计算机网络

9. WAN Technologies and Protocol Layering

厦门大学软件学院

黄炜 助理教授

PART II Packet Transmission

Ch 13 WAN Technologies and Routing

广域网技术与路由

13.2 Large Networks and Wide Areas

- A network technology is classified into one of three broad categories:
  - LAN: Local Area Network(局域网)
  - MAN: Metropolitan Area Network(城域网)
  - WAN: Wide Area Network(广域网)

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

11.3 Packet Switches 包交换

- WAN由计算机连接的交换机构成
  - Packet switch: 将分组从一个连接转发到另一个连接
  - 分组交换机是小型计算机
    - high-speed: used to connect the switch to a digital circuit that leads.
    - lower-speed: used to connect the switch to an individual computer.

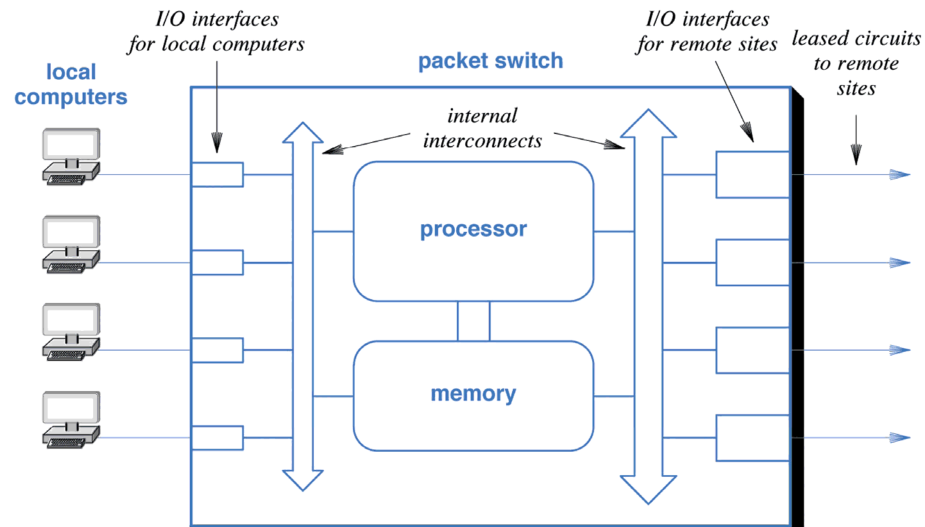


Figure 18.1 Illustration of traditional packet switch architecture.

18.3 Traditional WAN Architecture

- Since the advent of LAN technology, most WANs separate a packet switch into two parts:
  - a Layer 2 switch that connects local computers
  - a router that connects to other sites
- Communication with local computers can be separated from transmission across a WAN

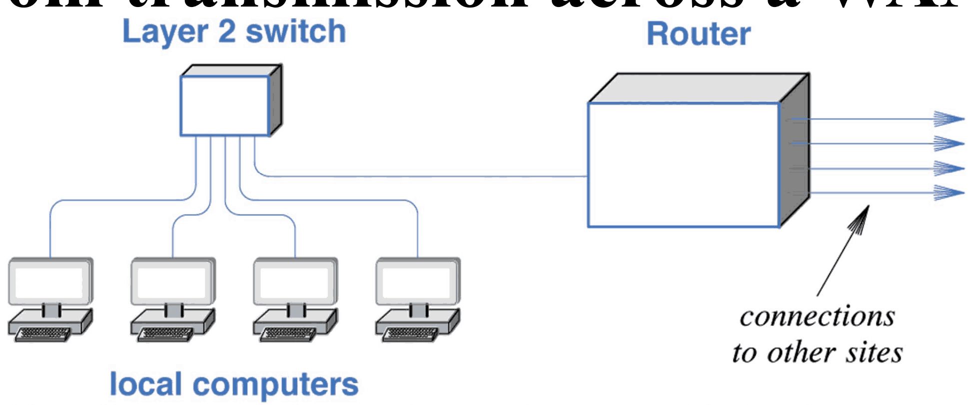


Figure 18.2 Illustration of a modern WAN site with local communication handled by a separate LAN.

- **A WAN can be formed by interconnecting a set of sites**
- **The exact details of the interconnections depend on**
  - **the data rate needed; the distance spanned; and the delay**
- **Many WANs use leased data circuits**
  - **However, other forms are also available**
    - **such as microwave and satellite channels**
- **A network designer must choose a topology**
  - **For a given set of sites, many topologies are possible**

13.4 Forming A WAN

- **A set of packet switches are interconnected to form a Wide Area Network.**
  - **a WAN does not need to be symmetric in the interconnections among packet switches**

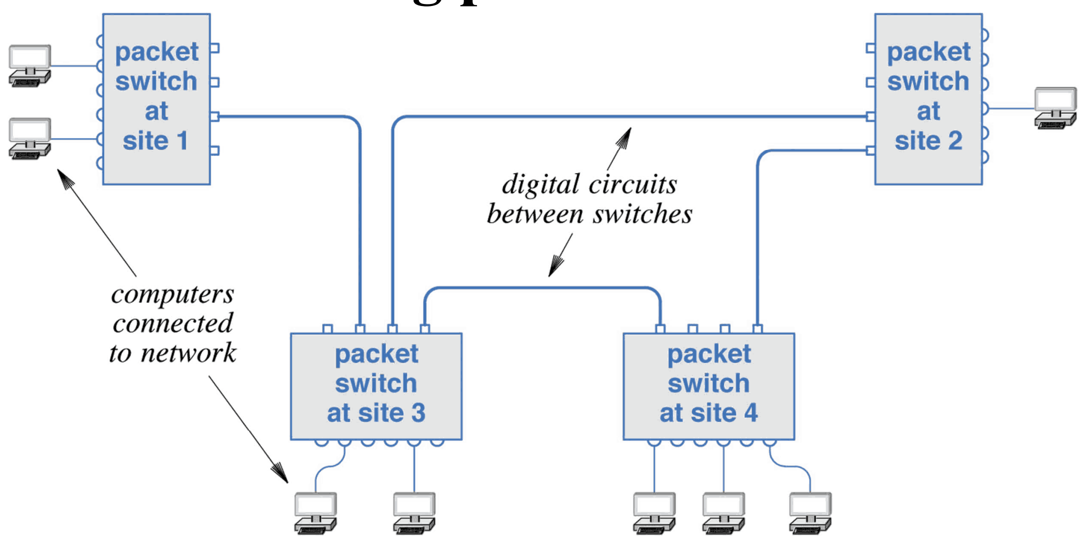


Figure 18.3 An example WAN formed by interconnecting packet switches.

13.5 Store and Forward 存储转发

- The goal of a WAN is to allow as many computers as possible to send packets simultaneously
- The fundamental paradigm to achieve simultaneous transmission is known as store and forward
  - To perform store and forward processing
    - a packet switch buffers packets in memory

13.5 Store and Forward 存储转发

- The store operation occurs when a packet arrives:
  - I/O hardware in the switch places a copy of the packet in memory
- The forward operation occurs once a packet has arrived and is waiting in memory. The processor
  - examines the packet
  - determines its destination
  - and sends the packet over the I / O interface that leads to the destination

13.6 Physical Addressing in A WAN

- **A traditional WAN network operates similar to a LAN**
  - **Each WAN technology defines the exact frame format a computer uses when sending and receiving data.**
  - **Each computer connected to a WAN is assigned a physical address.**
  - **Many WANs use a hierarchical addressing scheme.**

- **WANs addresses follow a key concept that is used in the Internet: hierarchical addressing**
  - **Hierarchical addressing divides each address into two parts:**

    `(site, computer at the site)`
  - **In practice, instead of a identifying a site, each packet switch is assigned a unique number**
    - **first part of an address identifies a packet switch**
    - **second part identifies a specific computer**
    - **A computer connected to port 6 on packet switch 2 is assigned address [2, 6]**

- In practice, an address is represented as a single binary value with some bits of the binary value
  - used to represent a packet switch, and a computer
- In Part 4 of the text, we'll show that each Internet address consists of a binary number, where
  - a prefix of bits identifies a specific network in the Internet
  - the remainder identifies a computer attached to the network

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

13.7 Next-Hop Forwarding

- **A packet switch does not keep complete information about how to reach all possible destinations.**
- **A given switch has information about the next place (hop) to send a packet.**
- **Next-hop information can be organized into a table.**

18.7 Next-Hop Forwarding

- What is the importance of hierarchical addressing?
- When a packet arrives
  - a switch must choose an outgoing path to forward it
- If a packet is destined
  - for a local computer
    - the switch sends the packet directly to the computer
  - Otherwise
    - the packet must be forwarded over to another switch

18.7 Next-Hop Forwarding

- To make the choice, a packet switch
  - examines the destination address in the packet
  - and extracts the packet switch number
    - If the number in the destination address is identical to the packet switch's own ID the packet is intended for a computer on the local packet switch
    - Otherwise, the packet is intended for a computer on another switch

18.7 Next-Hop Forwarding

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 for-
ward
        the packet over the link;
  }
```

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

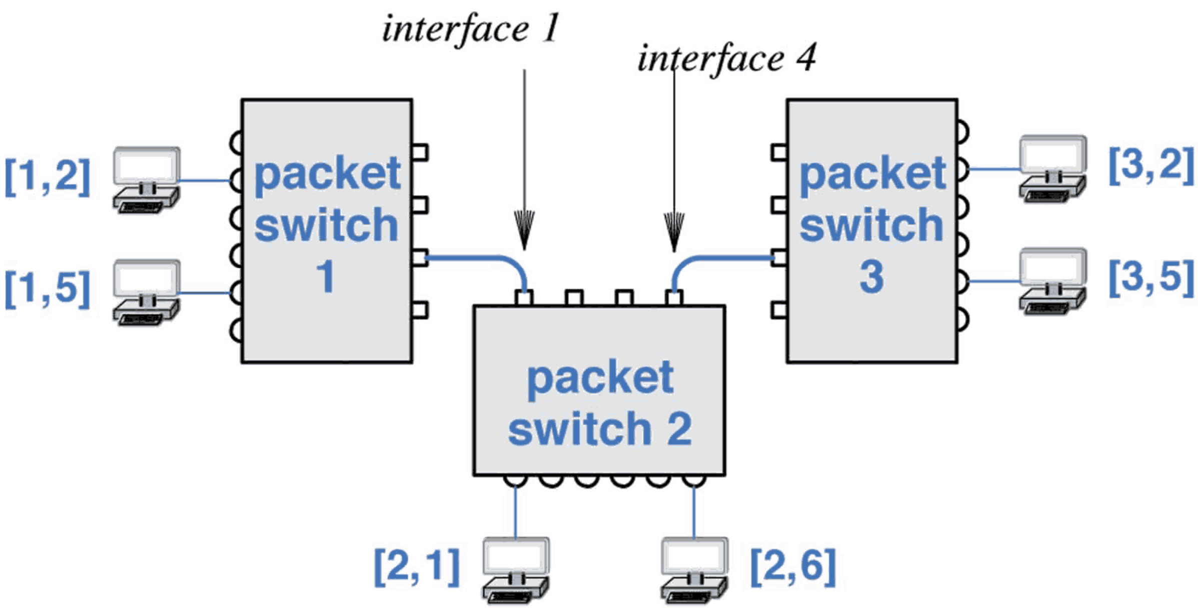
18.7 Next-Hop Forwarding

- A packet switch does not need to keep complete information about how to reach all possible computers
  - nor does a switch need to compute the entire route a packet will follow
  - Instead, a switch bases forwarding on packet switch IDs
    - which means that a switch only needs to know which outgoing link
  - A switch only needs to compute the next hop for a packet

18.7 Next-Hop Forwarding

- The process is called next-hop forwarding
  - and is analogous to the way airlines list flights
- To make the computation efficient
  - switches use table lookup
  - that is, each packet switch contains a forwarding table
  - such tables were originally called routing tables
    - lists all possible packet switches and gives a next hop for each

18.7 Next-Hop Forwarding



(a)

to reach	send to
switch 1	interface 1
switch 2	local delivery
switch 3	interface 4

(b)

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

18.7 Next-Hop Forwarding

- Using only one part of a **two-part hierarchical address** to forward a packet has two practical consequences
  - First, the **computation time** required to forward a packet is reduced because the forwarding table can be organized as an array that uses indexing instead of searching
  - Second, the forwarding table contains one entry per packet switch instead of one entry per destination computer
    - The reduction in table size can be substantial, especially for a large WAN that has many computers attached to each packet switch

18.7 Next-Hop Forwarding

- **A two-part hierarchical addressing scheme allows packet switches to use only the first part of the destination address until the packet reaches the final switch**
  - **Once the packet reaches the final switch**
    - **the switch uses the second part of the address to choose a specific computer**
  - **As Algorithm 18.1 describes**

13.8 Source Independence

- **Next-hop forwarding does not depend on the packet's original source or on the path the packet has taken before it arrives at a particular packet switch**
  - **Instead, the next hop to which a packet is sent depends only on the packet's destination**
  - **The concept is known as source independence**
  - **Source independence allows the forwarding mechanism in a computer network to be compact and efficient**

13.9 Relationship of hierarchical Addresses to Routing

- The table used to store next hop information is commonly called a **routing table**.
- The process of forwarding a packet to its next hop is known as **routing**.
- All destination addresses that have an identical first part will be forwarded to the same packet switch.
- When the packet reaches the switch to which the destination computer attaches, the switch examines the second part of the address and selects the appropriate computer.

18.9 Dynamic Routing Updates in a WAN

- **Values in the routing table must guarantee following:**
  - **Universal communication**
    - **The forwarding table in each switch must contain a valid next-hop route for each possible destination address**
  - **Optimal routes**
    - **In a switch, the next-hop value in the forwarding table for a given destination must point to the shortest path to the destination**

- **Network failures further complicate forwarding**
  - **E.g., if two paths exist to a given destination and one of the paths becomes unavailable because hardware fails, forwarding should be changed to avoid the unavailable path**
  - **A network manager cannot merely configure a forwarding table to contain static values that do not change**
  - **Instead, software running on the packet switches continually tests for failures, and reconfigures the forwarding tables automatically**

Illustration of Graph

- As Figure 18.6 shows, nodes in the graph are given a label that is the same as the number assigned to the corresponding packet switch

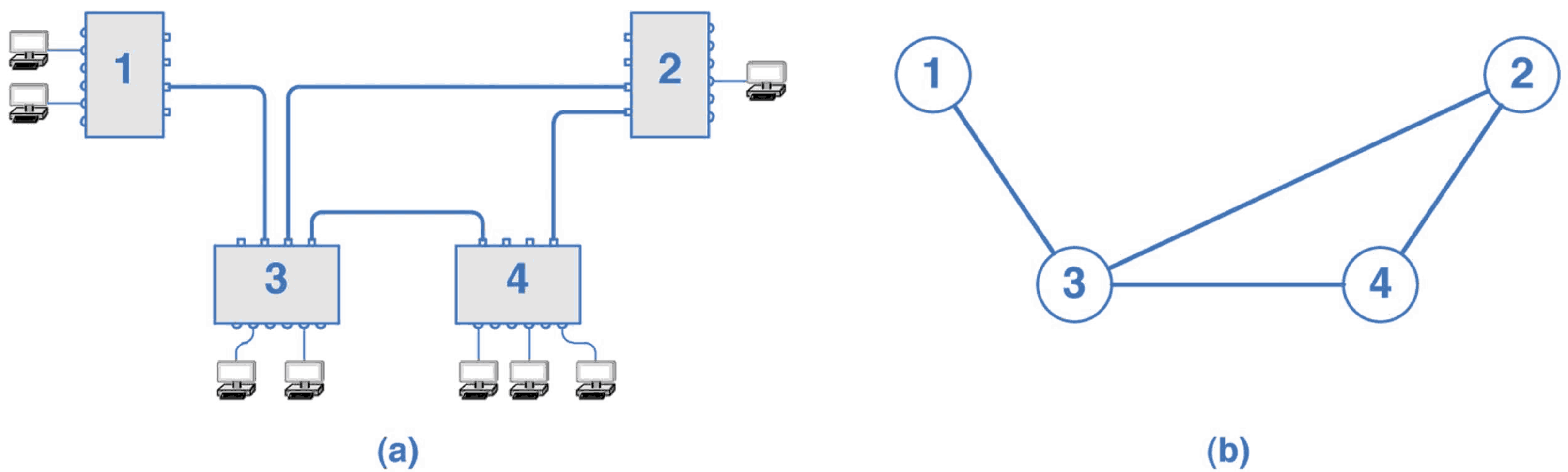


Figure 18.6 Illustration of a WAN and the corresponding graph.

18.9 Dynamic Routing Updates in a WAN

- **A graph representation is useful in computing next-hop forwarding**
  - **because graph theory has been studied and efficient algorithms have been developed**
  - **a graph abstracts away details, allowing routing software to deal with the essence of the problem**
- **When it computes next-hop forwarding for a graph**
  - **a routing algorithm must identify a link**
- **notation (*k*, *j*) to denote a link from node *k* to node *j***
  - **When a routing algorithm runs on the graph in Fig. 18.6b**
  - **The algorithm produces output as shown in Fig. 18.7**

Illustration of Graph

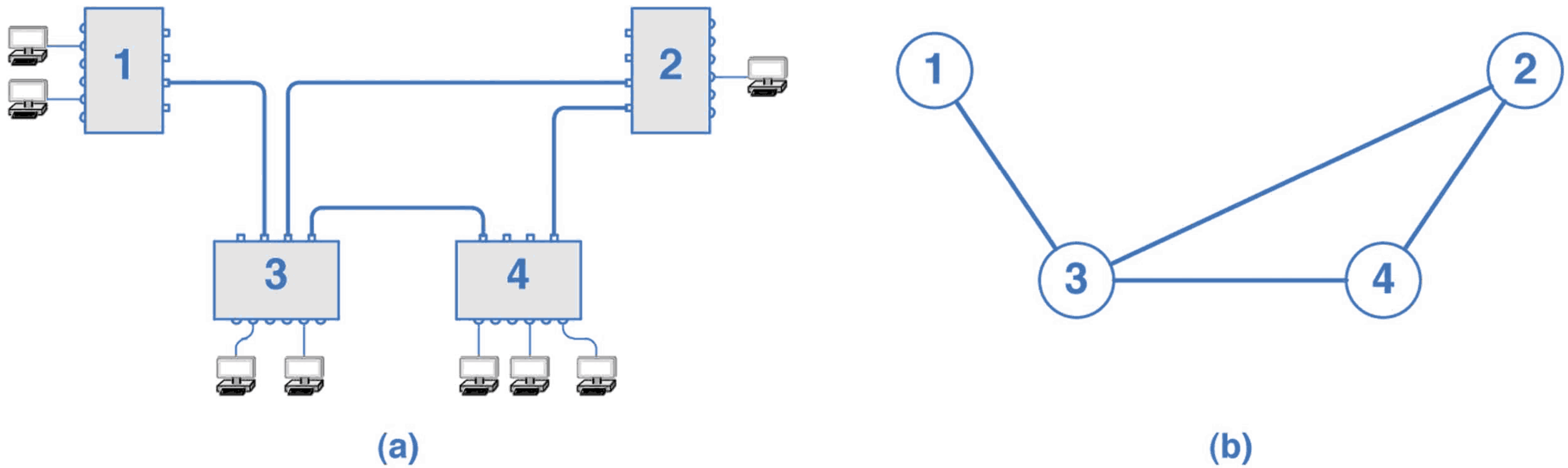


Figure 18.6 Illustration of a WAN and the corresponding graph.

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

node 1

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

node 2

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

node 3

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

node 4

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

13.11 Use of Default Routes

- **The abbreviated routing table still contains many entries with the same next hop.**
- **Most WAN systems include a mechanism that can be used to eliminate the common case of duplication routing.**
- **The forwarding table for node 1 in Fig. 18.7**
  - **a forwarding table may contain many entries that point to the same next hop**

13.11 Use of Default Routes

- **Default route**
  - a mechanism that allows a single entry in a forwarding table to replace a long list of entries having same next-hop value
  - If the forwarding mechanism does not find an explicit entry for a given destination, it uses the default entry.
- **Only one default entry is allowed.**
- Default routing is optional
  - a default entry is present only if more than one destination has the same next-hop value

13.11 Use of Default Routes

node 1

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

node 2

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

node 3

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

node 4

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

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

node 1

to reach	next hop
1	±
*	(1,3)

node 2

to reach	next hop
2	±
4	(2,4)
*	(2,3)

node 3

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

node 4

to reach	next hop
2	(4,2)
4	±
*	(4,3)

Figure 18.8 The forwarding tables from Figure 18.7 with default routes denoted by an asterisk.

13.12 Routing Table Computation

- There are two basic approaches:
  - Static routing: a program computes and installs routes when a packet switch boots; the routes do not change.
  - Dynamic routing: a program builds an initial routing table when a packet switch boots; the program then alters the table as conditions in the network change.

13.12 Routing Table Computation

- **Each approach has advantages and disadvantages**
  - **Advantages of static routing are simplicity and low overhead**
  - **Disadvantage is inflexibility**
    - **static routes cannot be changed when communication is disrupted**
- **Large networks are designed with redundant connections to handle occasional hardware failures**
  - **most WANs use a form of dynamic routing**

18.12 Distributed Route Computation

- **Algorithm 18.2 shows how a forwarding table can be computed**
  - **after information about a network is encoded in a graph**

Algorithm 18.2

```
Given:
   A graph with a nonnegative weight assigned to each edge
   and a designated source node
Compute:
   The shortest distance from the source node to each other
   node and a next-hop routing table
Method:
  Initialize set S to contain all nodes except the source node;
  Initialize array D so that D[v] is the weight of the edge from the
     source to v if such an edge exists, and infinity otherwise;
  Initialize entries of R so that R[v] is assigned v if an
     edge exists from the source to v, and zero otherwise;

  while (set S is not empty) {
     choose a node u from S such that D[u] is minimum;
     if (D[u] is infinity) {
        error: no path exists to nodes in S; quit;
     }
     delete u from set S;
     for each node v such that (u,v) is an edge {
        if (v is still in S) {
           c = D[u] + weight(u,v);
           if (c < D[v])  {
              R[v] = R[u];
              D[v] = c;
           }
        }
     }
  }
```

18.12 Distributed Route Computation

- In practice, WANs need to perform distributed **route computation**
  - Instead of a centralized program computing all shortest paths, each packet switch must compute its own forwarding table locally
  - All packet switches must participate in distributed route computation

18.12 Distributed Route Computation

- There are two general forms:
  - Link-State Routing (LSR), which uses Dijkstra's algorithm
  - Distance-Vector Routing (DVR), which uses another approach

Dijkstra's Algorithm

数据结构课程

18.12.1 Link-State Routing (LSR)

- **Link-state routing** or **Link-status routing**
  - the approach known as **Shortest Path First (SPF)** routing
  - Dijkstra algorithm used it to characterize the way it works
    - actually all routing algorithms find shortest paths
- packet switches periodically send messages across the network that carry the status of a link
  - For example, packet switches 5 and 9 measure the link between them and send a status message
    - such as "the link between 5 and 9 is up"
  - Each status message is broadcast to all switches

18.12.1 Link-State Routing (LSR)

- **Every switch collects incoming status messages**
  - **and uses them to build a graph of the network**
- **Each switch then uses Algorithm 18.2 to produce a forwarding table by choosing itself as the source**
- **An LSR algorithm can adapt to hardware failures**
- **If a link between packet switches fails**
  - **the attached packet switches will detect the failure**
  - **broadcast a status message that specifies the link is down**

18.12.1 Link-State Routing (LSR)

- All packet switches receive the broadcast
  - change their copy of the graph to reflect the change in the link's status
  - and re-compute shortest paths
- Similarly, when a link becomes available again
  - the packet switches connected to the link detect that it is working
  - and start sending status messages that report its availability

18.12.2 Distance Vector Routing (DVR)

- As with LSR, each link in the network is assigned a weight
- The distance to a destination between two packet switches is defined to be the sum of weights along the path between the two
- Like LSR, DVR arranges for packet switches to exchange messages periodically

18.12.2 Distance Vector Routing (DVR)

- **In DVR, a switch sends a complete list of destinations and the current cost of reaching each**
- **When it sends a DVR message**
  - **a switch sends a series of individual statements, of the form**
    - **"I can reach destination X, and its current distance from me is Y"**
- **DVR messages are not broadcast**
  - **Each switch periodically sends a DVR message to neighbors**
- **Each message contains pairs of (destination, distance)**

18.12.2 Distance Vector Routing (DVR)

- Each packet switch must keep a list of possible destinations
  - along with the current distance to the destination and the next hop to use
  - the list of destinations and the next hop for each can be found in the forwarding table
- DVR software can be considered as maintaining an extension to the forwarding table
  - that stores a distance for each destination

18.12.2 Distance Vector Routing (DVR)

- When a message arrives at a switch from neighbor N
  - the switch examines each item in the message
  - changes its forwarding table if the neighbor has a shorter path to some destination than the path currently being used
- Example:
  - if neighbor N advertises a path to dest. D as having cost 5 and the current path through neighbor K has cost 100
    - the current next hop for D will be replaced by N
    - and the cost to reach D will be 5 plus the cost to reach N

18.12.2 DVR

- **Algorithm 18.3 specifies how routes are updated when using the distance-vector approach**

Algorithm 18.3

Given:

A local forwarding table with a distance for each entry, a distance to reach each neighbor, and an incoming DV message from a neighbor

Compute:

An updated forwarding table

Method:

```
Maintain a distance field in each forwarding table entry;
Initialize forwarding table with a single entry that has the
   destination equal to the local packet switch, the
   next-hop unused, and the distance set to zero;

Repeat forever {
   Wait for a routing message to arrive over the network
      from a neighbor;  let the sender be switch N;
   for each entry in the message {
      Let V be the destination in the entry and let D
         be the distance;
      Compute C as D plus the weight assigned to the
         link over which the message arrived;
      Examine and update the local routing table:
      if (no route exists to V) {
         add an entry to the local routing table for destination
           V with next-hop N and distance C;
      } else if (a route exists that has next-hop N) {
           replace the distance in existing route with C;
      } else if (a route exists with distance greater than C) {
           change the next-hop to N and distance to C;
      }
}
```

18.13 Shortest Path Computation in a Graph

- **Once a graph has been created that corresponds to a network**
  - **software uses a method known as Dijkstra's Algorithm**

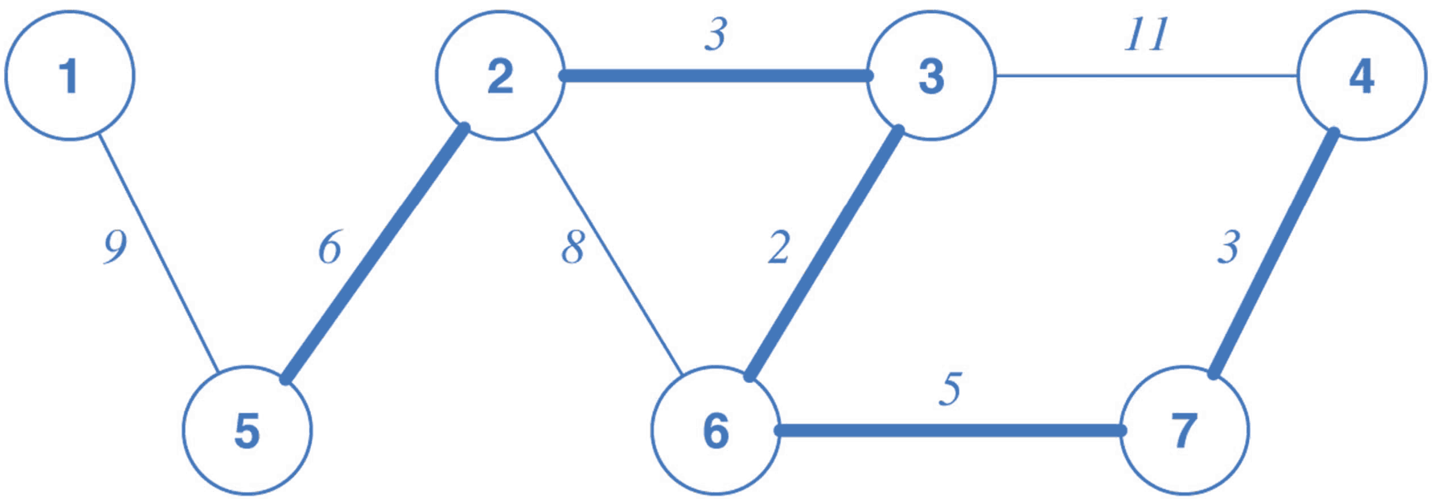


Figure 18.9 An example graph with a weight assigned to each edge and a shortest path between nodes *4* and *5* shown darkened.

18.13 Shortest Path Computation in a Graph

- **To find the shortest path from a source node to each of the other nodes in the graph:**
  - **a next-hop forwarding table is constructed during the computation of shortest paths**
  - **The algorithm must be run once for each node in the graph**
  - **That is, to compute the forwarding table for packet switch P**
    - **the node that corresponds to P is designated as the source node**
    - **and the algorithm is run**

18.14 Routing Problems

- In theory, either LSR or DVR routing will compute shortest paths
- Furthermore, each approach will eventually converge
  - means the forwarding tables in all packet switches agree
- However, problems do occur
  - For example, if LSR messages are lost, two packet switches can disagree about the shortest path
- DVR problems can be more severe

18.14 Routing Problems

- One of the primary reasons DVR protocols exhibit problems comes from backwash
  - (i.e., a packet switch receives information that it sent)
  - For example, suppose a switch tells its neighbors
    - "I can reach destination D at cost 3"
    - If the connection leading to destination D fails
      - the switch will remove the entry for D from its forwarding table
      - (or mark the entry invalid)
    - But the switch has told neighbors that a route exists

18.14 Routing Problems

- Imagine that just after the link fails
  - one of the neighbors sends a DVR message that specifies
  - “I can reach destination D at cost 4”
- Unfortunately
  - the message will be believed
  - and a routing loop will be created

18.14 Routing Problems

- **Most practical routing mech. contain constraints and heuristics to prevent problems like routing loops**
  - For example, DVR schemes employ split horizon
    - which specifies that a switch does not send information back to its origin
  - most practical routing systems introduce hysteresis
    - prevents the software from making many changes in a short time
  - However, in a large network where many links fail and recover frequently, routing problems can occur

PART II Packet Transmission

Ch 16 Protocols and Layering

协议与分层

16.2 The Need for Protocols

- All parties involved in a communication must agree on a set of rules to be used when exchanging messages.
- such an agreement is called a protocol (协议).
- Network protocol (网络协议): a set of rules that specify the format of messages and the appropriate action required for each message .
- The software that implements such rules is called protocol software (协议软件).

16.3 Protocol Suites 协议系列

- To divide the communication problem into subpieces and to design a separate protocol for each subpiece.
- Doing so make each protocol easier to design, analyze, implement, and test.
- Each protocol should handle part of the communication problem not handled by other protocols.
- Protocol should can share data structures and information.

16.3 Protocol Suites 协议系列

- The combination of protocols should handle all possible hardware failures or other exceptional condition.
- Protocol are designed and developed in complete, cooperative sets called suites (系列) or families (族) .
- Each protocol in a suite solves one part of the communication problem; together, they solve the entire communication problem.

16.4 A Plan for Protocol Design

- **Layering model (分层模型).**
- **The communication problem can be divided into subpieces, called layers (层) .**
- **A protocol suite can be designed by specifying a protocol that corresponds to each layer.**
- **A layering model is a tool to help protocol designers construct a suite of protocols that solves all communication problems.**

1.6 Protocol Suites and Layering Models

- A set of protocols must be constructed
  - to ensure that the resulting communication system is complete and efficient
- Each protocol should handle a part of communication not handled by other protocols
- How can we guarantee that protocols work well together?
  - Instead of creating each protocol in isolation, protocols are designed in complete, cooperative sets called suites or families

1.6 Protocol Suites and Layering Models

- **Each protocol in a suite handles one aspect of networking**
  - **The protocols in a suite cover all aspects of communication**
  - **The entire suite is designed to allow the protocols to work together efficiently**
- **The fundamental abstraction used to collect protocols into a unified whole is known as a layering model**
- **All aspects of a communication problem can be partitioned into pieces that work together**
  - **each piece is known as a layer**

1.6 Protocol Suites and Layering Models

- **Dividing protocols into layers helps both protocol designers and implementers manage the complexity**
  - to concentrate on one aspect of communication at a given time
- **Later chapters will help us understand layering**
  - by explaining protocols in detail
- **For now, it is sufficient to learn the purpose of each layer and how protocols are used for communication**

TCP/IP v.s. OSI Layers

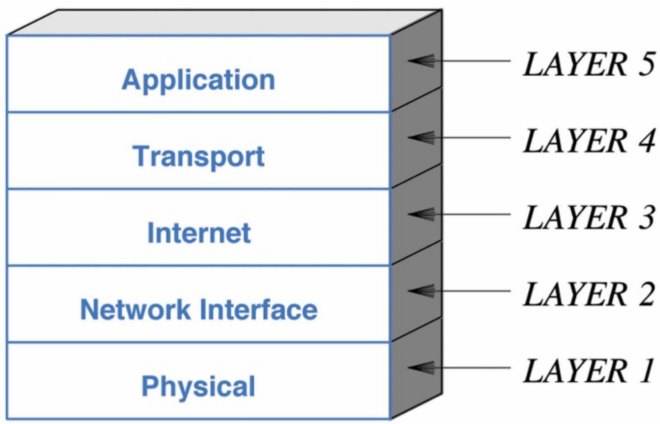


Figure 1.1 The layering model used with the Internet protocols (TCP/IP).

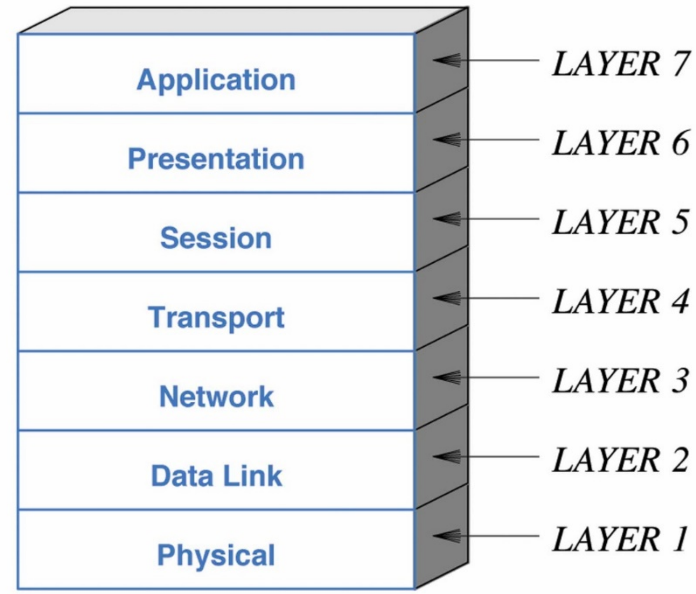


Figure 1.4 The OSI seven-layer model standardized by ISO.

1.6 Protocol Suites and Layering Models

- **Physical Layer (Layer 1)**
  - specify details about the underlying transmission medium and hardware
  - all specifications related to electrical properties, radio frequencies, and signals

1.6 Protocol Suites and Layering Models

- **Network Interface Layer (Layer 2)**
  - some publications use the term **Data Link**
  - specify details about communication between higher layers of protocols (softwr.) and the underlying network (hardwr.)
  - specifications about
    - network **addresses**
    - maximum **packet** size that a network can support
    - protocols used to access the underlying **medium**
    - and hardware addressing

1.6 Protocol Suites and Layering Models

- **Internet Layer (Layer 3)**
  - Protocols in the Internet layer form the fundamental basis for the Internet
  - Layer 3 protocols specify communication across the Internet (spanning multiple interconnected networks)

1.6 Protocol Suites and Layering Models

- **Transport Layer (Layer 4)**
  - Provide for communication from an application program on one computer to an application program on another
  - Includes specifications on
    - controlling the maximum rate a receiver can accept data
    - mechanisms to avoid network congestion (拥塞)
    - techniques to insure that all data is received in the correct order

1.6 Protocol Suites and Layering Models

- **Application Layer (Layer 5)**
  - specify how a pair of applications interact when they communicate
  - specify details about
    - the format and
    - the meaning of messages that applications can exchange
    - the procedures to be followed
  - Some examples of network applications in layer 5
    - email exchange; file transfer; web browsing; telephone services; and video teleconferencing

1.7 How Data Passes Through Layers

- **Protocol implementations follow the layering model**
  - by passing the output from a protocol in one layer to the input of a protocol in the next
- **To achieve efficiency**
  - rather than copy an entire packet
  - a pair of protocols in adjacent layers pass a pointer to the packet

1.7 How Data Passes Through Layers

- Fig. 1.2 illustrates layered protocols on the two compt.
  - Each computer contains a set of layered protocols
  - When an application sends data
    - it is placed in a packet, and the packet passes down through each layer of protocols
  - Once it has passed through all layers on the sender
    - the packet leaves the computer and is transmitted across the physical network
  - When it reaches the receiver
    - the packet passes up through the layers of protocols
  - If the application on the receiver sends a response, the process is reversed

1.7 How Data Passes Through Layers

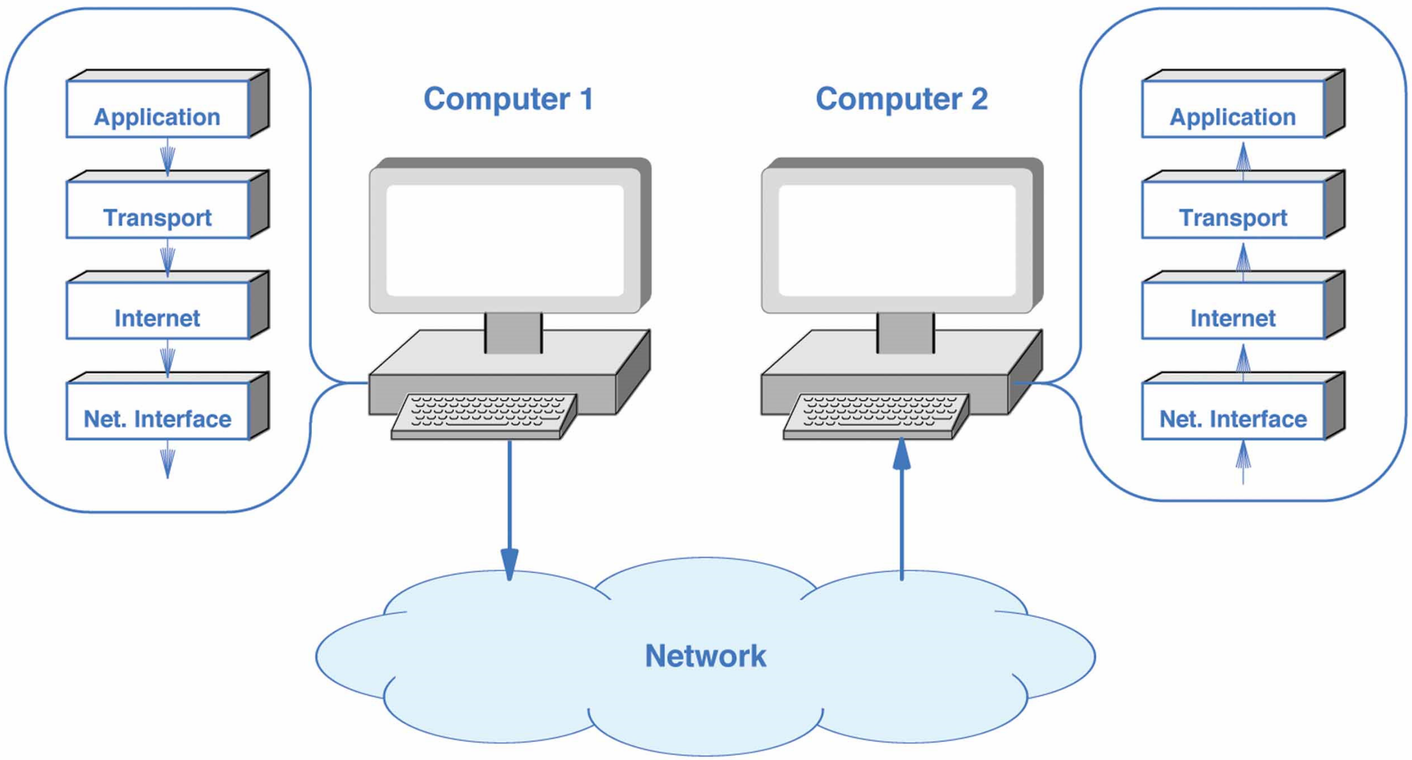


Figure 1.2 Illustration of how data passes among protocol layers when computers communicate across a network. Each computer has a set of layered protocols, and data passes through each layer.

1.8 Headers and Layers

- Each layer of protocol softwr. performs computations
  - that insure the messages arrive as expected
- To perform such computation, protocol software on the two machines must exchange information
  - each layer on the sender prepends extra information onto the packet
  - the corresponding protocol layer on the receiver removes and uses the extra information

1.8 Headers and Layers

- **Header: Additional info. added by a protocol**
  - **Headers are added by protocol software on the sending computer**
    - **That is, the Transport layer prepends a header, and then the Internet layer prepends a header, and so on**
  - **If we observe a packet traversing the network, the headers will appear in the order that Fig. 1.3 illustrates**
  - **Although the figure shows headers as the same size**
    - **in practice headers are not of uniform size**
    - **and a physical layer header is optional**

1.8 Headers and Layers

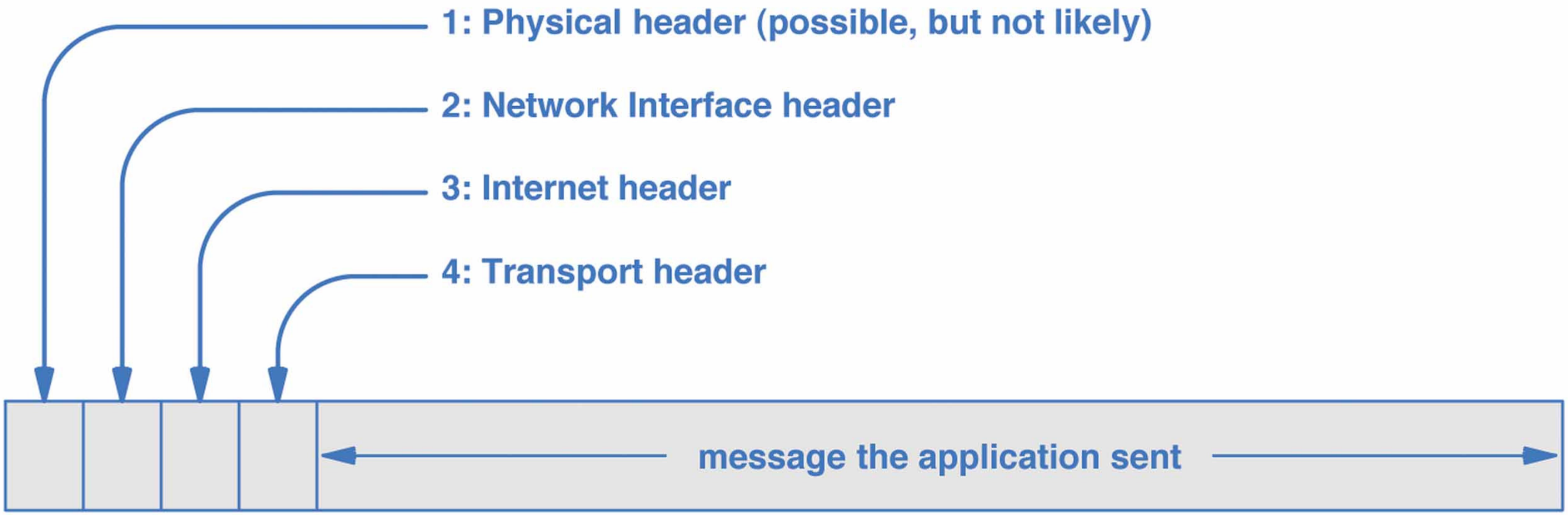


Figure 1.3 The nested protocol headers that appear on a packet as the packet travels across a network between two computers. In the diagram, the beginning of the packet (the first bit sent over the underlying network) is shown on the left.

1.9 ISO and the OSI Seven-Layer Reference Model

- **OSI Model**
  - the same time the Internet protocols were being developed
  - standards bodies jointly formed an alternative ref. model
    - International Standardization Organization (ISO)
    - Int'l Telecommunications Union, Telecommunication (ITU-T)
- **The ISO layering model is known as the Open Systems Interconnection (OSI) Seven-Layer Reference Model**
- **Figure 1.4 illustrates the seven layers in the model**

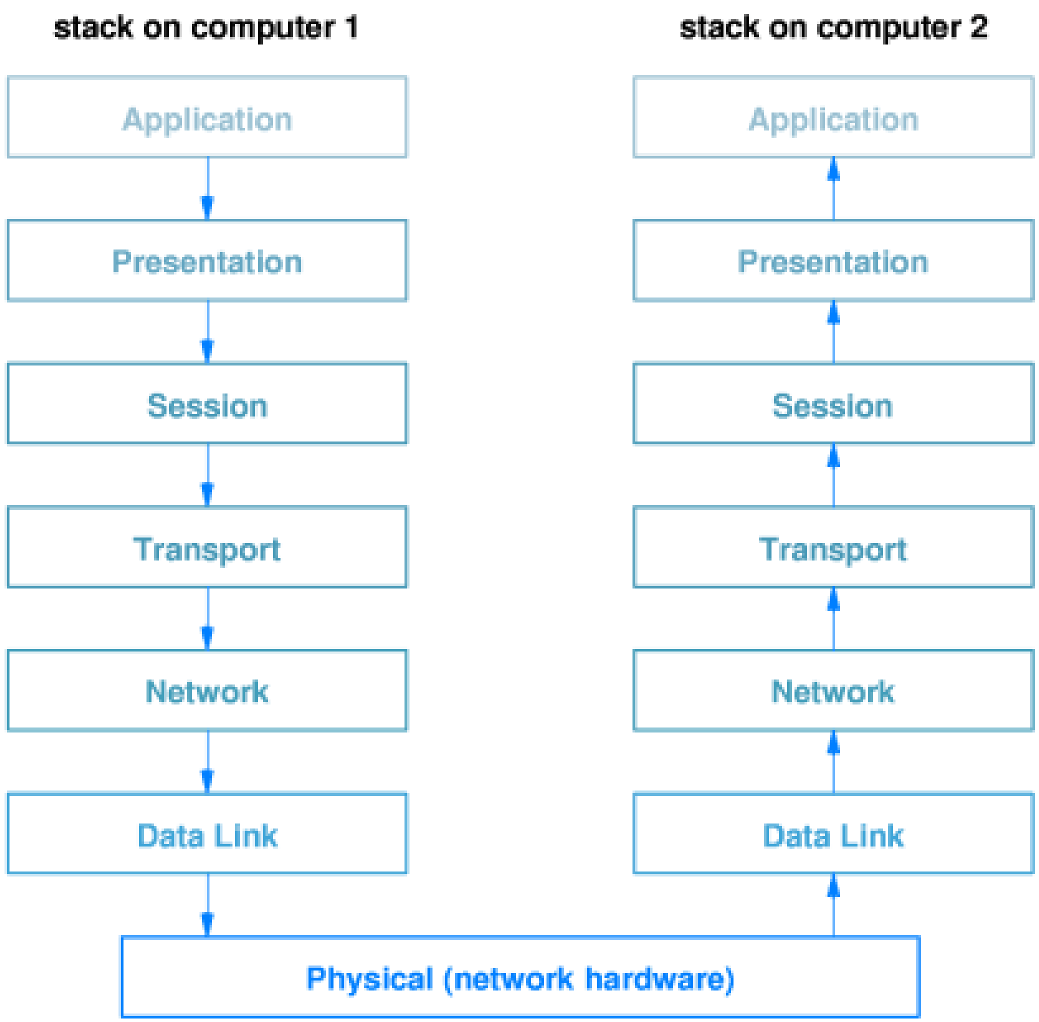
OSI七层模型

- **OSI Model**

	Data unit	Layer	Function
Host layers	Data	7. Application	Network process to application
		6. Presentation	Data representation, encryption and decryption, convert machine dependent data to machine independent data
		5. Session	Interhost communication, managing sessions between applications
	Segments	4. Transport	End-to-end connections, reliability and flow control
Media layers	Packet/ Datagram	3. Network	Path determination and logical addressing
	Frame	2. Data link	Physical addressing
	Bit	1. Physical	Media, signal and binary transmission

OSI七层模型

7
经理
书写信件大意
读取信件
经理
应用层
6
助理
润色翻译
翻译
助理
表示层
5
秘书
查找通信录书写地址
拆信存档
秘书
会话层
4
快递员
收件到快递公司
送件到公司
快递员
传输层
3
分拣员
分拣邮件
分拣邮件
分拣员
网络层
2
包装员
查找地址
查找地址
包装员
数据链路层
1
搬运工
运输工具
搬运工
物理层
公司
邮局


stack on computer 1
stack on computer 2
Application
Presentation
Session
Transport
Network
Data Link
Application
Presentation
Session
Transport
Network
Data Link
Physical (network hardware)

物理层

- 相关层
  – 硬件设备
  – 链路层
- 作用
  – 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

链路层

- 相关层
  - 物理层
  - 网络层
- 作用
  - 分为介质访问控制子层(MAC)和逻辑链路控制子层(LLC)
  - MAC子层
    - 解决广播型网络中多用户竞争
  - LLC子层
    - 链路差错控制、帧校验
    - 链路流量控制

网络层

- 相关层
  - 数据链路层
  - 传输层
- 作用
  - 负责点到点通信
  - 寻径
  - 流量控制
  - 拥塞控制

传输层

- 相关层
  - 网络层
  - 会话层
- 作用
  - 给上层提供通用的应用界面
  - 屏蔽网络层带来的数据表示结构的差异，实现端到端通信
  - 提供进程间通信机制
  - 提供传输的可靠性

会话层

- 相关层
  - 传输层
  - 表示层
- 作用
  - 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

表示层

- 相关层
  - 会话层
  - 应用层
- 作用
  - 信息压缩
  - 信息加密、解密
  - 与标准格式的转换

应用层

- 相关层
  - 表示层
  - 用户或者上层应用系统
- 作用
  - 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

- **Because each stack has been designed independently.**
- **Protocol from a given stack cannot interact with protocols from another.**
- **A computer can run more than one stack at the same time.**

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

16.7 How Layered Software Works

- Software in a given layer on the sending computer adds information to the outgoing data.
- Software in the same layer on the receiving computer uses the additional information to process incoming data.

16.8 Multiplex, Nested Headers

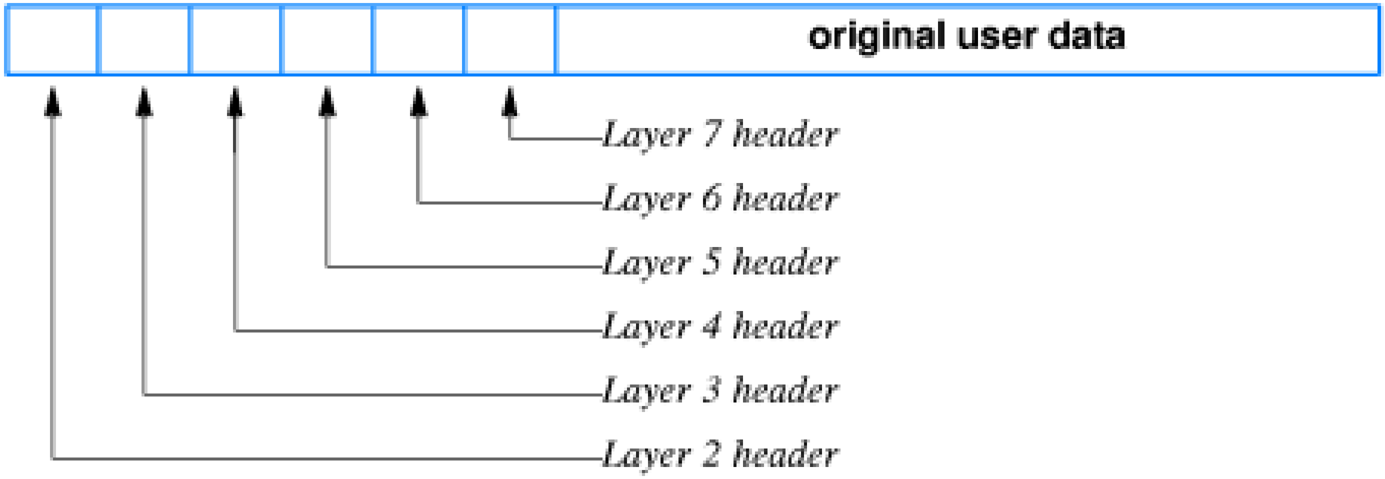
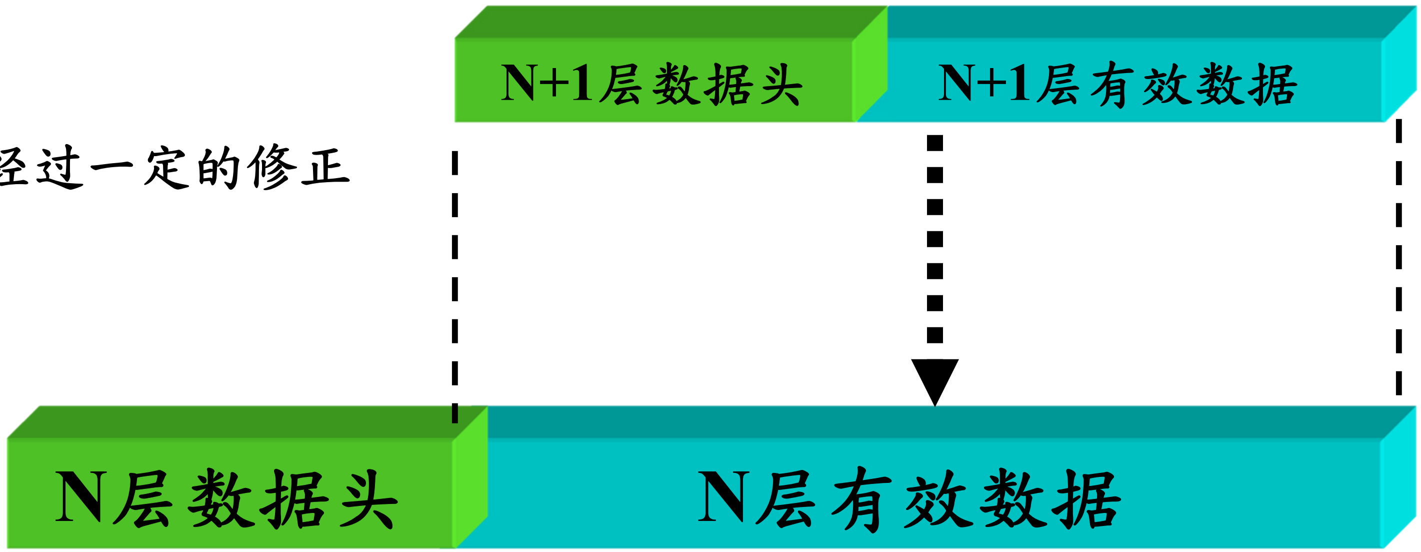


Figure 16.4 The nested protocol headers that appear in a frame as the frame travels across a network. Each layer of protocol software adds a header to an outgoing frame.

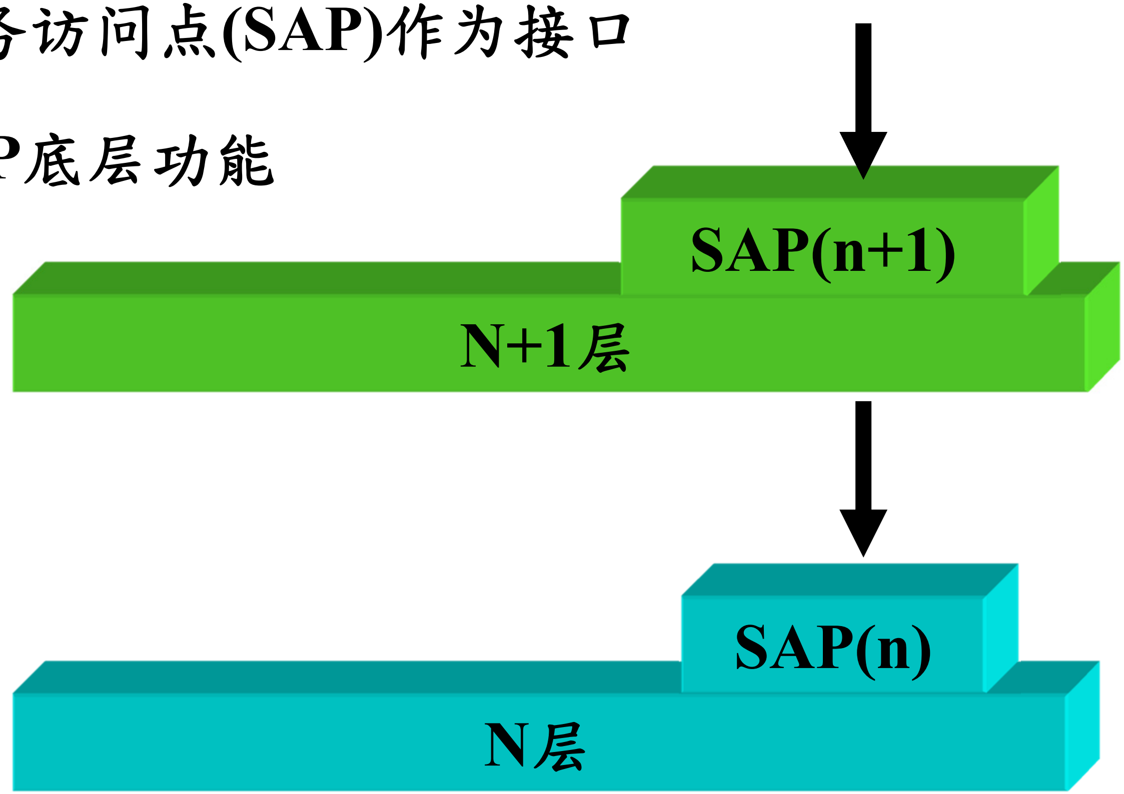
ISO-OSI层次模型的软件功能(1)

- 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



ISO-OSI层次模型的软件功能(2)

- 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



ISO-OSI的优势与问题

- 优势
  - 结构清晰
  - 软件重用
  - 规格统一，减少冲突
- 问题
  - 高层很难对低层通信进行优化
  - 层次过多、过细，软件要求复杂
  - 传输效率低

The True Story of Network Layering And The Origin of The OSI Model

- Despite the success of the Internet in replacing all the previous attempts to build a global computer communication system, some engineers haven't updated their thinking from old descriptions of networks. instead they still cling to the outdated 7-layer reference model that was invented by ISO instead of the 5-layer reference model that was invented for the Internet. Interestingly, engineers who insist on using the older model cannot identify single protocol at layers 5 and 6. But… because they learned the model somewhere in school, they desperately hope that the extra layers must be useful. As a result, they incorrectly classify applications as layer 7 instead of layer 5.

The True Story of Network Layering And The Origin of The OSI Model

- Researchers have begun looking into the origins and uses of the OSI 7-layer reference model to determine why a cumbersome and inaccurate model has had such staying power. They have recently uncovered some surprising facts. We have know for a long time that the model was the work of a group. We did not know, however, that the group met late one night in a bar and began making fun of American pop culture. As it turns out, they started scribbling names of the seven dwarfs from the Disney movie on a paper cocktail napkin, and somebody joked that seven was a really good number for network layers. The next morning at the standards committee meeting, the group passed around the cocktail napkin and generally agreed that that they had discovered something fundamental the previous night while they were drunk. By the end of the day, they had renamed the seven layers (with names that sounded more scientific), and produced the basic model. Here's the lineup and a bit of explanation:

The True Story of Network Layering And The Origin of The OSI Model

Layer	Dwarf	Name	Explanations
1	Sleepy	Physical	The group new that physical connections are boring, and figured it might as well assign the physical layer to dwarf ``Sleepy''. As it happens, a Layer 1 protocol specification does indeed put everyone to sleep (just try reading one late at night).
2	Sneezy	Link	If you monitor a network and watch the pattern of packets emitted by a computer, you'll immediately understand the relationship between link-layer protocols and ``Sneezy''.
3	Happy	Network	Everyone's happy with the network layer. Well... to be honest, the only network layer protocol that makes everyone's happy is the Internet Protocol. Unfortunately, the Internet protocol isn't part of OSI, and wasn't really built to follow the OSI model (the model didn't include internetworking). But, the designers had good intentions.
4	Doc	Transport	This one's obvious -- it definitely takes a Ph.D. to understand the subtleties of a transport layer protocol.

The True Story of Network Layering And The Origin of The OSI Model

Layer	Dwarf	Name	Explanations
5	Dopey	Session	Yep, even the designers realized that having a separate session layer is a dopey idea. They decided to follow Disney's approach of adding comic relief, so they stuck in a completely unnecessary layer and laughed about it.
6	Bashful	Presentation	Another little joke. The designers realized that sooner or later someone would create a presentation layer protocol. However, the group decided to classify such protocols as too ``bashful'' to appear in public. So, even if a presentation protocol is produced, no one gets to see it.
7	Grumpy	Application	Programmers who design network applications are incredibly grumpy -- they complain about the efficiency of other layers, the fundamental abstractions of the network, the long hours, the difficulty of debugging, and the API they are forced to use. And users add to the grumpiness because users never complain about protocols at other layers; they only complain about applications.

The True Story of Network Layering And The Origin of The OSI Model

- Moral of the story: If you're an engineer working on a standards committee, avoid drinking with colleagues -- a bad joke you hatch late one night in the bar could turn out to haunt the industry for decades.

9.

THANK YOU.

厦门大学软件学院

黄炜 助理教授