

计算机网络

T09

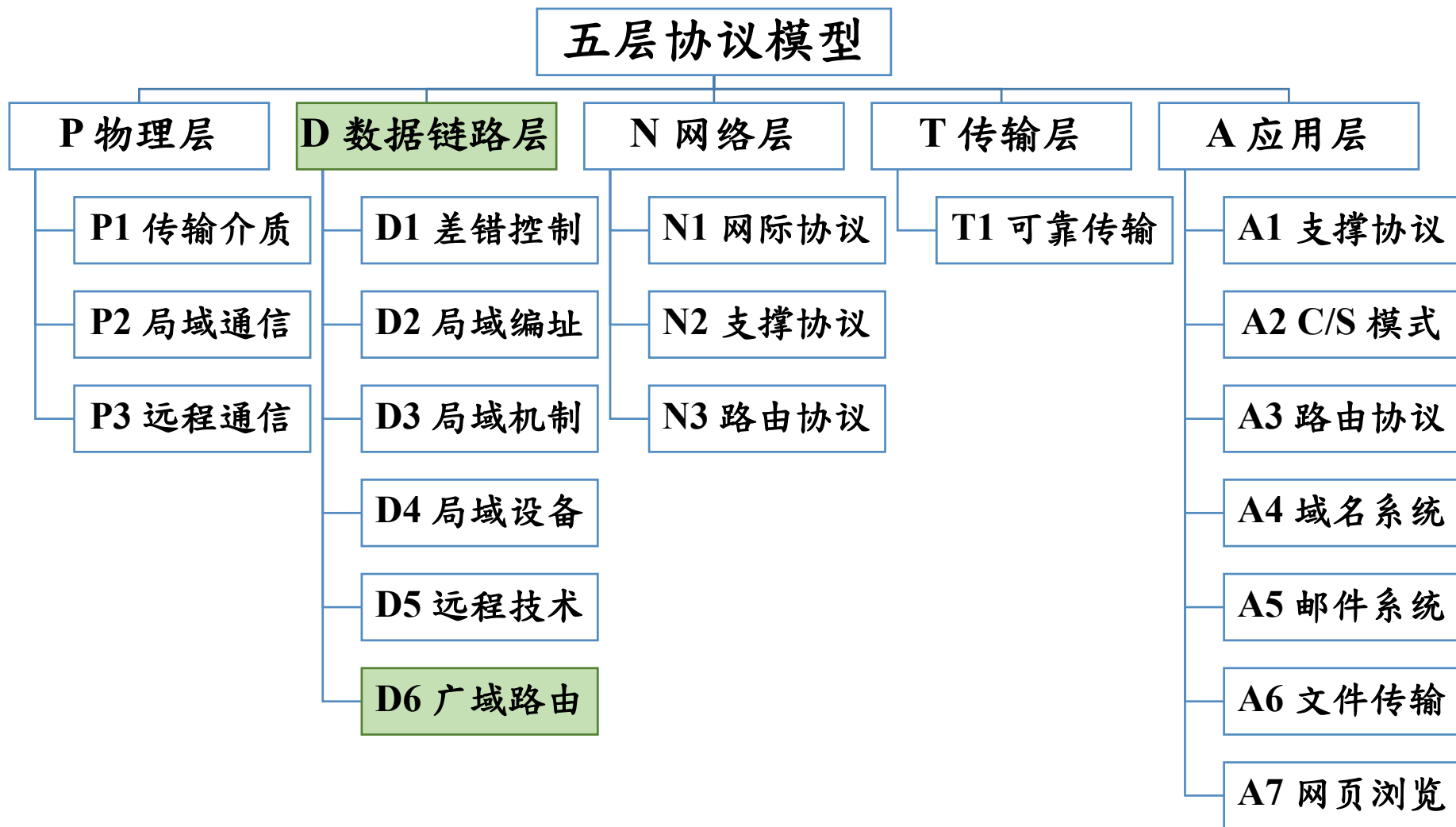


# 广域网技术 和协议分层

厦门大学信息学院软件工程系

黄炜 副教授

# 主要内容



# 主要内容

- 广域网技术与路由

- 广域网分层地址
- 存储转发和路由、路由算法：LSR和DVR
- 下一跳和路由表

- 协议与分层

- OSI/ISO协议（七层）、TCP/IP协议（四层）、五层协议
- 分层的优点、各层次的作用
- 帧在各层次间的封装和拆封



# 对应课本章节

- **PART III Packet Switching And Network Technologies**
  - Chapter 18 WAN Technologies And Dynamic Routing
- **PART IV Internetworking**
  - Chapter 20 Internetworking: Concepts, Architecture, and Protocols



# 1. 广域网技术与路由



# 大网络和广域

- 网络技术分为三大类：
  - 局域网（ Local Area Network ， LAN ）
  - 城域网（ Metropolitan Area Network ， MAN ）
  - 广域网（ Wide Area Network ， WAN ）

表 1-3 多个处理机互连的系统按其大小的分类

处理机之间的典型距离	处理机所在的范围	实 例
0.1 m	印制板	数据流计算机
1 m	系统	多处理机
10 m	房间	局域网、校园网、企业网
100 m	建筑物	
1 km	校园	
10 km	城市	城域网
100 km	国家	广域网
1000 km	国家，洲	广域网，互连的广域网



# 广域网遇到的新问题

## 局域网

距离近，通信随机发生  
主机少，设备性能接近

## 广域网

距离远，不确定因素多  
主机多，设备性能各异

• 解决方案：高层“分组交换机”

— 中央处理器：转发到下一站

— 存储器：帧，转发信息

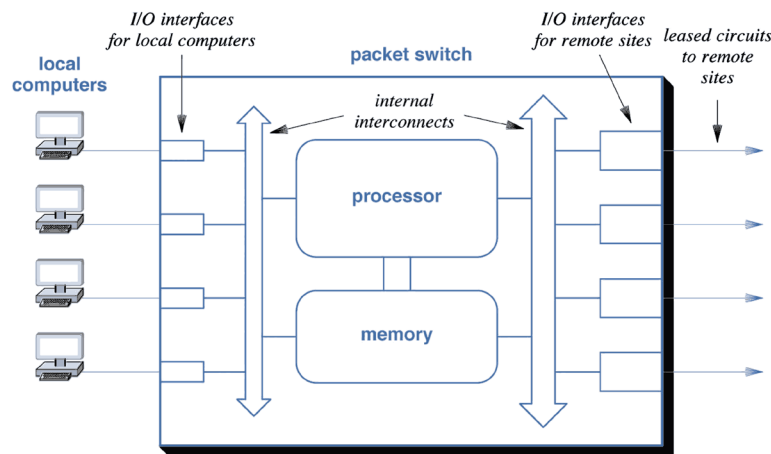


Figure 18.1 Illustration of traditional packet switch architecture.

# 交换机的机制：存储转发

- 存储转发（Store and Forward）
- 存储
  - 存储操作发生于分组到达时
  - 交换机中的I/O硬件将数据包的副本缓存在存储器中
- 转发
  - 转发操作发生于分组到达并等待在内存中。
  - 处理器检查包，解析目的地址，将数据包发送到通向目的地的I/O接口上





# 广域网的寻址方案

- 主机多使得广域网需要分层寻址方案

- 站点号+主机号：(site, computer at the site)

- 第一部分标识分组交换机：每个分组交换机被分配一个唯一的号码
    - 第二部分标识特定计算机



**Figure 18.4** Example of an address hierarchy where each address identifies a packet switch and a computer attached to the switch.

# 下一跳转发：层次寻址

- 数据包到达时，交换机必须选择一个传出路径来转发
- 交换机检查数据包的目的地址，提取交换机号
  - 本地：交换机将数据包直接发送到计算机
    - 如果目的地址中的数字与分组交换机自身的ID相同，则该分组将用于本地分组交换机上的计算机
  - 否则：数据包必须转发到另一个交换机
- 只基于交换机ID转发
  - 这意味着交换机只需要知道哪个传出链路



# 下一跳转发：层次寻址

- 交换机不需要保有所有可达的计算机的完整信息，也没有一个交换机需要计算整个路由
- 交换机只需要计算数据包的下一跳
- 该过程称为下一跳转发

## Algorithm 18.1

Given:

A packet that has arrived at packet switch Q

Perform:

The next-hop forwarding step

Method:

Extract the destination address from the packet;

Divide the address into a packet switch number, P, and a computer identification, C;

if (P == Q) { /\* the destination is local \*/

Forward the packet to local computer C;

} else {

Select a link that leads to another packet switch, and forward

the packet over the link;

}

**Algorithm 18.1** The two steps a packet switch uses to forward a packet when using next-hop forwarding.



# 下一跳转发：转发表

- 转发表：存储目的交换机和对应下一跳的表
  - 列出所有可能的分组交换机，并为每个给出下一跳
  - 这些“转发表”本应称为路由表（routing table）
  - 将数据包转发到下一跳的过程称为路由（routing）

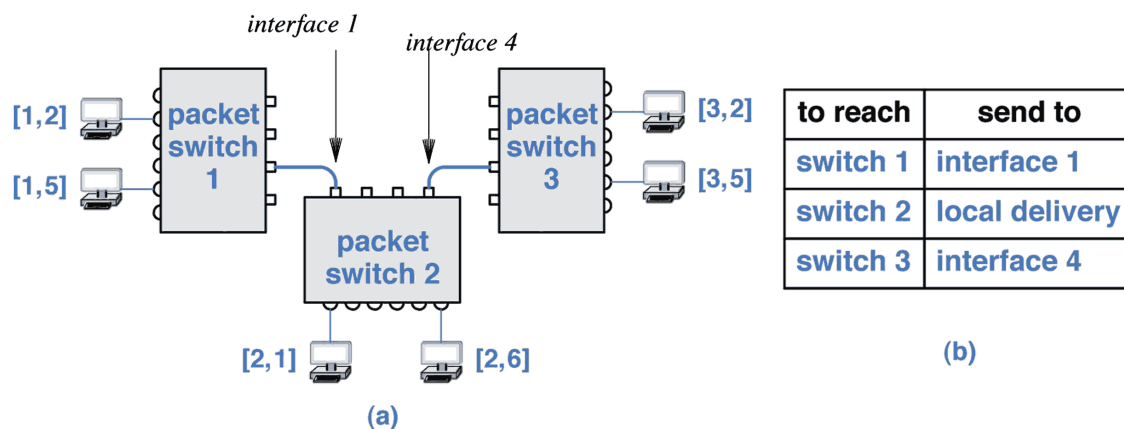


Figure 18.5 (a) A network of three packet switches, and (b) the next-hop forwarding table for switch 2.



# 下一跳转发：层次地址的优点

- 只用层次地址的一部分来转发包的优点

- 转发表所需的计算时间减少。

- 因为转发表可以被组织成一个数组，使用索引，而不是搜索

- 转发表所需的存储空间减少。

- 包含每个包交换的一个条目，而不是每个目标计算机的一个条目表

- 存储空间可能大大减少。广域网每个分组交换机连着许多计算机。

- 地址的第二部分作用

- 一旦数据包到达最终交换机，用该部分选择特定的计算机



# 源独立 ( Source Independence )

- 源独立性是指，下一跳转发不依赖于
  - 数据分组的原始源
  - 数据分组到达前的特定路径
- 下一跳仅取决于数据分组的目的地
- 源独立令计算机网络中的转发机制显得紧凑和高效



# 广域网的设备

- 广域网使用路由器连接成网络

- 第2层交换机 ( Switch ) 主要发挥交换功能
- 路由器 ( Router ) 主要发挥路由功能
- 许多广域网租用专用数据电路，其他形式也可用

- 广域网不需要对称

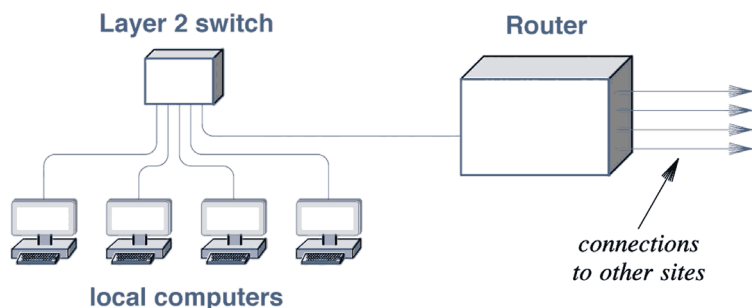


Figure 18.2 Illustration of a modern WAN site with local communication handled by a separate LAN.

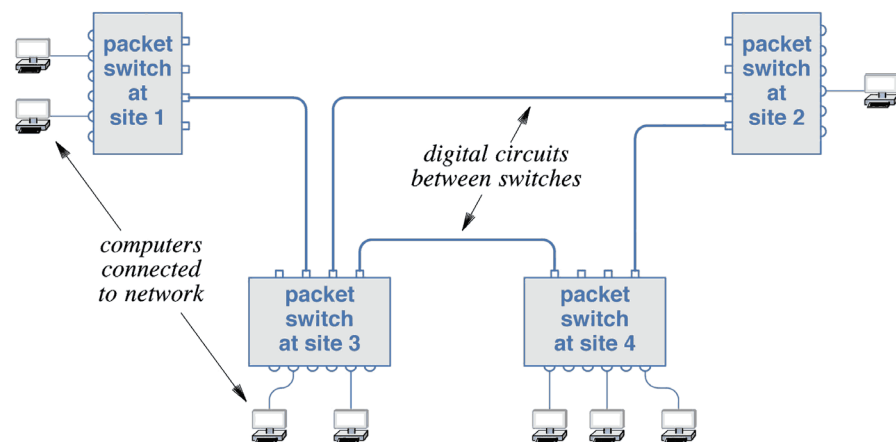


Figure 18.3 An example WAN formed by interconnecting packet switches.

# 路由表计算

- 静态路由 ( Static routing )

- 交换机启动时，计算并安装路由；路由不会改变
- 优点：简单和低开销；
- 缺点：缺乏灵活性，当通信中断时，无法更改静态路由

- 动态路由 ( Dynamic routing )

- 交换机启动，生成初始路由表；网络改变时，程序改变表。
- 大型网络的设计留有处理偶尔的硬件故障的冗余
- 大多数广域网使用一种动态路由





# 广域网动态路由更新

- 路由表中的值必须保证
  - 通用通信：转发表须包含每个目标地址的有效下一跳路由
  - 最优路线：转发表中下一跳值必须指向目标的最短路径
- 网络故障进一步使得转发复杂化
  - 网络管理器不能只配置转发表来包含不更改的静态值
  - 相反，运行在分组交换机的软件应不断测试失败，并自动重新配置转发表



# 路由表的制作

- 图抽象出细节，允许路由软件处理问题的本质

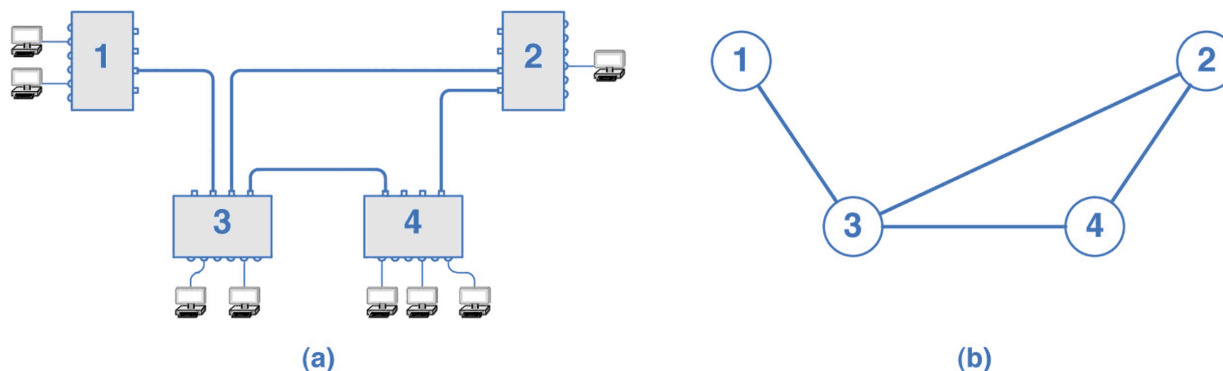


Figure 18.6 Illustration of a WAN and the corresponding graph.

to reach	next hop	to reach	next hop	to reach	next hop	to reach	next hop
1	±	1	(2,3)	1	(3,1)	1	(4,3)
2	(1,3)	2	±	2	(3,2)	2	(4,2)
3	(1,3)	3	(2,3)	3	±	3	(4,3)
4	(1,3)	4	(2,4)	4	(3,4)	4	±
node 1		node 2		node 3		node 4	

Figure 18.7 A forwarding table for each node in the graph of Figure 18.6b.

# 使用默认路由 ( Default Routes )

- 默认路由

- 一种允许转发表中的单个条目替换具有相同下一跳值的长条目列表的机制。
- 只允许一个默认条目。

- 默认路由是可选的

- 仅当多个目的地具有相同的下一跳值时才存在默认条目

to reach	next hop
1	±
*	(1,3)

node 1

to reach	next hop
2	±
4	(2,4)
*	(2,3)

node 2

to reach	next hop
1	(3,1)
2	(3,2)
3	±
4	(3,4)

node 3

to reach	next hop
2	(4,2)
4	±
*	(4,3)

node 4



# 路由表计算

- 广域网需要进行分布式路由计算
  - 而非一个集中的程序计算所有最短路径
  - 每个交换机必须计算自己的转发表
  - 本地所有交换机必须参与分布式路由计算

- 两种形式

- 链路状态路由 ( Link-State Routing , LSR )
- 距离矢量路由 ( Distance-Vector Routing , DVR )

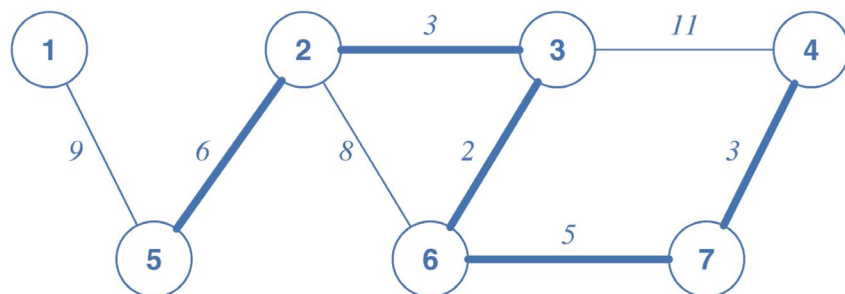


Figure 18.9 An example graph with a weight assigned to each edge and a shortest path between nodes 4 and 5 shown darkened.

# 链路状态路由（LSR）

- 工作方式是Dijkstra算法，又称为最短路径优先路由
- 分组交换机定期发送横跨链路状态的网络消息
  - 例如，分组交换机5和9测量它们之间的链路并发送状态消息如“5和9之间的链接是通的”
  - 每个状态消息都广播到所有交换机
- 每个交换机收集传入状态消息并利用它们建立网络图
- 然后使用算法产生一个转发表，通过选择本身作为源



# 链路状态路由 (LSR)

- LSR算法能够适应硬件故障
- 如果分组交换机之间的链路失败
  - 连接的交换机检测故障，广播指定链接已关闭的状态消息
  - 所有分组交换机接收广播，更改图的副本以反映链接状态的变化，重新计算最短路径
- 当一个链接再次可用
  - 连接到链路的分组交换机检测到它正在工作
  - 开始发送状态消息，报告其可用性



# 距离矢量路由 (DVR)

- 如同LSR，网络中的每一个链路都分配一个权重
  - 两个交换机间的目的地的距离被定义为两者之间的路径的权重的总和
- DVR安排交换机定期交换信息，但消息不被广播
- 发送一个完整的清单，目的地和到达各项的当前成本
- 当发送一个DVR的消息，交换机发送一系列单独语句
  - 形式（目的地，距离）：“我到达目的地X的距离是y”
  - 每个交换机周期性地发送一个消息给邻居



# 距离矢量路由 (DVR)

- DVR维持一个扩展的转发表，为每个目的地存储距离
  - 每个交换机必须保留一个目的地列表以及当前到达目的地下一跳的列表，以便在转发表中找到目的地列表和下一跳
- 当消息从邻居N到达交换机时，检查消息中的每个项
  - 如果邻居对某些目标拥有较短的路径，则更改其转发表
    - 例子:如果邻居N广告路径到目的地D的成本为5，和当前路径通过邻居K成本为100，D的当前下一跳将被N替换，而达到D的成本将是5加上达到N的成本





# 路由的问题

- 理论上，LSR或DVR计算最短路径，最终会收敛
  - 表示所有分组交换机中的转发表一致
- 如果LSR消息丢失，两个交换机的最短路径不一致
- DVR消息丢失的问题会更严重
  - 主要原因之一DVR协议具有来自反冲洗的问题
    - 例如，假设一个交换机告诉邻居“我可以以成本3到达目的地D”
    - 如果连接导致目标D失败交换机将从其转发表中删除D项（或标记无效），但交换机已经告诉邻居存在路由，将创建一个路由循环



## 2. 协议与分层



# 计算机网络体系结构的形成

- 协议 ( protocol ) , 网络协议
  - 参与通信的各方必须在交换消息时使用的一组指定的消息格式和每个消息所需的适当的行动规则。
  - 实施这些规则的软件称为协议软件 ( protocol software ) 。
- 分层模型 ( layering model )
  - 将通信问题划分成子块，为每个子块设计一个单独的协议。
  - 使得每个协议更容易设计，分析，实现和测试。
  - 每个协议应该处理其他协议无法处理的通信问题的一部分。



# 协议系列 ( Protocol Suites )

- 协议的设计为完整的、相互配合的集合，称为系列 ( suites ) 或族 ( family ) 。
  - 不是孤立地创建每个协议
  - 协议组合应处理所有可能的硬件故障或其他异常情况。
  - 在一个系列中的每个协议解决了通信问题的一部分，他们一起解决了整个通信问题。
- 在本章，学习每个层的目的和协议如何用于通信是足够的 ( 整本书是细节 )



# 两种国际标准

- 只要遵循标准，一个系统就可以和位于世界上任何地方的、也遵循这同一标准的任何系统进行通信。
  - 法律上的(de jure)国际标准：ISO/OSI 模型
  - 事实上的(de facto)标准：TCP/IP 模型

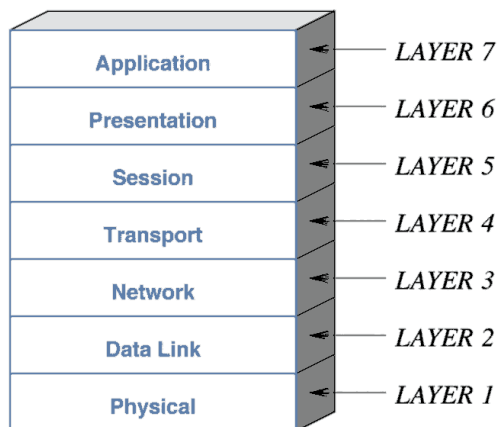


Figure 1.4 The OSI seven-layer model standardized by ISO.

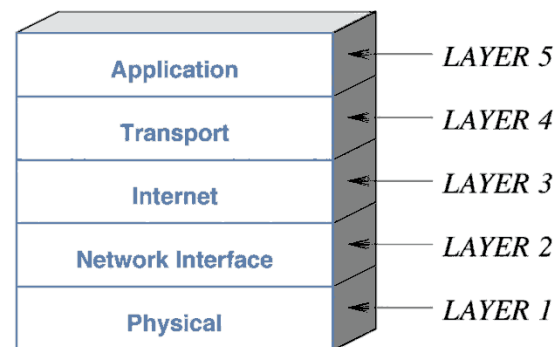


Figure 1.1 The layering model used with the Internet protocols (TCP/IP).

# 分层的好处

- 分层的好处

- 各层之间是独立的，灵活性好，结构上可分割开。
- 易于实现和维护。
- 能促进标准化工作。

- 层数多少要适当

- 若层数太少，就会使每一层的协议太复杂。
- 层数太多又会在描述和综合各层功能的系统工程任务时遇到较多的困难。



# 协议的复杂性

- 协议必须把所有不利的条件事先都估计到，而不能假定一切都是正常的和非常理想的。
- 看一个计算机网络协议是否正确，不能光看在正常情况下是否正确，而且还必须非常仔细地检查这个协议能否应付各种异常情况。



# OSI/ISO参考模型

- **OSI/ISO参考模型**

- 英文：Open Systems Interconnection Reference Model
- (1) 物理层；(2) 数据链路层；(3) 网络层；
- (4) 传输层；(5) 会话层；(6) 表示层；(7) 应用层

- **在市场化方面 OSI 失败的原因**

- 专家们在完成 OSI 标准时没有商业驱动力；
- 周期太长，按 OSI 标准生产的设备无法及时进入市场；
- 层次划分不合理，有些功能在多个层次中重复出现。





# TCP/IP协议族

- TCP/IP协议族

- (1) 网络接口层：最下面的网络接口层并没有具体内容；
- (2) 网际层；(3) 传输层；(4) 应用层

- 不足

- 网际层之下的网络接口层较为笼统，没有具体内容



# 五层协议体系结构

- 往往折中综合 OSI 和 TCP/IP 的优点，采用五层协议
  - 应用层(application layer)
  - 传输层(transport layer)
  - 网络层(network layer)
  - 数据链路层(data link layer)
  - 物理层(physical layer)



# 分层的协议

- 承上启下

- 两个对等实体间的通信使得本层能够向上一层提供服务。
- 要实现本层协议，还需要使用下层所提供的服务。
- 本层的服务用户只能看见服务而无法看见下面的协议。
- 下面的协议对上面的服务用户是透明的。

- 协议是水平的：协议是控制对等实体之间通信的规则。
- 服务是垂直的：服务是由下层向上层通过层间接口提供的。



# ISO-OSI层次模型的软件功能

- 分层的数据结构

- 数据头

- 抽象的概念，不一定在开始位置

- 有效数据

- 有可能经过一定的修正

- 层次间的调用关系

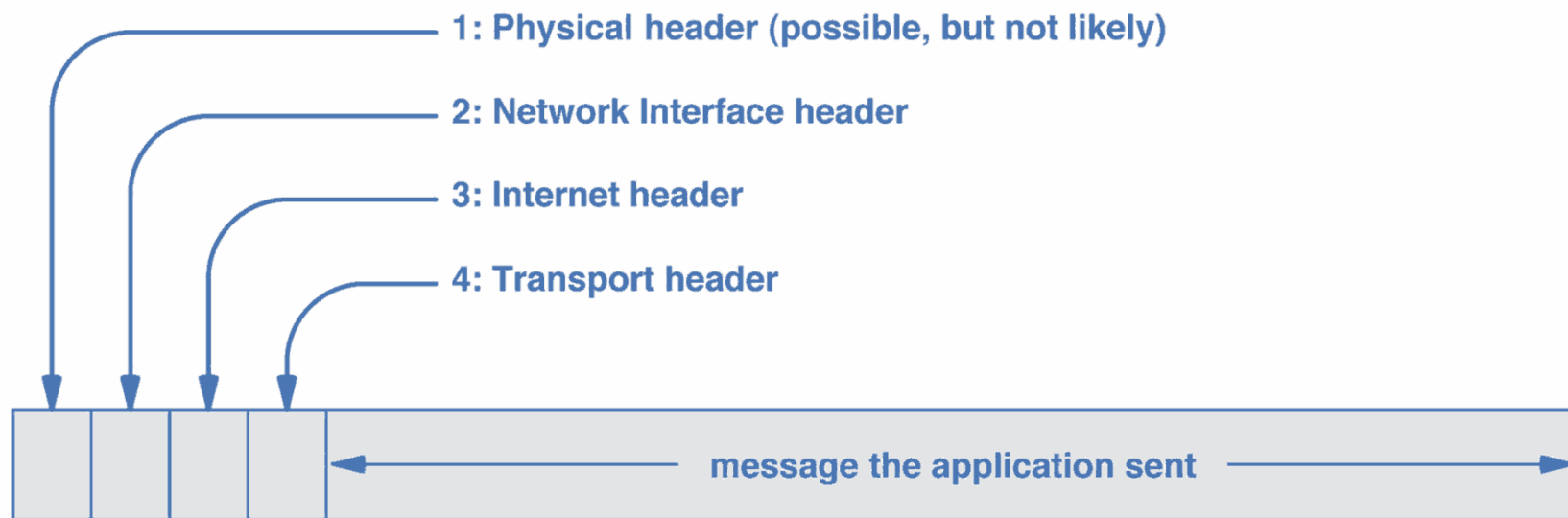
- 底层为上层提供服务访问点(SAP)作为接口

- 上层通过统一的SAP底层功能



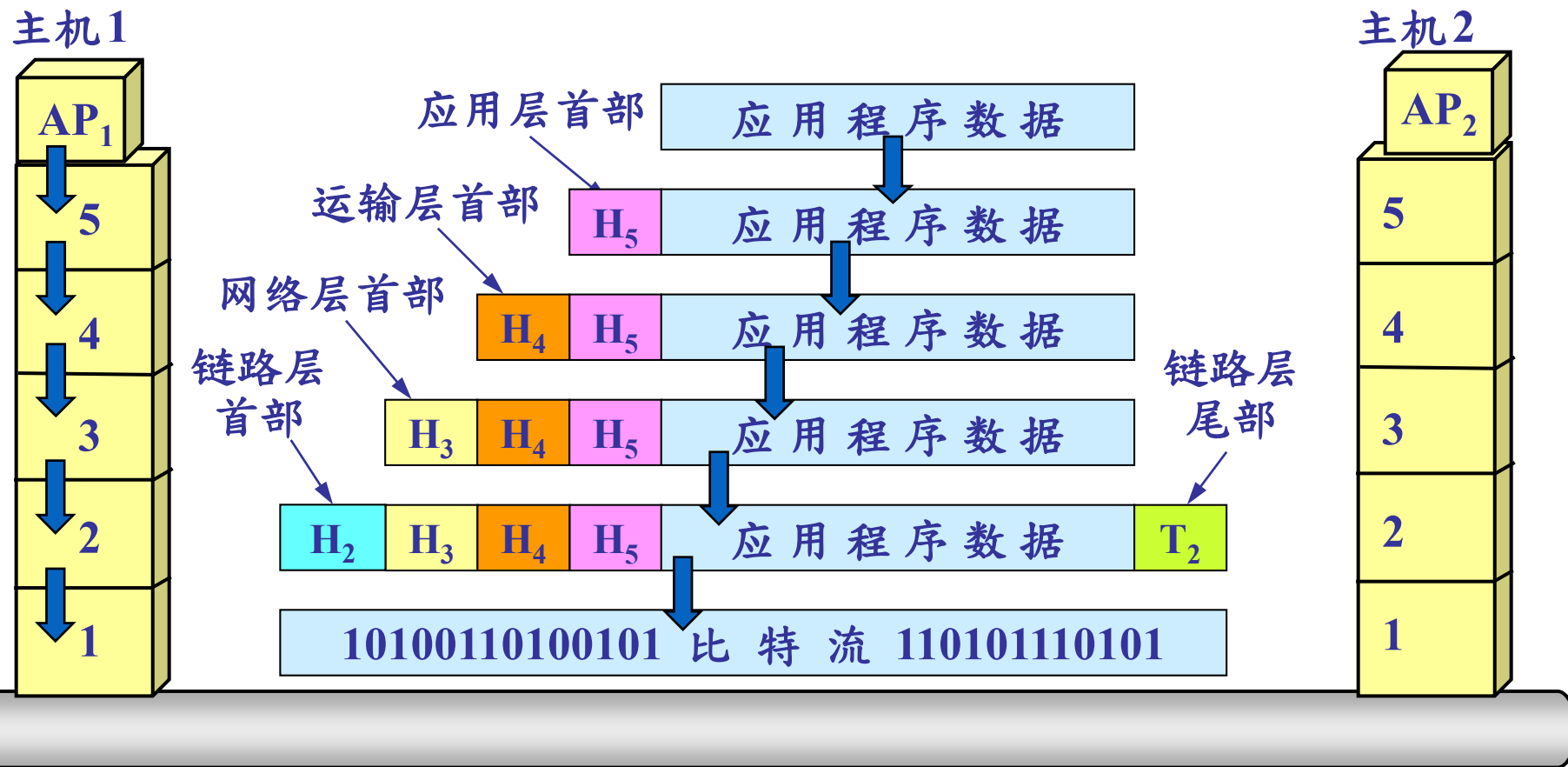
# 数据在各层次之间的传输

- 加入或剥去首部（尾部）的层次
  - 发送计算机上的给定层中的软件向传出数据添加信息。
  - 接收计算机上同一层的软件使用附加信息处理输入的数据。

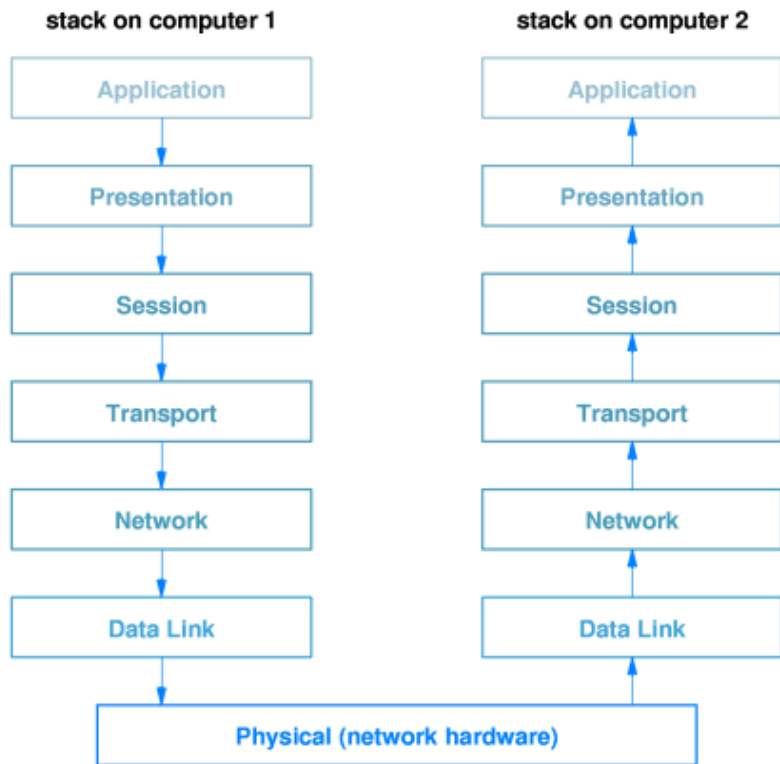


# 数据在各层次之间的传输

- 加入或剥去首部（尾部）的层次



# OSI七层模型



# OSI七层模型的作用

- (1) 物理层

- 完成对bit和载波之间的转换
- 处理与物理传输介质相关的接口

- (2) 数据链路层

- 介质访问控制子层(MAC)
  - 解决广播型网络中多用户竞争
- 逻辑链路控制子层(LLC)
  - 链路差错控制、帧校验；链路流量控制





# OSI七层模型的作用

- (3) 网络层

- 主机到主机间的通信
- 路由寻径
- 流量控制、拥塞控制

- (4) 传输层

- 进程间端到端的通信
- 提供传输的可靠性



# OSI七层模型的作用

- (5) 会话层

- 会话建立、撤销
- 传输同步
- 面向连接的交互活动管理：口令认证，数据传输规范

- (6) 表示层

- 信息压缩
- 信息加密、解密
- 与标准格式的转换



# OSI七层模型的作用

- (7) 应用层

- 提供最通用的应用程序（电子邮件、BBS、Web、文件传输等）
- 完成用户信息或者软件转换信息的交互



# 协议分层

- 因为每个堆栈都是独立设计的，协议不能从一个给定的堆栈与另一个协议交互。
- 计算机可以同时运行多个堆栈。

Vendor ( 供应商 )	Stack ( 栈 )
Novell Corporation	Netware
Banyan System Corporation	VINES
Apple Computer Corporation	AppleTalk
Digital Equipment Corporation	DECNET
IBM	SNA
(many vendors)	TCP/IP



计算机网络

T09



谢谢

厦门大学信息学院软件工程系

黄炜 副教授