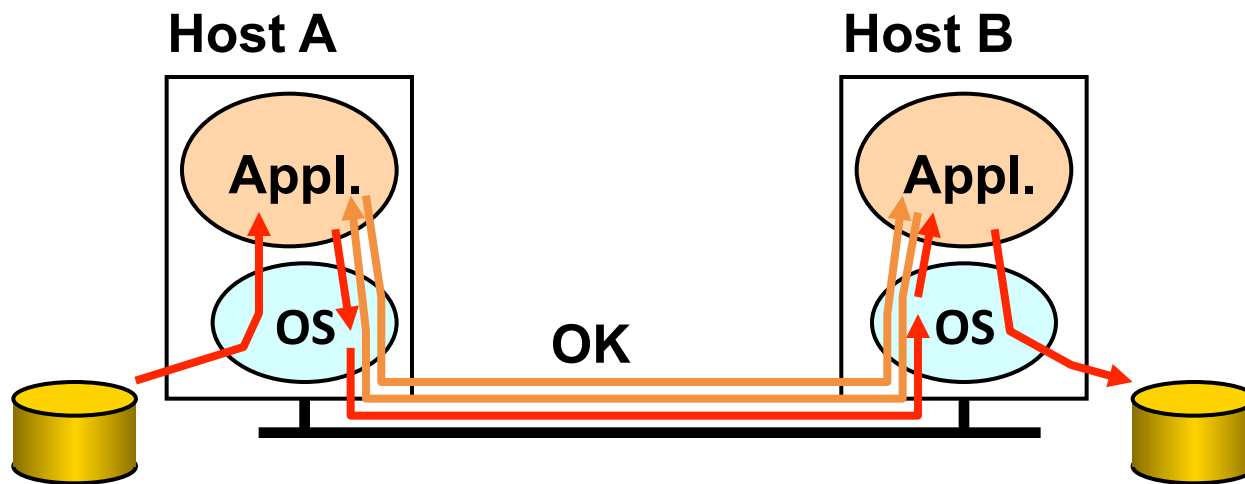
- 低效：信息屏蔽导致性能优化困难

# 计算机网络体系结构（续）

- 端到端问题
  - 功能该如何划分?
    - 高层和底层都可以实现的功能
      - 以高层实现为主
      - 实现在底层的功能应保证尽可能小的影响那些不使用该功能的高层功能的性能
      - 提高性能的功能以底层实现为主

# 计算机网络体系结构（续）

- 实例：可靠的文件传输服务



- 方案1：路径上每一段都保证传输的正确性
- 方案2：端到端检查传输的正确性

# 计算机网络体系结构（续）

- 接收端必须检查传输正确性
- 因此只需要端到端进行正确性检查（高层功能），没有必要在中间节点（底层功能）进行传输正确性检查
- 底层没有必要实现可靠传输？
  - 在提高性能的情况下：有必要！！！！
  - 高差错率的传输线路

# ISO/OSI参考模型

- 计算机网络的标准化
  - 电信标准
    - 国际电信联盟ITU (International Telecommunication Union)
      - 1865年成立
      - 1947年成为联合国的一个部门
      - ITU-R: 无线标准
      - ITU-T: 电信标准, 1956-1993年称为CCITT, 下设多个研究组SG, 研究组下设专题, 如: Q42/SG VII专门研究OSI参考模型
      - ITU-D: 电信发展部门

# ISO/OSI参考模型（续）

- 国际标准

- 国际标准化组织ISO（International Standards Organization）
  - 1946年成立
  - 89个成员国家，85个其他成员
  - 200多个技术委员会TC，下设分委员会SC，每个分委员会会有多个工作组WG组成
  - TC97：计算机和信息处理
  - 国际标准形成过程
    - 委员会草案CD（Committee Draft）
    - 国际标准草案DIS（Draft International Standard）
    - 国际标准IS（International Standard）
  - ISO和ITU之间有很紧密的合作关系



# ISO/OSI参考模型（续）

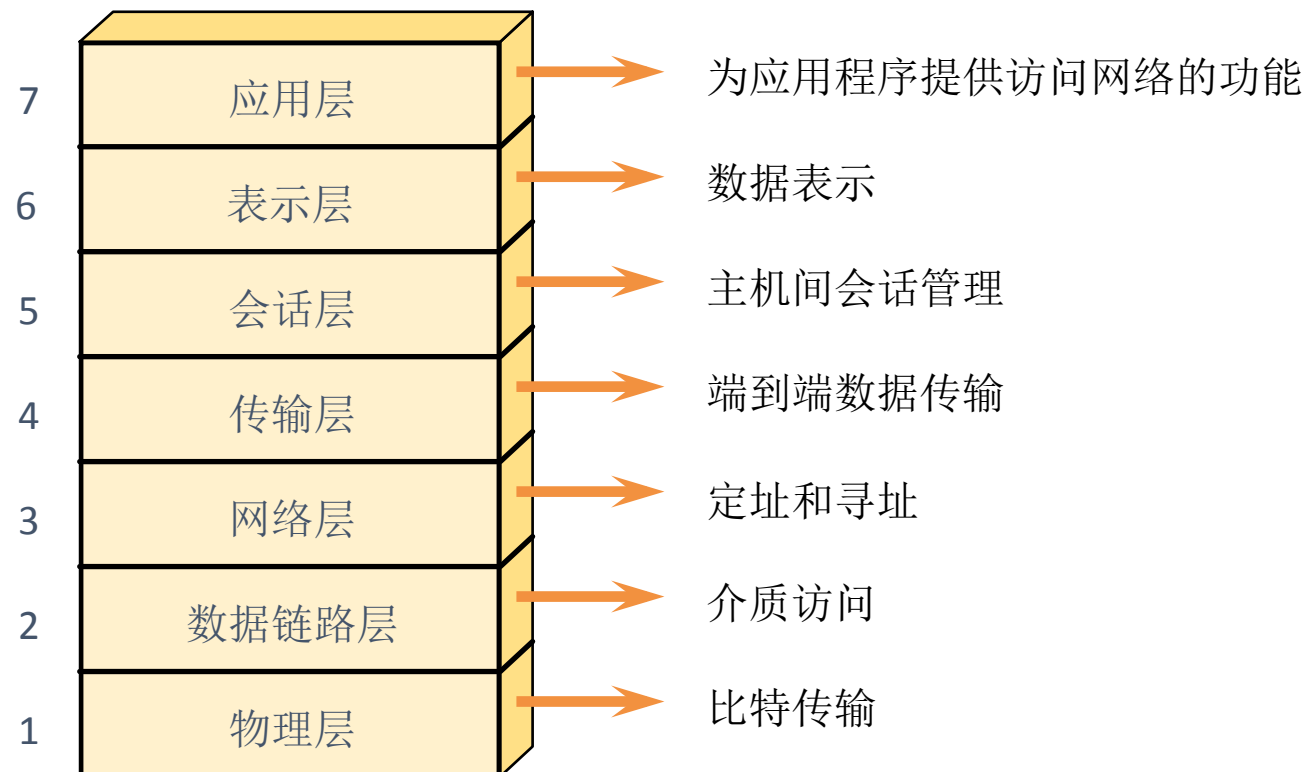
- 其他标准化组织
  - ANSI: 美国国家标准协会，ISO的美国代表
  - NIST: 美国国家标准和技术协会，美国商业部的标准化组织
  - IEEE: 电子和电器工程师协会，发布行业标准
    - IEEE 802委员会专注于局域网标准
  - CCSA: 中国通信标准化协会，发布行业标准

# ISO/OSI参考模型（续）

- Internet标准
  - Internet的标准是自发的，不是政府干预的
  - 称为RFC（Request for Comment）
  - 1969年产生第一个RFC，目前已达到8470个
  - <http://www.ietf.org/rfc>
  - 1983年成立IAB（Internet Architecture Board）
  - 1986年在IAB下成立IETF（Internet Engineering Task Force）
  - 1989年成立IRTF（Internet Research Task Force）

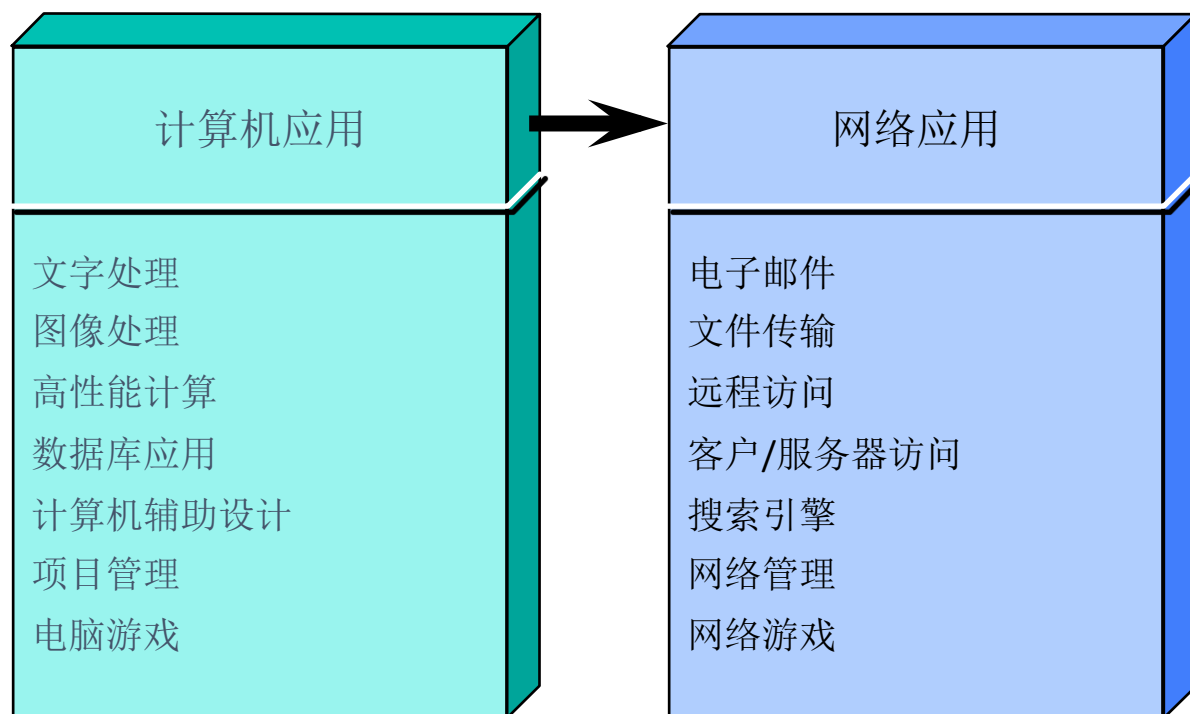
# ISO/OSI参考模型（续）

- ISO/OSI（Open System Interconnection）参考模型（1983年）



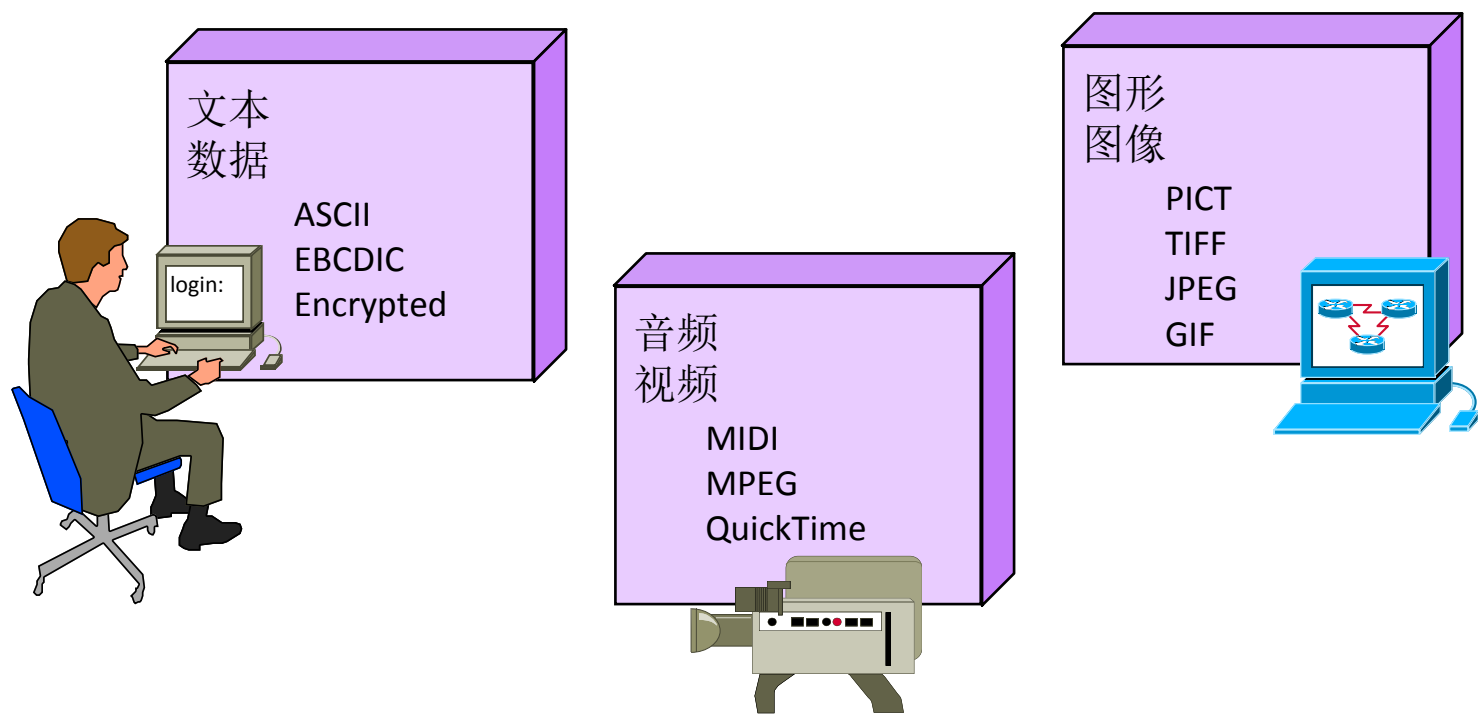
# ISO/OSI参考模型（续）

- 应用层
  - 为应用提供网络接口



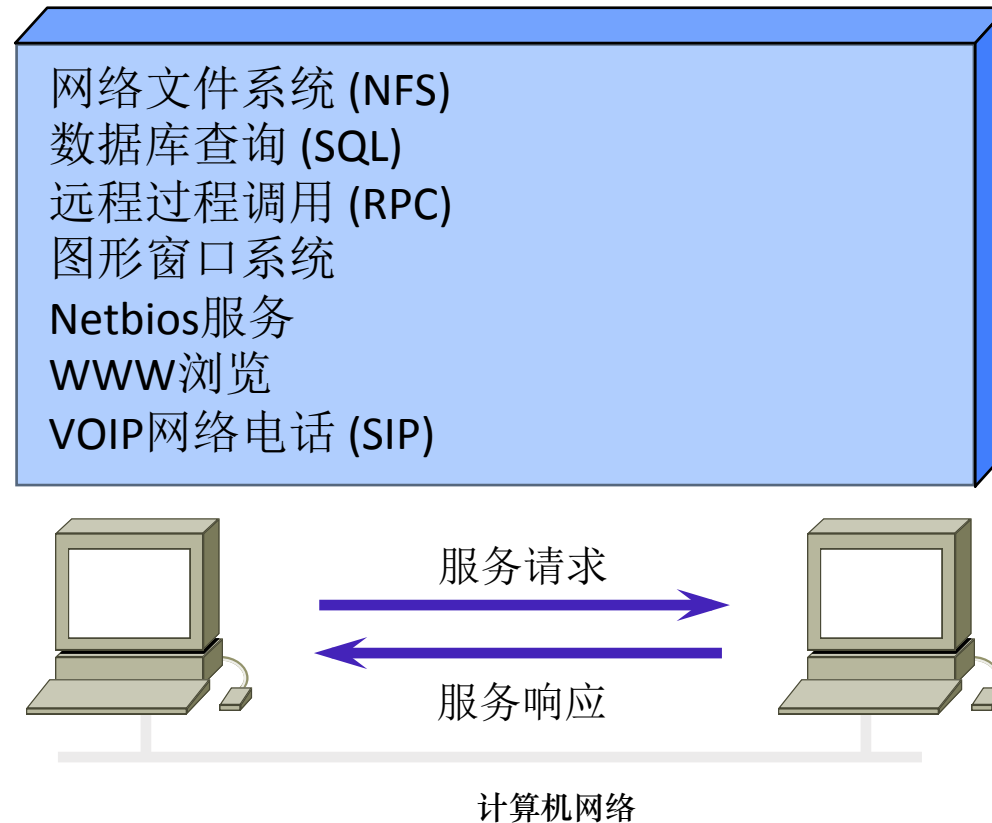
# ISO/OSI参考模型（续）

- 表示层
  - 为应用数据提供编码格式



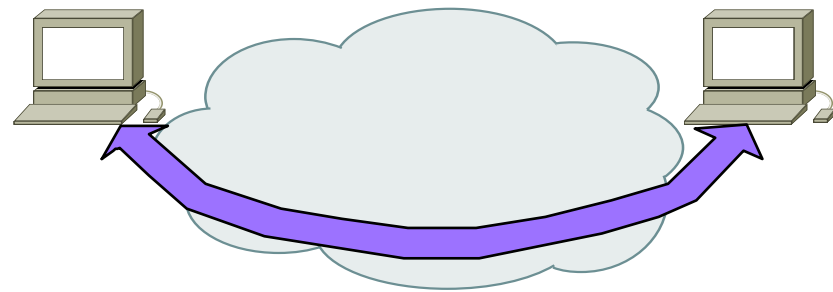
# ISO/OSI参考模型（续）

- 会话层
  - 实现主机间的会话管理



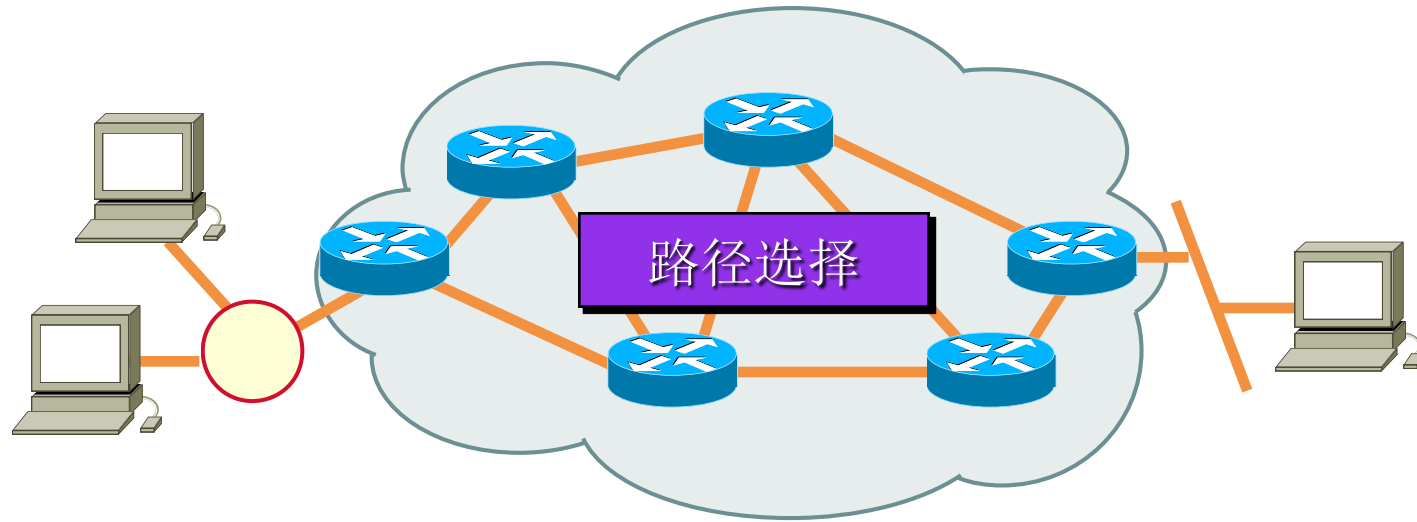
# ISO/OSI参考模型（续）

- 传输层
  - 实现主机间端到端的数据传输
    - 对上层数据进行分段和重组
    - 连接管理（面向连接）
    - 数据传输
    - 传输可靠性保证（可选）
    - 流量控制和速率匹配



# ISO/OSI参考模型（续）

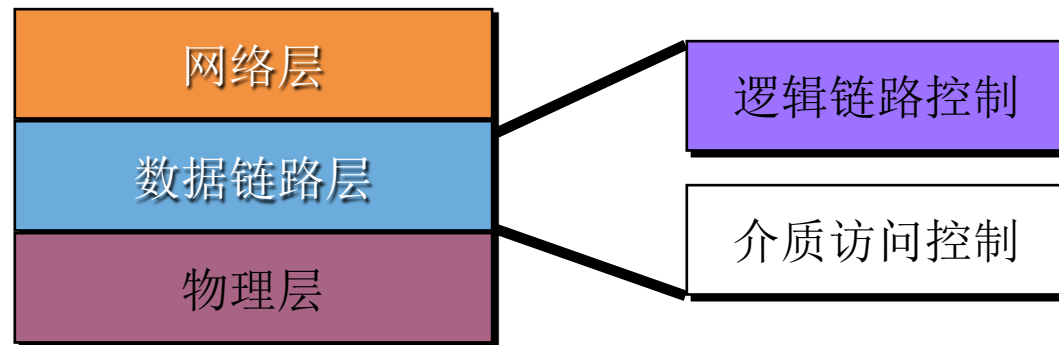
- 网络层
  - 定址和寻址





# ISO/OSI参考模型（续）

- 数据链路层
  - 使用介质/媒体（Media）进行数据传输
    - 逻辑链路控制LLC（Logical Link Control）
      - 软件功能为主：通过SAP向网络层提供服务
    - 介质访问控制MAC（Media Access Control）
      - 硬件功能为主：使用介质传输数据帧

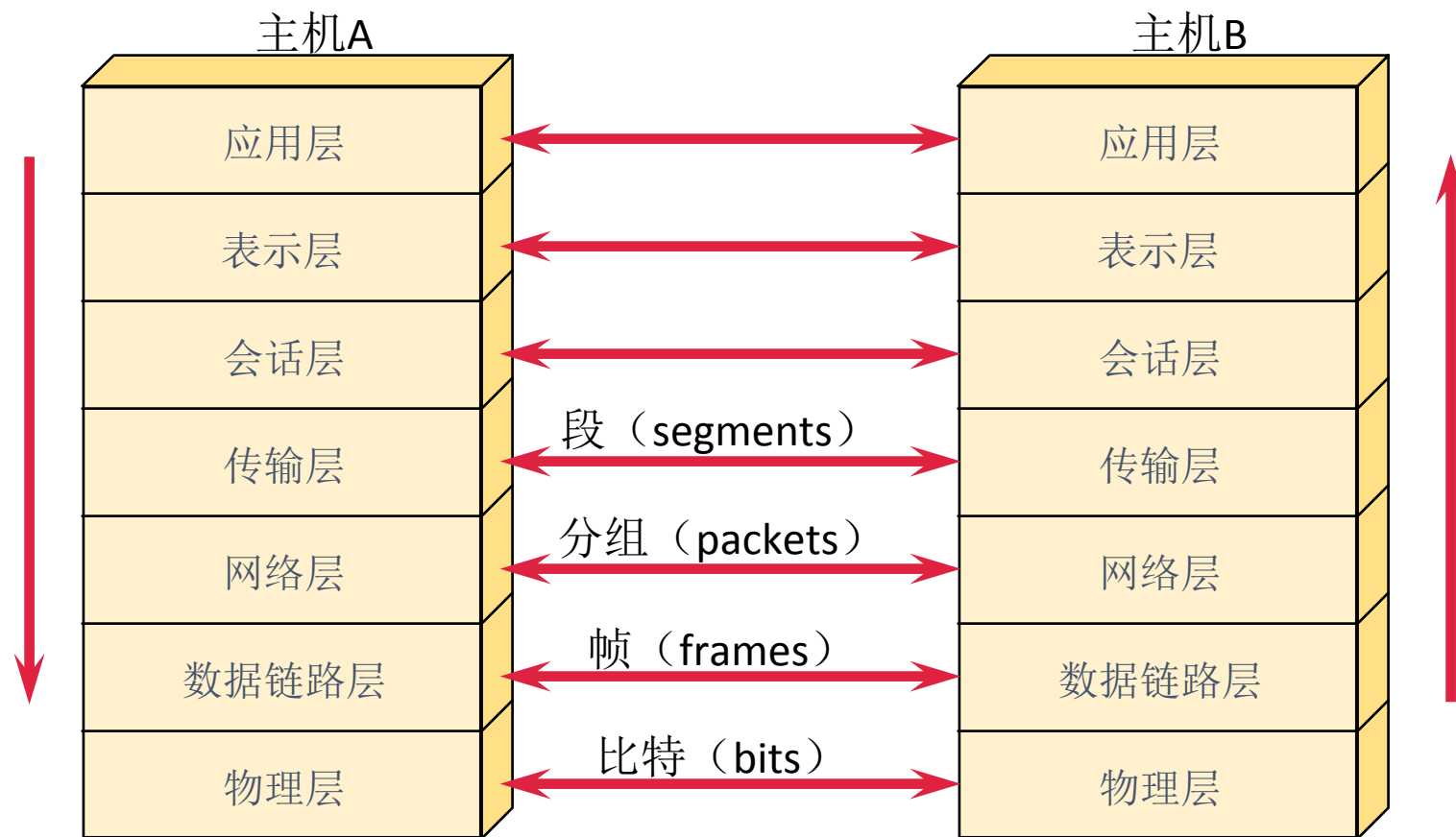


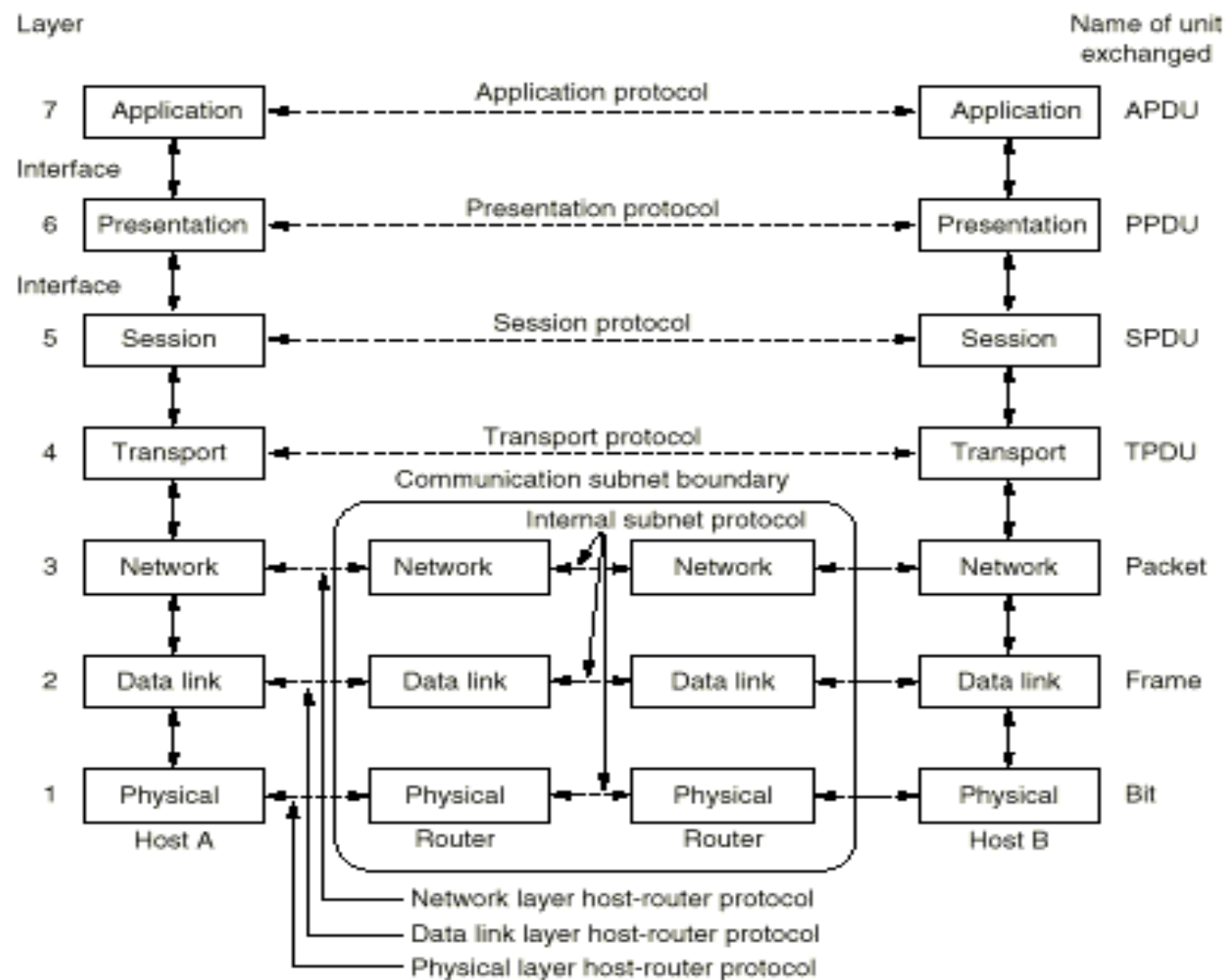
# ISO/OSI参考模型（续）

- 物理层
  - 进行比特传输
    - 定义电气特性和物理特性
    - 电压、电流、阻抗
    - 接口物理尺寸
    - 线序
    - ...

# ISO/OSI参考模型（续）

- 对等进程通信

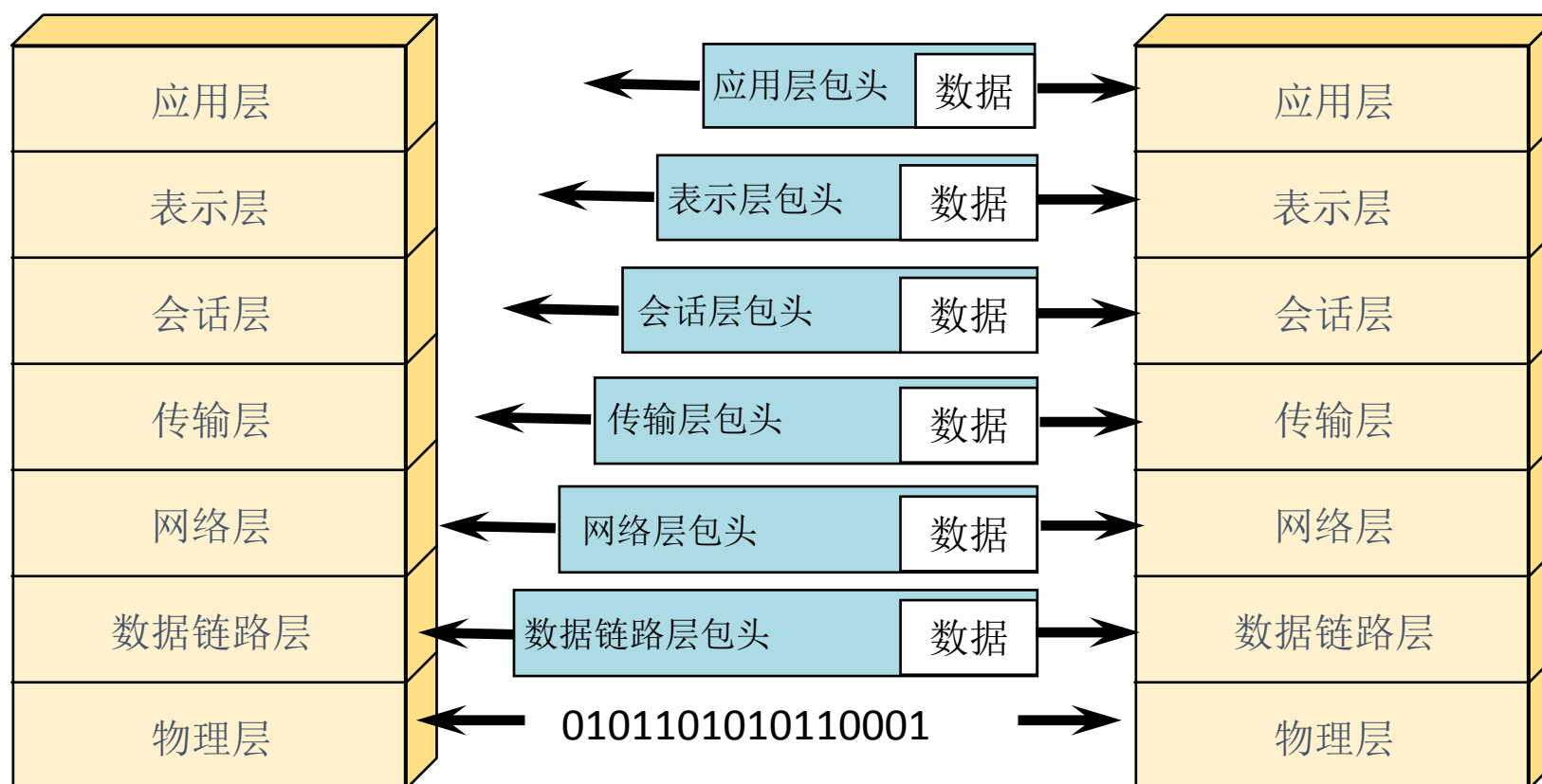


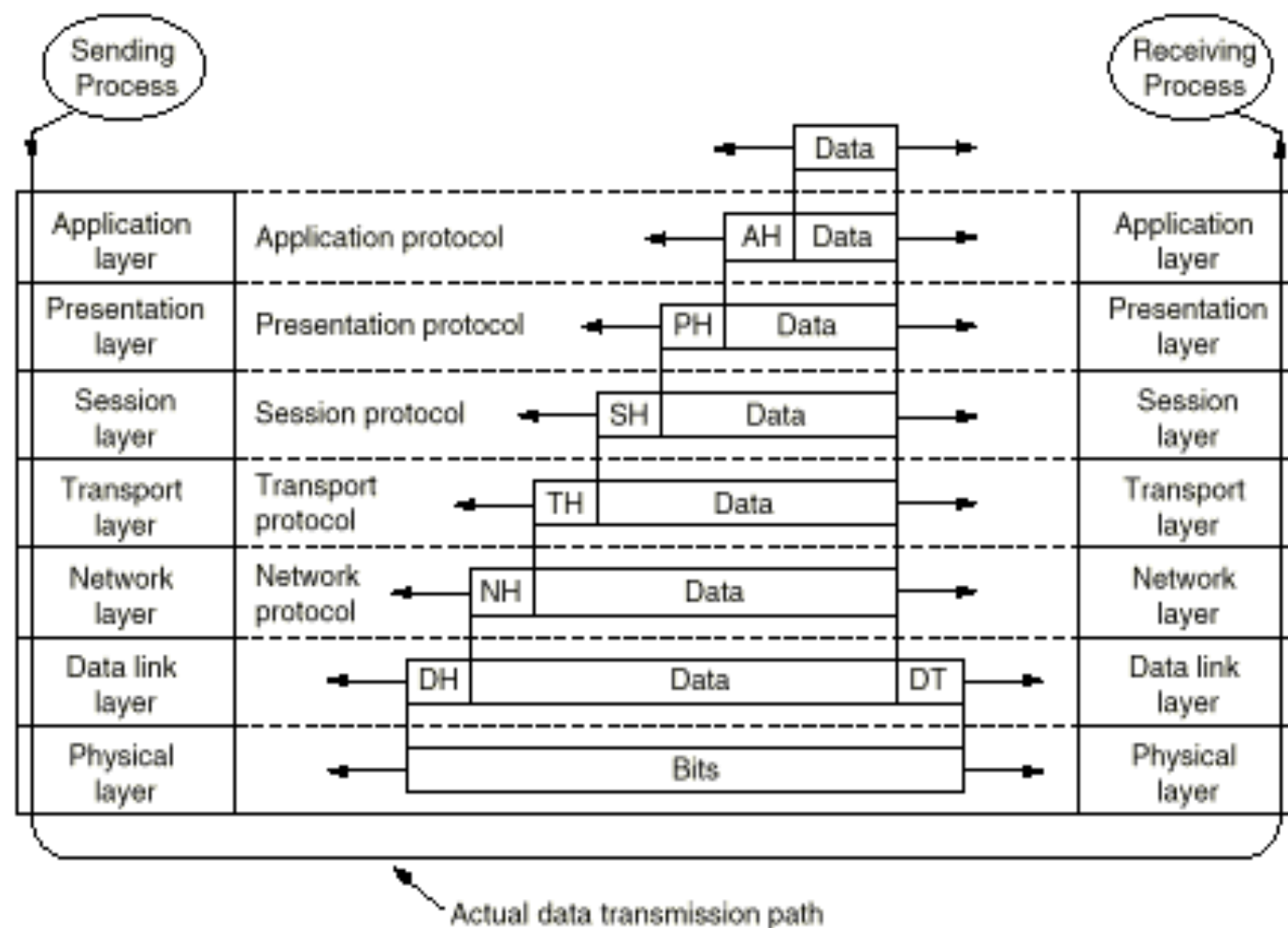


**Fig. 1-16.** The OSI reference model.

# ISO/OSI参考模型（续）

- 数据封装和解封装





**Fig. 1-17.** An example of how the OSI model is used. Some of the headers may be null. (Source: H.C. Folts. Used with permission.)

# ISO/OSI参考模型（续）

- OSI参考模型的历史经验和教训
  - OSI是80年代网络技术、体系结构的主流
    - OSI网络系统结构的核心和贡献
      - 分层模型
      - 协议、服务、接口
    - OSI的教训
      - 糟糕的时间
      - 糟糕的技术
      - 糟糕的实现
      - 糟糕的策略

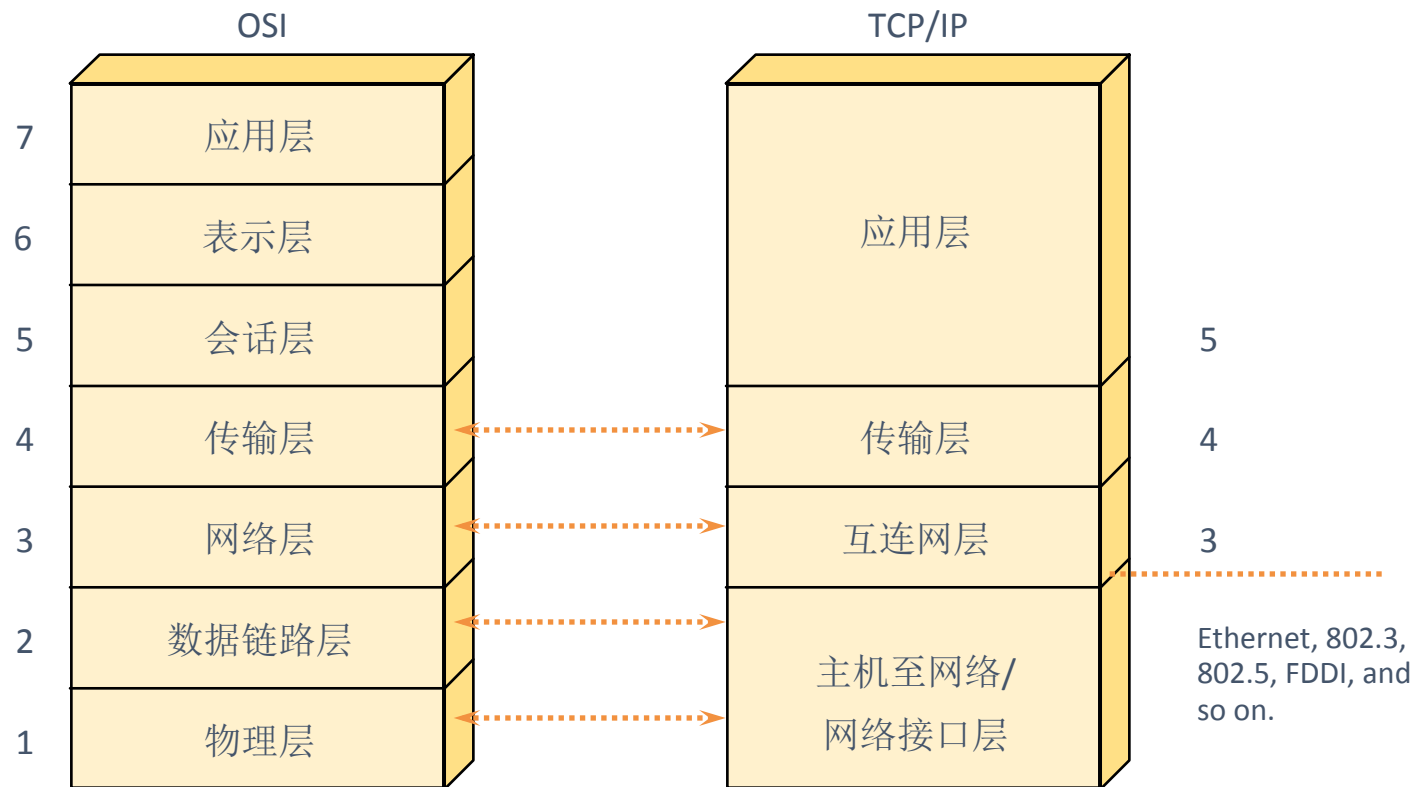
# TCP/IP参考模型

- TCP/IP参考模型
  - 以TCP/IP协议为核心的Internet网络体系结构
    - 应用层 (Application Layer)
      - 提供各种Internet管理和应用服务
    - 传输层 (Transport Layer)
      - 提供端到端的数据传输服务
    - 互连网层/IP层 (internet Layer)
      - 控制通信子网提供源点到目的点的分组传送
    - 主机至网络层 (Host-to-Network/Network Interface)
      - 物理层 (Physical Layer)
      - 数据链路层 (Data Link Layer)
      - TCP/IP协议没有进行明确规定



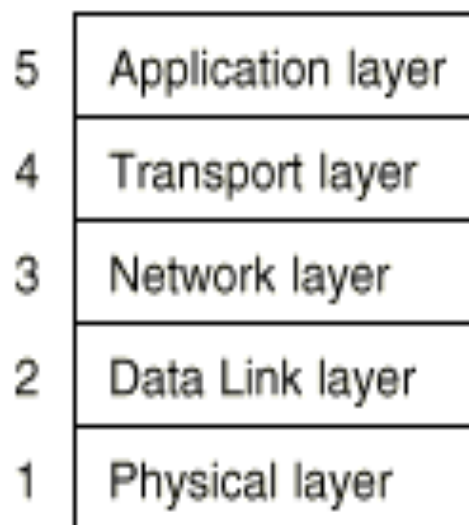
# TCP/IP参考模型

- TCP/IP与ISO/OSI的比较



# TCP/IP参考模型

- 本课程中使用的TCP/IP体系结构



**Fig. 1-21.** The hybrid reference model to be used in this book.

# TCP/IP参考模型

- TCP/IP体系结构中的协议

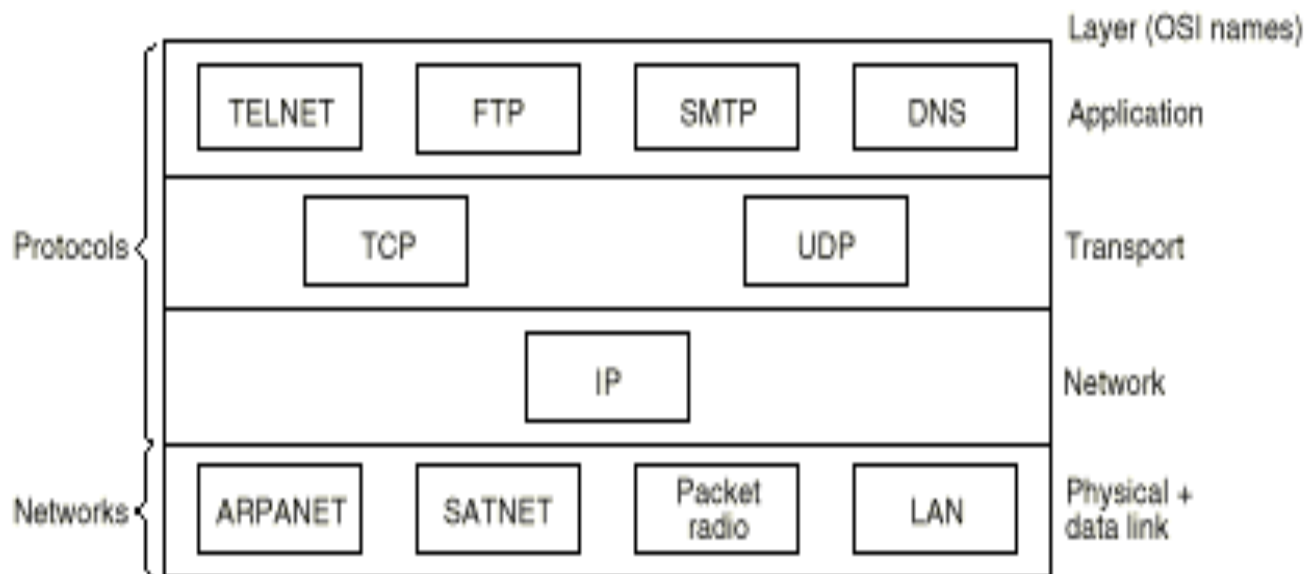
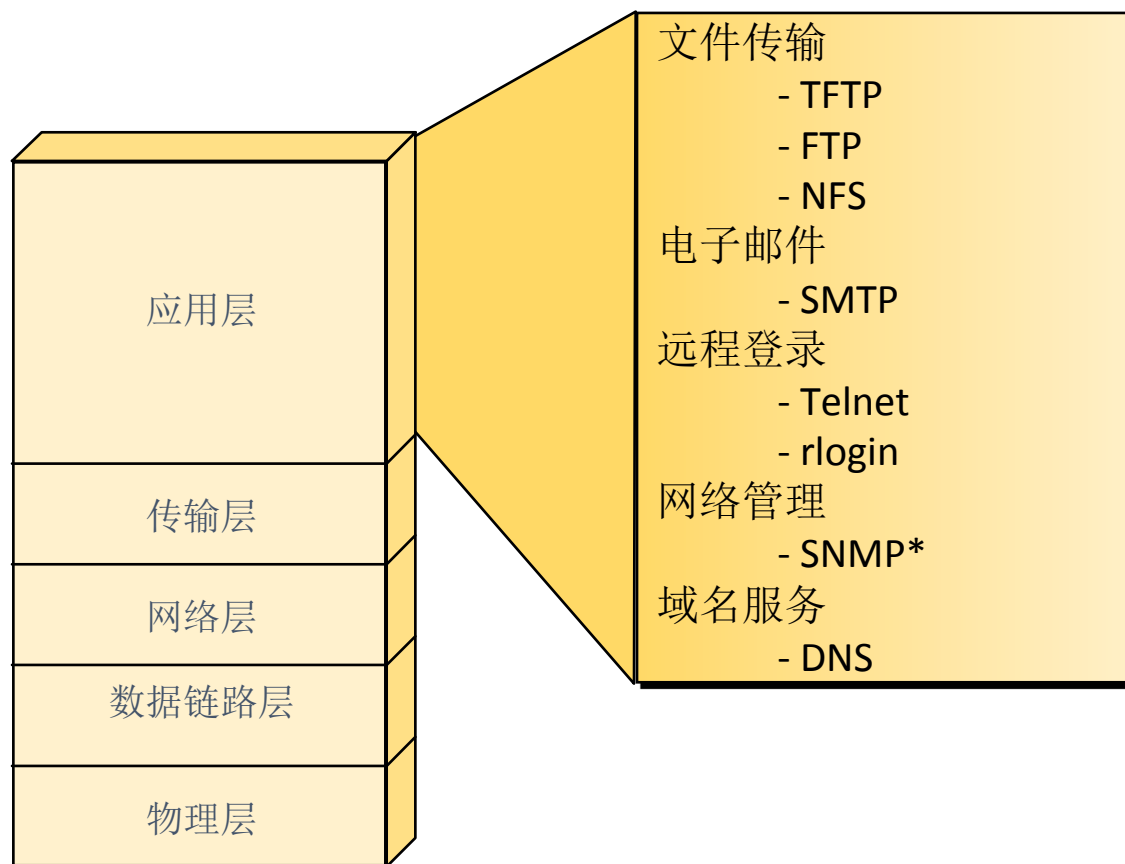


Fig. 1-19. Protocols and networks in the TCP/IP model initially.

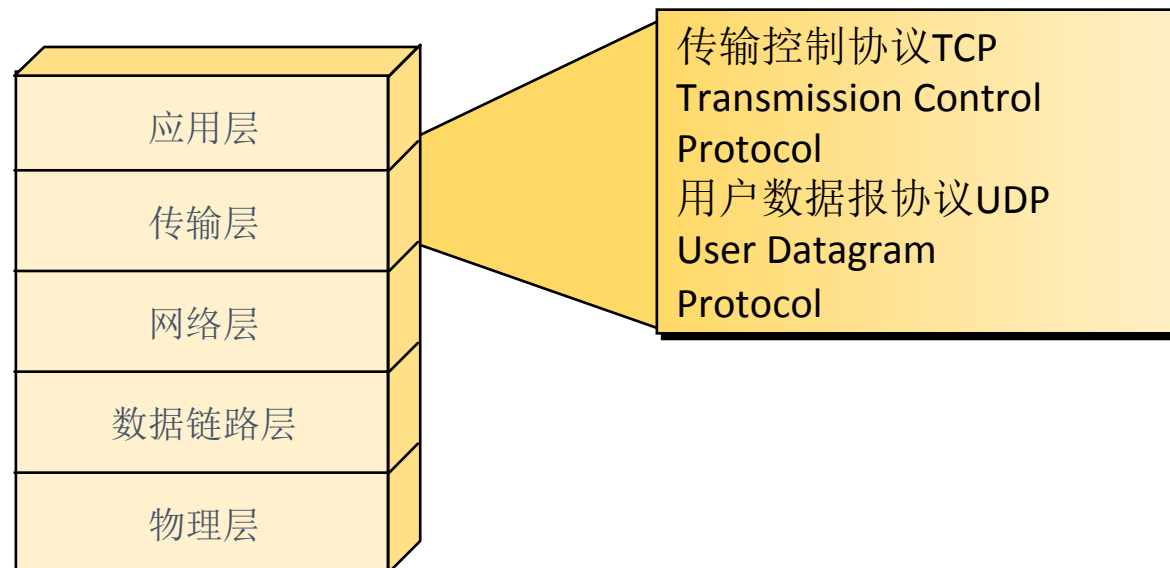
# TCP/IP参考模型

## • 应用层



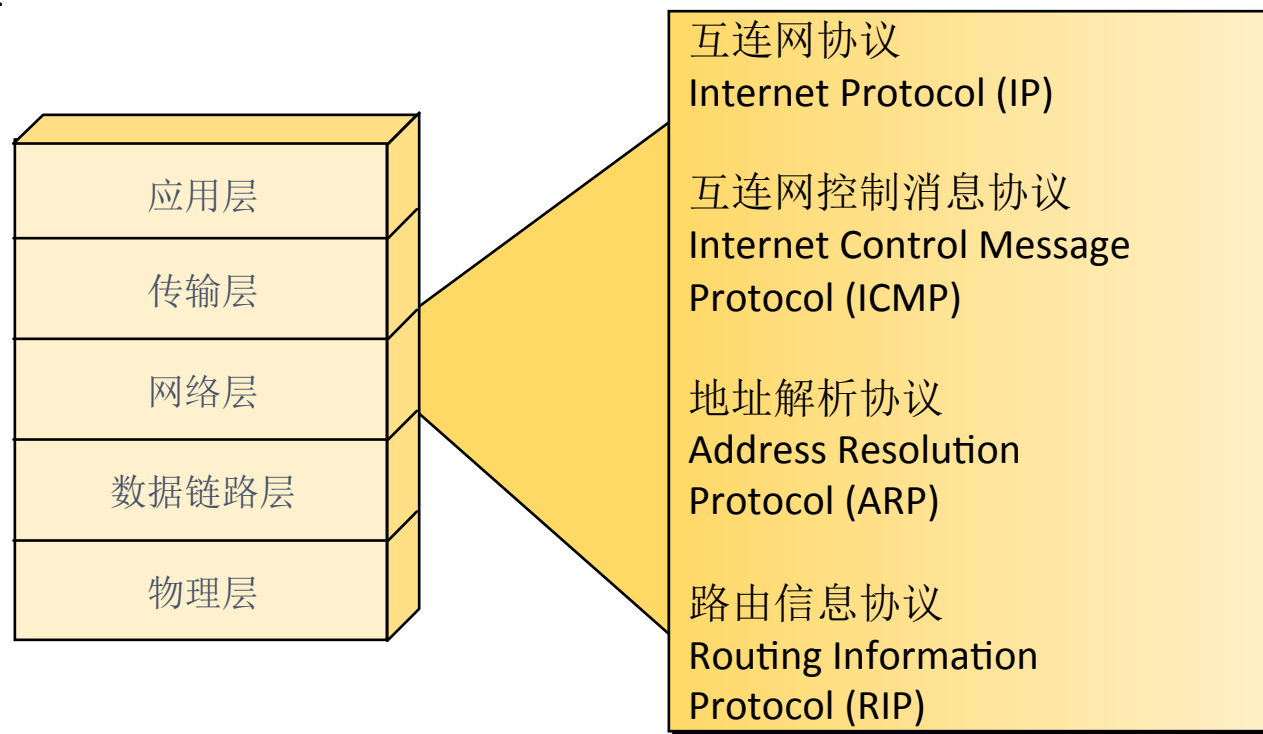
# TCP/IP参考模型

- 传输层
  - 端到端传输控制
    - 传输控制协议TCP：面向连接
    - 用户数据报协议UDP：无连接



# TCP/IP参考模型

- 互连网层
  - 定址和寻址

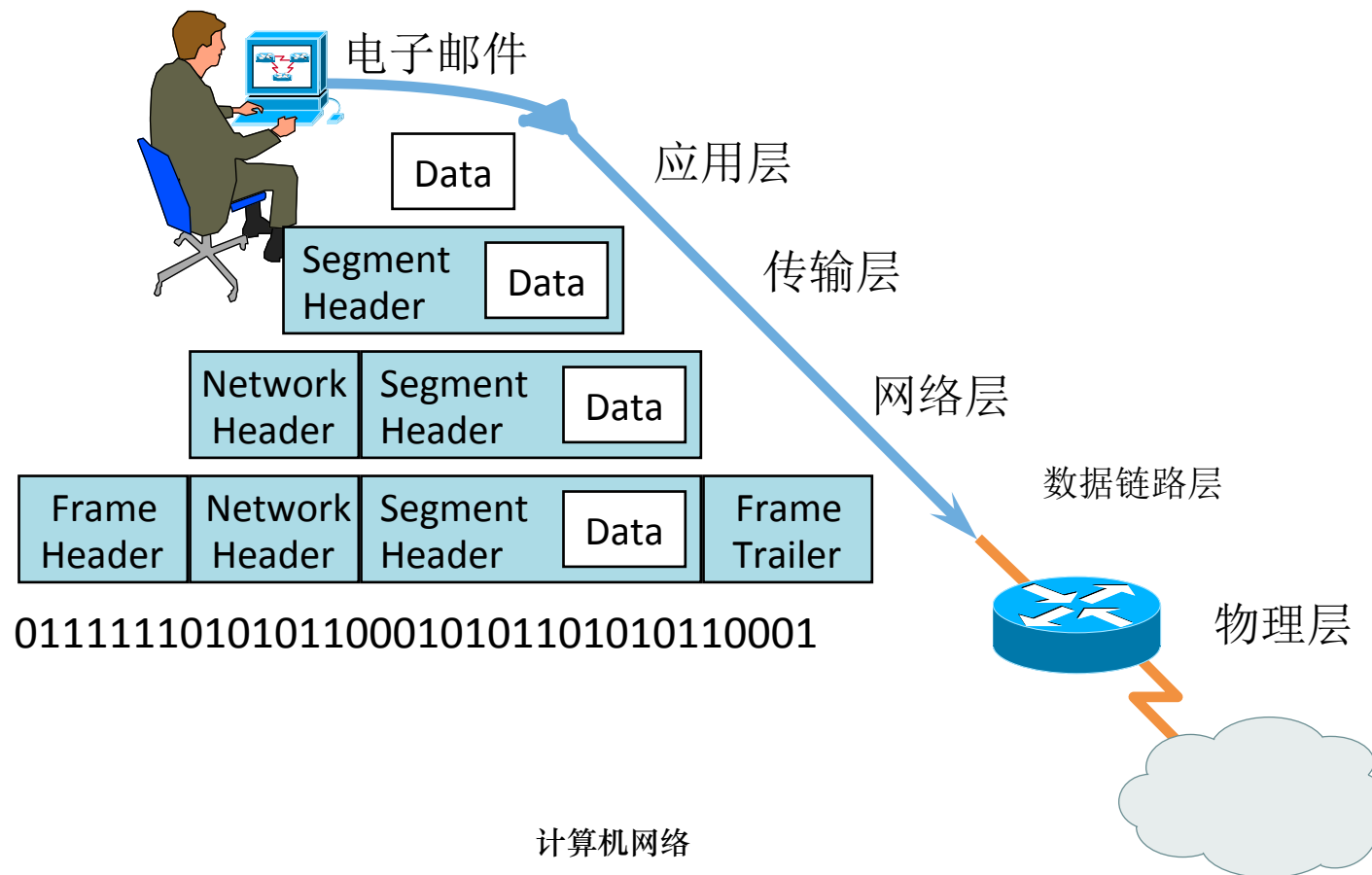


# TCP/IP参考模型

- 主机至网络层
  - 数据链路层和物理层
  - TCP/IP没有明确规定
    - 以太网 (Ethernet)
    - 令牌环网 (Token Ring)
    - 光纤分布式数据接口 (FDDI)
    - X.25
    - Packet Over SDH/SONET
    - ...

# TCP/IP参考模型

- 封装/解封装





# TCP/IP应用实例

- TCP/IP应用实例

- HTTP访问

- 典型的客户/服务器应用

- 客户端

- 安装TCP/IP协议栈的主机

- 支持HTTP的浏览器（IE、FireFox、Chrome、safari）

- 主动发起请求

- 服务器

- WWW服务器

- 被动监听，应答客户机的请求

- http://59.66.30.10/

- IPv6地址:

- http://ngmylife.edu.cn

# TCP/IP 实例

- 应用层信息交互

- 客户端请求 (Request)

- GET / HTTP/1.1
    - Host: 59.66.30.10
    - User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.12) Gecko/20070718 Red Hat/1.5.0.12-0.3.el4 Firefox/1.5.0.12
    - Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,\*/\*;q=0.5
    - Accept-Language: en-us,en;q=0.5
    - Accept-Encoding: gzip,deflate
    - Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7
    - Keep-Alive: 300
    - Connection: keep-alive

# TCP/IP 实例

- 服务器应答
  - HTTP/1.1 200 OK
  - ETag: W/"7879-1222229446000"
  - Last-Modified: Wed, 24 Sep 2008 04:10:46 GMT
  - Content-Type: text/html
  - Content-Length: 7879
  - Date: Thu, 25 Sep 2008 11:41:34 GMT
  - Server: Apache-Coyote/1.1
- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`
- `<html xmlns="http://www.w3.org/1999/xhtml">`
- `<head>`
- `<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />`
- `<title>NGmylife </title>`

# TCP/IP 实例

- 客户机请求
  - GET /ipv6css.css HTTP/1.1
  - Host: 59.66.30.10
  - User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.0.12) Gecko/20070718 Red Hat/1.5.0.12-0.3.el4 Firefox/1.5.0.12
  - Accept: text/css,\*/\*;q=0.1
  - Accept-Language: en-us,en;q=0.5
  - Accept-Encoding: gzip,deflate
  - Accept-Charset: ISO-8859-1,utf-8;q=0.7,\*;q=0.7
  - Keep-Alive: 300
  - Connection: keep-alive
  - Referer: http://59.66.30.10/

# TCP/IP 实例

- 服务器应答
    - HTTP/1.1 200 OK
    - ETag: W/"7436-1222229154000"
    - Last-Modified: Wed, 24 Sep 2008 04:05:54 GMT
    - Content-Type: text/css
    - Content-Length: 7436
    - Date: Thu, 25 Sep 2008 11:41:34 GMT
    - Server: Apache-Coyote/1.1
  - /\* CSS Document \*/
  - \* {font-family:Tahoma, Arial, Helvetica, sans-serif; font-size:12px;}
  - body{ padding:0; margin:0;}
  - a:link,a:visited,a:active{
  - text-decoration:none;
  - }
  - a:hover{
  - text-decoration:underline;
  - }
- 18-9-25

# TCP/IP 实例

- 工具

- Tcpdump (或者windump for windows)

- 命令参数

- -D: 显示端口列表
      - -i <interface>: 监听指定端口
      - -w <file>: 将分组内容保存到文件中
      - -r <file>: 从文件中读出保存的分组
      - -s <size>: 每个分组保存的长度
      - -x: 以十六进制方式显示分组内容
      - <expression>: 匹配条件

- 示例:

- Tcpdump -D
      - Tcpdump -i eth0 -s 200 -x -w tcpdump.pac host 59.66.30.10 and port 80

# TCP/IP 实例

- 实验步骤
  - 启动tcpdump
  - 启动浏览器，清除浏览器的缓存
  - 输入目标地址
  - `http://59.66.30.10/`
  - 网页打开后，Ctrl-C中断tcpdump
  - 使用tcpdump读出文件中保存的分组

# TCP/IP 实例

- Tcpdump 结果

0x0000: 0016 36e0 7ad0 0090 fb07 c774 0800 4500	..6.z.....t..E.
0x0010: 024e 7d66 4000 4006 2352 3b42 109f 3b42	.N}f@.@.#R;B..;B
0x0020: 10cf 80ea 0050 23fb ce5c 2065 cfc3 8018	.....P#..\e....
0x0030: 05b4 6b1d 0000 0101 080a 372d b339 4dc7	..k.....7-.9M.
0x0040: f50f 4745 5420 2f20 4854 5450 2f31 2e31	..GET./.HTTP/1.1
0x0050: 0d0a 486f 7374 3a20 3539 2e36 362e 3136	..Host:.59.66.16
0x0060: 2e32 3037 0d0a 5573 6572 2d41 6765 6e74	.207..User-Agent
0x0070: 3a20 4d6f 7a69 6c6c 612f 352e 3020 2858	:.Mozilla/5.0.(X
0x0080: 3131 3b20 553b 204c 696e 7578 2069 3638	11;.U;.Linux.i68
0x0090: 363b 2065 6e2d 5553 3b20 7276 3a31 2e38	6;.en-US;.rv:1.8
0x00a0: 2e30 2e31 3229 2047 6563 6b6f 2f32 3030	.0.12).Gecko/200
0x00b0: 3730 3731 3820 5265 6420 4861 742f 312e	70718.Red.Hat/1.
0x00c0: 352e 302e 3132 2d30 2e33 2e65 6c34 2046	5.0.12-0.3.el4.F
0x00d0: 6972 6566 6f78 2f31 2e35 2e30 2e31 320d	irefox/1.5.0.12.



数据链路层包头  
目的MAC地址: 0016-36e0-7ad0  
源MAC地址: 0090-fb07-c774  
协议类型: 0800

- Tcpdump结果

0x0000:	0016 36e0 7ad0 0090 fb07 c774 0800 4500	..6.z.....t..E.
0x0010:	024e 7d66 4000 4006 2352 3b42 109f 3b42	.N}f@.@.#R;B..;B
0x0020:	10cf 80ea 0050 23fb ce5c 2065 cfc3 8018	.....P#..\e....
0x0030:	05b4 6b1d 0000 0101 080a 372d b339 4dc7	..k.....7-.9M.
0x0040:	f50f 4745 5420 2f20 4854 5450 2f31 2e31	..GET./.HTTP/1.1
0x0050:	0d0a 486f 7374 3a20 3539 2e36 362e 3136	..Host:.59.66.16
0x0060:	2e32 3037 0d0a 5573 6572 2d41 6765 6e74	.207..User-Agent
0x0070:	3a20 4d6f 7a69 6c6c 612f 352e 3020 2858	:.Mozilla/5.0.(X
0x0080:	3131 3b20 553b 204c 696e 7578 2069 3638	11;.U;.Linux.i68
0x0090:	363b 2065 6e2d 5553 3b20 7276 3a31 2e38	6;.en-US;.rv:1.8
0x00a0:	2e30 2e31 3229 2047 6563 6b6f 2f32 3030	.0.12).Gecko/200
0x00b0:	3730 3731 3820 5265 6420 4861 742f 312e	70718.Red.Hat/1.
0x00c0:	352e 302e 3132 2d30 2e33 2e65 6c34 2046	5.0.12-0.3.el4.F
0x00d0:	6972 6566 6f78 2f31 2e35 2e30 2e31 320d	irefox/1.5.0.12.

网络层包头  
目的IP地址: 3b42-10cf (59.66.16.207)  
源IP地址: 3b42-109f (59.66.16.159)

- Tcpdump结果

```
0x0000: 0016 36e0 7ad0 0090 fb07 c774 0800 4500 ..6.z.....t..E.
0x0010: 024e 7d66 4000 4006 2352 3b42 109f 3b42 .N}f@. @. #R;B.;B
0x0020: 10cf 80ea 0050 23fb ce5c 2065 cfc3 8018 .....P#..\e....
0x0030: 05b4 6b1d 0000 0101 080a 372d b339 4dc7 ..k.....7-.9M.
0x0040: f50f 4745 5420 2f20 4854 5450 2f31 2e31 ..GET./.HTTP/1.1
0x0050: 0d0a 486f 7374 3a20 3539 2e36 362e 3136 ..Host:.59.66.16
0x0060: 2e32 3037 0d0a 5573 6572 2d41 6765 6e74 .207..User-Agent
0x0070: 3a20 4d6f 7a69 6c6c 612f 352e 3020 2858 :.Mozilla/5.0.(X
0x0080: 3131 3b20 553b 204c 696e 7578 2069 3638 11;.U;.Linux.i68
0x0090: 363b 2065 6e2d 5553 3b20 7276 3a31 2e38 6;.en-US;.rv:1.8
0x00a0: 2e30 2e31 3229 2047 6563 6b6f 2f32 3030 .0.12).Gecko/200
0x00b0: 3730 3731 3820 5265 6420 4861 742f 312e 70718.Red.Hat/1.
0x00c0: 352e 302e 3132 2d30 2e33 2e65 6c34 2046 5.0.12-0.3.el4.F
0x00d0: 6972 6566 6f78 2f31 2e35 2e30 2e31 320d irefox/1.5.0.12.
```

传输层包头  
目的端口: 0050 (80)  
源端口: 80ea (33002)

- Tcpdump结果

```
0x0000: 0016 36e0 7ad0 0090 fb07 c774 0800 4500  .6.z.....t..E.
0x0010: 024e 7d66 4000 4006 2352 3b42 109f 3b42  .N}f@.@.#R;B..;B
0x0020: 10cf 80ea 0050 23fb ce5c 2065 cfc3 8018  ....P#..\..e....
0x0030: 05b4 6b1d 0000 0101 080a 372d b339 4dc7  ..k.....7-.9M.
0x0040: f50f 4745 5420 2f20 4854 5450 2f31 2e31  ..GET./.HTTP/1.1
0x0050: 0d0a 486f 7374 3a20 3539 2e36 362e 3136  ..Host:.59.66.16
0x0060: 2e32 3037 0d0a 5573 6572 2d41 6765 6e74  .207..User-Agent
0x0070: 3a20 4d6f 7a69 6c6c 612f 352e 3020 2858  :.Mozilla/5.0.(X
0x0080: 3131 3b20 553b 204c 696e 7578 2069 3638  11;.U;.Linux.i68
0x0090: 363b 2065 6e2d 5553 3b20 7276 3a31 2e38  6;.en-US;.rv:1.8
0x00a0: 2e30 2e31 3229 2047 6563 6b6f 2f32 3030  .0.12).Gecko/200
0x00b0: 3730 3731 3820 5265 6420 4861 742f 312e  70718.Red.Hat/1.
0x00c0: 352e 302e 3132 2d30 2e33 2e65 6c34 2046  5.0.12-0.3.el4.F
0x00d0: 6972 6566 6f78 2f31 2e35 2e30 2e31 320d  irefox/1.5.0.12.
```

应用层信息！

- Tcpdump结果

0x0000: 0016 36e0 7ad0 0090 fb07 c774 0800 4500	.....t..E.
0x0010: 024e 7d66 4000 4006 2352 3b42 109f 3b42	<a href="#">f@.@.#R;B.;B</a>
0x0020: 10cf 80ea 0050 23fb ce5c 2065 cfc3 8018	.....P#..\e....
0x0030: 05b4 6b1d 0000 0101 080a 372d b339 4dc7	..k.....7-.9M.
0x0040: f50f <b><u>4745 5420 2f20 4854 5450 2f31 2e31</u></b>	..GET./.HTTP/1.1
0x0050: <b><u>0d0a 486f 7374 3a20 3539 2e36 362e 3136</u></b>	..Host:.59.66.16
0x0060: <b><u>2e32 3037 0d0a 5573 6572 2d41 6765 6e74</u></b>	.207..User-Agent
0x0070: <b><u>3a20 4d6f 7a69 6c6c 612f 352e 3020 2858</u></b>	:.Mozilla/5.0.(X
0x0080: <b><u>3131 3b20 553b 204c 696e 7578 2069 3638</u></b>	11;.U;.Linux.i68
0x0090: <b><u>363b 2065 6e2d 5553 3b20 7276 3a31 2e38</u></b>	6;.en-US;.rv:1.8
0x00a0: <b><u>2e30 2e31 3229 2047 6563 6b6f 2f32 3030</u></b>	.0.12).Gecko/200
0x00b0: <b><u>3730 3731 3820 5265 6420 4861 742f 312e</u></b>	70718.Red.Hat/1.
0x00c0: <b><u>352e 302e 3132 2d30 2e33 2e65 6c34 2046</u></b>	5.0.12-0.3.el4.F
0x00d0: <b><u>6972 6566 6f78 2f31 2e35 2e30 2e31 320d</u></b>	irefox/1.5.0.12.

# 小结

- 计算机网络体系结构
- ISO/OSI参考模型
- TCP/IP参考模型
- TCP/IP网络应用实例

# 作业

- 1、TCP和UDP之间最主要的区别是什么?
- 2、一个系统有 $n$ 层协议的层次结构，应用程序产生的消息长度为 $M$ 字节，在每一层上需要加上 $h$ 字节的头，请计算网络在传输这些分组时的效率。

## 作业（续）

- 通过浏览器访问清华大学BBS，使用tcpdump记录访问中发生的分组交换过程，分析结果并形成报告（可以不提交）
  - 应用层信息的类型
  - 端口号和地址信息
  - 统计分组数目
    - 客户端发给服务器的应用层消息数目
    - 客户端发给服务器的分组数目
    - 服务器发给客户端的应用层消息数目
    - 服务器发给客户端的分组数目

# 作业