第一章
计算机网络基本原理与机制
提纲

◆ 网络基本概念
◆ Internet简介
◆ 协议分层和OSI参考模型
◆ End-to-End Arguments
◆ Internet设计原则
提纲

◆ 网络基本概念
◆ Internet简介
◆ 协议分层和OSI参考模型
◆ End-to-End Arguments
◆ Internet设计原则
“Cool” Internet appliances

IP picture frame
http://www.ceiva.com/

World’s smallest web server
http://www-ccs.cs.umass.edu/~shri/iPic.html

Web-enabled toaster + weather forecaster
What’s a network

◆ **network edge**: millions of end-system devices:
  - pc’s workstations, servers
  - PDA’s, phones, toasters running network apps
◆ **network core**: routers, switches forwarding data
  - **packets**: packet switching
  - **calls**: circuit switching
◆ **communication links**
  - fiber, copper, radio, ...
What’s a protocol?

A human protocol and a computer network protocol:

- **Human Protocol**
  - Hi
  - Hi
  - Got the time?
  - 2:00

- **Computer Network Protocol**
  - TCP connection req.
  - TCP connection reply.
  - GET http://www.tsinghua.edu.cn
  - <file>
What's a protocol?

**human protocols:**
- “what's the time?”
- “I have a question”
- Introductions

**network protocols:**
- machines rather than humans
- all communication activity in Internet governed by protocols

... specific msgs sent

... specific actions taken when msgs received, or other events
What’s a protocol?

- 协议的组成:
  - 语法 (syntax): 以二进制形式表示的命令和相应的结构
  - 语义 (semantics): 命令请求、完成的动作和回送的响应的具体含义
  - 定时关系 (timing): 有关事件顺序的说明

- protocols define **format, order of msgs** sent and received among network entities, and **actions** taken on msg transmission, receipt
A closer look at network structure

**network edge:**
- applications and hosts

**network core:**
- routers
- network of networks

**access networks, physical media:**
- communication links
The network edge

- **end systems (hosts)**
  - run application programs
  - e.g., WWW, email
  - at “edge of network”

- **client/server model**
  - client host requests, receives service from server
  - e.g., WWW client (browser)/ server; email client/server

- **peer-peer model**
  - host interaction symmetric
  - e.g.: BT, Maze, e-Donkey
The network core

- **mesh of interconnected routers**
- **the fundamental question:** how is data transferred through net?
  - **circuit switching:** dedicated circuit per call: telephone net
  - **packet switching:** data sent thru net in discrete “chunks”
Communication networks can be classified based on the way in which the nodes exchange information:

- **Communication Network**
  - **Switched Communication Network**
    - **Circuit-Switched Communication Network**
    - **Packet-Switched Communication Network**
  - **Broadcast Communication Network**
    - **Datagram Network**
    - **Virtual Circuit Network**
Broadcast vs. Switched Communication Networks

- **Broadcast communication networks**
  - Information transmitted by one node is received by every other node in the network
    - E.g., Satellite Network, CCTV-Net
  - **Problem:** coordinate the access of all nodes to the shared communication medium (Multiple Access Problem)

- **Switched communication networks**
  - Information is transmitted to a sub-set of designated nodes
    - E.g., WANs (Telephony Network, Internet)
  - **Problem:** how to forward information to intended node(s)
    - Done by special nodes (e.g., routers, switches) running routing protocols
A Taxonomy of Communication Networks

Communication networks can be classified based on the way in which the nodes exchange information:

- Circuit-Switched Communication Network
- Switched Communication Network
- Packet-Switched Communication Network
- Broadcast Communication Network
- Datagram Network
- Virtual Circuit Network
Circuit Switching

◆ Three phases
  1. circuit establishment
  2. data transfer
  3. circuit termination
◆ If circuit not available: “Busy signal”
◆ Examples
  – Telephone networks
  – ISDN (Integrated Services Digital Networks)
Timing in Circuit Switching

- Circuit Establishment
- Data Transmission
- Circuit Termination

Host 1 Host 2
Node 1 Node 2

propagation delay between Host 1 and Node 1
propagation delay between Host 2 and Node 1
processing delay at Node 1

DATA
Circuit Switching

*A node (switch) in a circuit switching network*
Circuit Switching: Multiplexing/Demultiplexing

- Time divided in **frames** and frames divided in **slots**
- Relative slot position inside a frame **determines** which conversation the data belongs to
- Needs **synchronization** between sender and receiver
- In case of non-permanent conversations
  - Needs to **dynamic** bind a slot to a conservation
Communication networks can be classified based on the way in which the nodes exchange information:

- Communication Network
  - Switched Communication Network
  - Circuit-Switched Communication Network
  - Packet-Switched Communication Network
  - Datagram Network
  - Virtual Circuit Network

- Broadcast Communication Network
Packet Switching

- Data are sent as formatted bit-sequences, so-called **packets**
- Packets have the following structure:

  ![Diagram showing packet structure]

  - **Header** and **Trailer** carry control information (e.g., destination address, checksum)
  - Each packet is passed through the network from node to node along some path (Routing)
  - At each node the entire packet is received, stored briefly, and then forwarded to the next node (**Store-and-Forward** Networks)
Packet Switching

A node in a packet switching network
Packet Switching:
Multiplexing/Demultiplexing

Data can be transmitted at any given time
How to tell them apart?
- Use meta-data (header) to describe data
Communication networks can be classified based on the way in which the nodes exchange information:

- Switched Communication Network
- Circuit-Switched Communication Network
- Packet-Switched Communication Network
- Datagram Network
- Virtual Circuit Network
- Broadcast Communication Network
Datagram Packet Switching

✓ Each packet is independently switched
  – Each packet header contains destination address
✓ No resources are pre-allocated (reserved) in advance
✓ Example: IP networks
Datagram Packet Switching
How do loss and delay occur?

- **router buffers is full**
- **packet arrival rate to link exceeds output link capacity**

Diagram:
- Output link (loss)
- Dropped (loss) if no free buffers
Four sources of packet delay

1. nodal processing
   - check bit errors
   - determine output link

2. queueing
   - time waiting at output link for transmission
   - depends on congestion level of router
Four sources of packet delay

3. Transmission delay:
   - $R=$ link bandwidth (bps)
   - $L=$ packet length (bits)
   - time to send bits into link = $L/R$

4. Propagation delay:
   - $d =$ length of physical link
   - $s =$ propagation speed in medium ($\sim 2 \times 10^8$ m/sec)
   - propagation delay = $d/s$

Note: $s$ and $R$ are very different quantities!
Communication networks can be classified based on the way in which the nodes exchange information:

- **Switched Communication Network**
  - Circuit-Switched Communication Network
  - Packet-Switched Communication Network
- **Broadcast Communication Network**
- **Datagram Network**
- **Virtual Circuit Network**
Virtual-Circuit Packet Switching

- Hybrid of circuit switching and packet switching
  - Data is transmitted as packets
  - All packets from one packet stream are sent along a pre-established path (=virtual circuit)
- Guarantees in-sequence delivery of packets
- However: Packets from different virtual circuits may be interleaved
- Example: ATM networks
Virtual-Circuit Packet Switching

- Communication with virtual circuits takes place in three phases
  1. VC establishment
  2. Data transfer
  3. VC disconnect

- **Note**: packet headers don’t need to contain the full destination address of the packet
Timing of Virtual-Circuit Packet Switching

Host 1

Node 1

Node 2

Host 2

VC establishment

Packet 1
Packet 2
Packet 3

Data transfer

Packet 1
Packet 2
Packet 3

Packet 1
Packet 2
Packet 3

VC termination

propagation delay between Host 1 and Node 1
Virtual-Circuit Packet Switching
Packet-Switching vs. Circuit-Switching

- Most important advantage of packet-switching over circuit switching: ability to exploit statistical multiplexing
  - Efficient bandwidth usage
  - Ratio between peak and average rate is 3:1 for audio, and 15:1 for data traffic
However, packet-switching needs to deal with **congestion**
- More complex routers
- Harder to provide good network services (e.g., delay and bandwidth guarantees)

In practice they are **combined**
- IP over SONET, IP over Frame Relay
提纲

◆ 网络基本概念
◆ Internet简介
◆ 协议分层和OSI参考模型
◆ End-to-End Arguments
◆ Internet设计原则
◆ 基本机制回顾
The Internet

- Global scale, general purpose, heterogeneous-technologies, public, computer network
- Internet Protocol
  - Open standard: Internet Engineering Task Force (IETF) as standard body
    http://www.ietf.org
  - Technical basis for other types of networks
    • Intranet: enterprise IP network
- Developed by the research community
Internet History

1961-1972: Early packet-switching

- 1961: Kleinrock - queueing theory shows effectiveness of packet-switching
- 1964: Paul Baran - packet-switching in military nets
- 1967: ARPAnet conceived by Advanced Research Projects Agency – Larry Roberts
- 1969: first ARPAnet node operational
- 1972: ARPAnet demonstrated publicly
  - NCP (Network Control Protocol) first host-host protocol
  - first e-mail program
  - ARPAnet has 15 nodes

1961-1972: Early packet-switching principles
1970: ALOHAnet satellite network in Hawaii
1973: Metcalfe’s PhD thesis proposes Ethernet
1974: Cerf and Kahn – architecture for interconnecting networks
late 70’s: proprietary architectures: DECnet, SNA, XNA
late 70’s: switching fixed length packets (ATM precursor)
1979: ARPAnet has 200 nodes

Cerf and Kahn’s internetworking principles:
– minimalism, autonomy
  • no internal changes required to interconnect networks
– best effort service
– stateless routers
– decentralized control
define today’s Internet architecture

1972-1980: Internetworking, new and proprietary nets

[NOTE: THIS MAP DOES NOT SHOW ARPA’S EXPERIMENTAL SATELLITE CONNECTIONS]
NAMES SHOWN ARE IMP NAMES, NOT (NECESSARILY) HOST NAMES
Internet History

1980-1990: new protocols, a proliferation of networks

- **1982**: SMTP e-mail protocol defined
- **1983**: deployment of TCP/IP
- **1983**: DNS defined for name-to-IP-address translation
- **1985**: FTP protocol defined
- **1988**: TCP congestion control

- New national networks: Csnet, BITnet, NSFnet, Minitel
- 100,000 hosts connected to confederation of networks
Internet History

Early 1990's: ARPAnet decommissioned

1991: NSF lifts restrictions on commercial use of NSFnet (decommissioned, 1995)


1994: Mosaic, later Netscape

late 1990s: commercialization of the Web

Late 1990's – 2000's:

- more killer apps: instant messaging, peer2peer file sharing (e.g., BT, Napster)
- network security to forefront

- est. 50 million host, 100 million+ users
- backbone links running at Gbps

1990, 2000's: commercialization, the Web, new apps

Netscape
- late 1990’s: commercialization of the Web
Time Line of the Internet

- **1968**: ARPANET Demonstrated
- **1973**: TCP/IP Invented
- **1983**: ARPANET Widely Used
- **1983**: MILNET/ARPANET Split
- **1986**: NSI-net Initiated
- **1993**: Internet Society Founded
- **1996**: Many Thousands of Everything
- **1996**: World Wide Web
- **1996**: Multi-Protocol Environment

- **Operational Networks On Internet**
  - 3
  - 20
  - 60
  - 300
  - 500
  - 900
  - 19,000
  - 50,000

*Source: Internet Society*
Growth of the Internet

- **Number of Hosts on the Internet:**
  - Aug. 1981: 213
  - Oct. 1984: 1,024
  - Dec. 1987: 28,174
  - Oct. 1990: 313,000
  - Oct. 1993: 2,056,000
  - Apr. 1995: 5,706,000
  - Jan. 1997: 16,146,000
  - Jan. 1999: 56,218,000
  - Jan. 2001: 109,374,000
  - Jan. 2003: 171,638,297
  - Jan. 2006: 394,991,609
  - July 2007: 489,774,269
  - Jan. 2009: 625,226,456
  - July. 2009: 681,064,561

Data available at: http://www.isc.org/
Growth of the Internet

Internet Usage by World Region

- Asia: 437 millions
- Europe: 322 millions
- North America: 233 millions
- Latin America: 110 millions
- Africa: 34 millions
- Middle East: 20 millions
- Australia/Oceania: 19 millions

Copyright © 2007, www.internetworldstats.com
Who is Who in the Internet?

Internet Society (ISOC)

- ISOC is a professional membership society with more than 100 organizations and over 20,000 individual members in over 180 countries.
- It provides leadership in addressing issues of the Internet, and is the organization home for the groups responsible for Internet infrastructure standards, including IETF and IAB.
Who is Who in the Internet?

◆ Internet Engineering Task Force (IETF)
  – The IETF is the protocol engineering and development arm of the Internet
  – Subdivided into many working groups, which specify Request For Comments or RFCs
Who is Who in the Internet?

◆ IRTF (Internet Research Task Force)
  - The Internet Research Task Force is composed of a number of focused, long-term and small Research Groups

◆ Internet Architecture Board (IAB)
  - The IAB is responsible for defining the overall architecture of the Internet, providing guidance and broad direction to the IETF
Who is Who in the Internet?

◆ The Internet Engineering Steering Group (IESG)
  - The IESG is responsible for technical management of IETF activities and the Internet standards process
  - Composed of the Area Directors of the IETF working groups
Who is Who in the Internet?

- **IETF and IESG Chair**
  - Russ Housley, Vigil Security, LLC

- **Applications Area (app)**
  - Lisa Dusseault, Open Source Applications Foundation
  - Alexey Melnikov, Isode Limited

- **Internet Area (int)**
  - Jari Arkko, Ericsson
  - Ralph Droms, Cisco

- **Operations and Management Area (ops)**
  - Ronald Bonica, Juniper Networks
  - Dan Romascanu, Avaya

- **Real-time Applications and Infrastructure Area (rai)**
  - Cullen Jennings, Cisco Systems
  - Robert Sparks, Tekelec

- **Routing Area (rtg)**
  - Ross Callon, Juniper Networks
  - Adrian Farrel, Huawei

- **Security Area (sec)**
  - Pasi Eronen, Nokia
  - Tim Polk, National Institute of Standards and Technology

- **Transport Area (tsv)**
  - Lars Eggert, Nokia Research Center
  - Magnus Westerlund, Ericsson
Internet Standardization Process

◆ All standards of the Internet are published as RFC *(Request for Comments)*. But not all RFCs are Internet Standards.
  - available: http://www.ietf.org

◆ A typical (but not only) way of standardization is:
  - BOF *(Birds of a feather)*
  - Internet Drafts
  - RFC
  - Proposed Standard
  - Draft Standard *(requires 2 working implementation)*
  - Internet Standard *(declared by IAB)*
Internet Standardization Process

–David Clark, MIT, 1992: We reject: kings, presidents, and voters, and believe in: rough consensus and running code.
Services Provided by the Internet

◆ Shared access to computing resources
  – Telnet (1970’s)

◆ Shared access to data/files
  – FTP, NFS, AFS (1980’s)

◆ Communication medium over which people interact
  – Email (1980’s), on-line chat rooms (1990’s)
  – Instant messaging, IP Telephony (2000’s)
Services Provided by the Internet

◆ A medium for information dissemination
  – USENET (1980’s)
  – WWW (1990’s)
    • Replacing newspaper, magazine?
  – Audio, video (2000’s)
    • Replacing radio, CD, TV...
Internet structure: network of networks

- roughly hierarchical
- at center: “tier-1” ISPs (e.g., UUNet, BBN/Genuity, Sprint, AT&T), national/international coverage
  - treat each other as equals

Tier-1 providers interconnect (peer) privately

Tier-1 providers also interconnect at public network access points (NAPs)
Tier-1 ISP: e.g., Sprint

Sprint US backbone network

POP: point-of-presence

to/from backbone

to/from customers

peering

DS3 (45 Mbps)
OC3 (155 Mbps)
OC12 (622 Mbps)
OC48 (2.5 Gbps)
**Internet structure: network of networks**

- **“Tier-2” ISPs: smaller (often regional) ISPs**
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs

Tier-2 ISPs also peer privately with each other, interconnect at NAP.

Tier-2 ISP pays tier-1 ISP for connectivity to rest of Internet.

Tier-2 ISP is customer of tier-1 provider.
Internet structure: network of networks

◆ “Tier-3” ISPs and local ISPs
  - last hop (“access”) network (closest to end systems)

Local and tier-3 ISPs are customers of higher tier ISPs connecting them to rest of Internet
Internet structure: network of networks

- a packet passes through many networks!

Try a traceroute!
UUNET’s North America Internet network

For more information visit www.uunet.net

NB. UUNET also has infrastructure within individual countries, which is not shown on this map. June 2003

- Single Hub City
- Multiple Hubs City
中国教育和科研计算机网 CERNET 主干网
Access Networks

Q: How to connect end systems to edge router?

◆ residential access nets
◆ institutional access networks (school, company)
◆ mobile access networks

Keep in mind:

◆ bandwidth (bits per second) of access network?
◆ shared or dedicated?
Residential access: point to point access

- **Dialup via modem**
  - up to 56Kbps direct access to router (often less)

- **ISDN**: integrated services digital network
  - 128kbps + regular phone line

- **ADSL**: asymmetric digital subscriber line
  - up to 1 Mbps upstream (today typically < 256 kbps)
  - up to 8 Mbps downstream (today typically < 5 Mbps)
  - FDM: 50 kHz - 1 MHz for downstream
Residential access: point to point access

- **ADSL:** asymmetric digital subscriber line
  - **ADSL**充分利用了双绞铜线的频谱
    - (1) 传统话音业务频段，约为4 kHz带宽
    - (2) **ADSL**上行低速信道，位于话音频谱之上
    - (3) **ADSL**下行高速信道，位于高频部分
  - 频分多路复用(FDM)或回波抵消(Echo Cancellation)技术
  - 在回波抵消**ADSL**系统中，下行信道与上行信道有重叠
  - 与频分复用技术相比，回波抵消技术消除了因频率叠加所带来的干涉(如近端串音)，可使**ADSL**系统在性能指标上有较大的提高，复杂度较高
Residential access: cable modems

- **HFC: hybrid fiber coax**
  - asymmetric: up to 10Mbps upstream, 30 Mbps downstream

- **network of cable and fiber attaches homes to ISP router**
  - shared access to router among homes
  - issues: congestion, dimensioning

- **deployment: available via cable companies, e.g., ComCast**
Residential access: cable modems

Diagram: http://www.cabledatacomnews.com/cmic/diagram.html
Cable Network Architecture: Overview

Typically 500 to 5,000 homes
Cable Network Architecture: Overview

- server(s)
- cable headend
- cable distribution network
- home
Cable Network Architecture: Overview
Cable Network Architecture: Overview

FDM:

cable headend

cable distribution network

home

Channels

1 2 3 4 5 6 7 8 9
Company access: local area networks

- company/univ local area network (LAN) connects end system to edge router
- Ethernet:
  - shared or dedicated link (switched Ethernet) connects end system and router
  - 10 Mbps, 100 Mbps, Gigabit Ethernet
- deployment: institutions, home LANs happening now
Wireless access networks

- shared wireless access network connects end system to router
  - via base station “access point”
- wireless LANs:
  - 802.11b: 11 Mbps
  - 802.11g: 54 Mbps
  - 802.11n: Marvell 提供 450 Mbps
- wider-area wireless access
  - provided by telecom operator
  - 3G ~ 384 kbps
  - WAP/GPRS in Europe
  - GPRS/CDMA in China
Home networks

Typical home network components:
- ADSL or cable modem
- router/firewall/NAT
- Ethernet
- wireless access point

Diagram:
- To/from cable headend
- Cable modem
- Router/firewall
- Ethernet (switched)
- Wireless access point
- Wireless laptops
提纲

- 网络基本概念
- Internet 简介
- 协议分层和 OSI 参考模型
- End-to-End Arguments
- Internet 设计原则
The Big Question

◆ Many different network styles and technologies
  – circuit-switched vs packet-switched, etc.
  – wireless vs wired vs optical, etc.
◆ Many different applications
  – ftp, email, web, P2P, etc.
◆ How do we organize this mess?
Why Layering?

Do we re-implement every application for every technology?
Obviously not, but how does the Internet architecture avoid this?
Why Layering?

Solution: introduce an intermediate layer that provides a unique abstraction for various network technologies.

![Diagram showing the layers and media types](attachment:diagram.png)
Architecture

◆ Architecture is **not** the implementation itself
◆ Architecture is how to “organize” implementations
  – what interfaces are supported
  – where functionality is implemented
◆ Architecture is the **modular design** of the network
Software Modularity

Break system into modules:
- **Well-defined interfaces give flexibility**
  - can change implementation of modules
  - can extend functionality of system by adding new modules
- **Interfaces hide information**
  - allows for flexibility
  - but can hurt performance
Network Modularity

Like software modularity, but with a twist:

- Implementation distributed across routers and hosts
- Must decide both:
  - how to break system into modules
  - where modules are implemented
Layering

- Layering is a particular form of modularization

- The system is broken into a vertical hierarchy of logically distinct entities (layers)

- The service provided by one layer is based solely on the service provided by layer below
Layering

◆ Advantages
  - **Modularity** – protocols easier to manage and maintain
  - **Abstract functionality** – lower layer can be changed *without* affecting the upper layer
  - **Reuse** – upper layer can reuse the functionality provided by lower layer

◆ Disadvantages
  - **Information hiding** – inefficient implementations
Layering

- Layer N software on the destination computer must receive exactly the message sent by layer N software on the sending computer.
- Mathematically, if the sender applies a transformation $T$, the receiver must apply the inverse $T^{-1}$. 
ISO OSI Reference Model

- ISO – International Standard Organization
- OSI – Open System Interconnection
- Started in 1978; first standard 1979
  - ARPANET started in 1969; TCP/IP protocols ready by 1974
- Goal: a general open standard
  - Allow vendors to enter the market by using their own implementation.
ISO OSI Reference Model

- Seven layers
  - Lower three layers are **node-to-node**
  - Next four layers are **end-to-end**

```
Application
Presentation
Session
Transport
Network
Datalink
Physical
```

```
Application
Presentation
Session
Transport
Network
Datalink
Physical
```

**Physical medium**
Data Transmission

- A layer can use **only** the service provided by the layer immediate below it
- Each layer may change and add a header to data packet
**OSI Model Concepts**

- **Service** – says *what* a layer does
- **Interface** – says *how* to access the service
- **Protocol** – says *how* is the service implemented
  - A set of rules and formats that govern the communication between two peers
  - protocol does not govern the implementation on a single machine, but how the layer is implemented between machines
The Origins of OSI

- Much of the work on the design of OSI was actually done by a group at Honeywell Information Systems
  - Charlie Bachman as the principal technical member
- The group studied some of the existing solutions, including IBM’s system network architecture (SNA), the work on protocols being done for ARPANET, the result of this effort was the development by 1977 of a seven-layer architecture known internally as the distributed systems architecture (DSA)
- In 1977, ISO formed a subcommittee on Open Systems Interconnection (Technical Committee 97, Subcommittee 16)
- This model was chosen as the only proposal to be submitted to the ISO subcommittee
- A consensus was reached at that meeting that this layered architecture would satisfy most requirements of Open Systems Interconnection, and had the capacity of being expanded later to meet new requirements. A provisional version of the model was published in March of 1978
- The next version, with some minor refinements, was published in June of 1979 and eventually standardized
- The resulting OSI model is essentially the same as the DSA model developed in 1977
Physical Media

- **Bit**: propagates between transmitter/rcvr pairs
- **physical link**: what lies between transmitter & receiver
- **guided media**:
  - signals propagate in solid media: copper, fiber, coax
- **unguided media**:
  - signals propagate freely, e.g., radio

**Twisted Pair (TP)**
- two insulated copper wires
  - Category 3: traditional phone wires, 10 Mbps Ethernet
  - Category 5 TP: 100Mbps Ethernet
Physical Media: coax, fiber

**Coaxial cable:**
- two concentric copper conductors
- bidirectional
- baseband:
  - single channel on cable
  - legacy Ethernet
- broadband:
  - multiple channel on cable

**Fiber optic cable:**
- glass fiber carrying light pulses, each pulse a bit
- high-speed operation:
  - high-speed point-to-point transmission (e.g., 10 Gbps-40Tbps)
- low error rate: repeaters spaced far apart; immune to electromagnetic noise
Physical media: radio

- signal carried in electromagnetic spectrum
- no physical “wire”
- bidirectional
- propagation environment effects:
  - reflection
  - obstruction by objects
  - interference

Radio link types:
- terrestrial microwave
  - e.g. up to 45 Mbps channels
- LAN (e.g., 802.11)
  - 2Mbps, 11Mbps, 55Mbps
- wide-area (e.g., cellular)
  - e.g. 3G: hundreds of kbps
- satellite
  - up to 50Mbps channel (or multiple smaller channels)
  - 270 msec end-end delay
Physical Layer

◆ **Service**: move the information between two systems connected by a physical link

◆ **Interface**: specifies how to send a bit

◆ **Protocol**: coding scheme used to represent a bit, voltage levels, duration of a bit

◆ **Examples**: coaxial cable, optical fiber links; transmitters, receivers
物理层的四个重要特性

◆ 机械特性 *(mechanical characteristics)*
  - 主要定义物理连接的边界点，即接插装置
  - 规定物理连接所采用的规格、引脚的数量和排列

◆ 电气特性 *(electrical characteristics)*
  - 规定传输二进制位时，线路上信号的电压高低、阻抗匹配、传输速率和距离限制

◆ 功能特性 *(functional characteristics)*
  - 主要定义各条物理线路的功能

◆ 规程特性 *(procedural characteristics)*
  - 主要定义各条物理线路的工作规程和时序关系
Hubs

- Physical Layer devices: essentially repeaters operating at bit levels: repeat received bits on one interface to all other interfaces
- Hubs can be arranged in a hierarchy (or multi-tier design), with backbone hub at its top
Hubs (more)

- Each connected LAN referred to as LAN segment
- Hubs do not isolate collision domains: segments form a large collision domain
  - if a node CS and a node EE transmit at same time: collision
Hubs (more)

Hub Advantages:
- simple, inexpensive device
- Multi-tier provides graceful degradation: portions of the LAN continue to operate if one hub malfunctions
- extends maximum distance between node pairs (100m per Hub)
Datalink Layer (2)

- **Service:**
  - Framing, i.e., attach frames separator
  - Send data frames between peers attached to the same physical media
  - Others (optional):
    - Arbitrate the access to common physical media
    - Ensure reliable transmission
    - Provide flow control

- **Interface:** send a data unit (packet) to a machine connected to the *same* physical media

- **Protocol:** layer addresses, implement Medium Access Control (MAC) (e.g., CSMA/CD)...
Random Access Protocols

◆ When node has packet to send
  - transmit at full channel data rate R
  - no \textit{a priori} coordination among nodes

◆ two or more transmitting nodes \(\rightarrow\) \textit{“collision”}

◆ \textit{random access MAC protocol} specifies:
  - how to detect collisions
  - how to recover from collisions (e.g., via delayed retransmissions)

◆ Examples of random access MAC protocol
  - ALOHA
  - slotted ALOHA
  - CSMA, CSMA/CD, CSMA/CA
局域网链路层协议

\( A: \) sense channel, if idle

then {

transmit and monitor the channel;

\textbf{If} detect another transmission

then {

abort and send \textit{jam} signal;

update \# collisions;

delay as required by exponential backoff algorithm;

goto A

}

else \{done with the frame; set collisions to zero\}

}

else \{wait until ongoing transmission is over and goto A\}
**Jam Signal:** make sure all other transmitters are aware of collision; 48 bits;

**Exponential Backoff:**
- first collision for given packet: choose \( K \) randomly from \{0,1\}; delay is \( K \times 512 \) bit transmission times
- after second collision: choose \( K \) randomly from \{0,1,2,3\}...
- After next collision double \( K \) (and keep doubling on collisions until...)
- after ten or more collisions, choose \( K \) randomly from \{0,1,2,3,4,...,1023\}
IEEE 802.3

- IEEE定义了采用1-坚持型CSMA/CD技术的802.3局域网标准
- 802.3基带系统都使用曼彻斯特编码
- 二进制指数后退算法（binary exponential backoff）
Ethernet Bus implementations

◆ **Early days** - the vampire tap

◆ **Later:** “thinnet” (coax with T connectors)
Star topology

- Twisted pair is better
- 10baseT introduced in early 90s, never went back
- Connection choices: hub or switch
Bridges

- **Link layer device**
  - stores and forwards Ethernet frames
  - examines frame header and **selectively** forwards frame based on MAC dest address
  - when frame is to be forwarded on segment, uses CSMA/CD to access segment
  - can connect **different** type Ethernet

- **transparent**
  - hosts are unaware of presence of bridges

- **plug-and-play, self-learning**
  - bridges do not need to be configured
Bridges: traffic isolation

- Bridge installation breaks LAN into LAN segments
- Bridges **filter** packets:
  - segments become separate **collision domains**

![Diagram showing LAN segments and bridges]

- collision domain
- bridge
- collision domain

- LAN segment
- LAN (IP network)
- LAN segment

= hub
= host
Forwarding

How do determine to which LAN segment to forward frame?
• Looks like a routing problem...
Self learning

- bridge has a **bridge table**
- entry in bridge table:
  - (Node LAN Address, Bridge Interface, Time Stamp)
  - stale entries in table dropped (TTL can be 60 min)
- bridges **learn** which hosts can be reached through which interfaces
  - when frame received, bridge “learns” location of sender: incoming LAN segment
  - records sender/location pair in bridge table
Filtering/Forwarding

When bridge receives a frame:

index bridge table using MAC dest address
if entry found for destination
  then {
    if dest on segment from which frame arrived
      then drop the frame
    else forward the frame on interface indicated
  }
else flood
  forward on all but the interface on which the frame arrived
Scenario:
- C sends frame to D
- D replies back with frame to C

Bridge Table

<table>
<thead>
<tr>
<th>address</th>
<th>port</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>H</td>
<td>2</td>
</tr>
<tr>
<td>I</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>3</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
</tr>
</tbody>
</table>
Interconnection without backbone

Not recommended for two reasons
- single point of failure at Computer Science hub
- all traffic between EE and SE must pass over CS segment
Backbone configuration

Recommended!
Bridges Spanning Tree

- for increased reliability, desirable to have redundant, alternative paths from source to dest
- with multiple paths, cycles result - bridges may multiply and forward frame forever
- **solution:** organize bridges in a **spanning tree** by disabling subset of interfaces
LAN Addresses and ARP

32-bit IP address:
- *network-layer* address
- used to get datagram to destination IP network

LAN (or MAC or physical or Ethernet) address:
- used to get datagram from one interface to another physically-connected interface (same network)
- 48 bit MAC address (for most LANs) burned in the adapter ROM
LAN Addresses and ARP

Each adapter on LAN has unique LAN address
LAN Address (more)

- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- MAC flat address => portability
  - can move LAN card from one LAN to another
- IP hierarchical address NOT portable
  - depends on IP network to which node is attached
ARP: Address Resolution Protocol

Question: how to determine MAC address of B knowing B’s IP address?

- Each IP node (Host, Router) on LAN has ARP table.
- ARP Table: IP/MAC address mappings for some LAN nodes < IP address; MAC address; TTL >
  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)
ARP protocol

- A wants to send datagram to B, and A knows B's IP address
- Suppose B's MAC address is not in A's ARP table
- A broadcasts ARP query packet, containing B's IP address
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - frame sent to A's MAC address (unicast)
- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - **soft state:** information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - nodes create their ARP tables without intervention from net administrator
**Service:**
- Deliver a packet to specified destination
- Perform segmentation/reassemble (fragmentation/defragmentation)
- Others:
  - Packet scheduling
  - Buffer management

**Interface:** send a packet to a specified destination

**Protocol:** define global unique addresses; construct routing tables
Data and Control Planes

◆ **Data plane**: concerned with
  – Packet forwarding
  – Buffer management
  – Packet scheduling

◆ **Control Plane**: concerned with installing and maintaining state for data plane
Example: Routing

- **Data plane**: use **Forwarding Table** to forward packets
- **Control plane**: construct and maintain Forwarding Tables (e.g., Distance Vector, Link State protocols)
Router Architecture Overview

Two key router functions:
- run routing algorithms/protocol (RIP, OSPF, BGP)
- switching datagrams from incoming to outgoing link
Bridges vs. Routers

- **both** *store-and-forward* devices
  - routers: *network layer* devices (examine network layer headers)
  - bridges are *link layer* devices
- **routers** maintain routing tables, implement routing algorithms
- **bridges** maintain bridge tables, implement filtering, learning and spanning tree algorithms
Ethernet Switches

- Essentially a **multi-interface bridge**
- layer 2 (frame) forwarding, filtering using LAN addresses
- **Switching**: A-to-A’ and B-to-B’ simultaneously, no collisions
- large number of interfaces
- **often**: individual hosts, star-connected into switch
  - Ethernet, but no collisions!
## Summary comparison

<table>
<thead>
<tr>
<th></th>
<th>hubs</th>
<th>bridges</th>
<th>routers</th>
<th>switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>traffic isolation</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>plug &amp; play</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>optimal routing</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>cut through</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>
Transport Layer (4)

◆ **Service:**
  - Provide an error-free and flow-controlled end-to-end connection
  - Multiplex multiple transport connections to one network connection
  - Split one transport connection in multiple network connections

◆ **Interface:** send a packet to specify destination

◆ **Protocol:** implement reliability and flow control

◆ **Examples:** TCP and UDP
Session Layer (5)

◆ **Service:**
  - Dialog control
  - Access management, e.g., token control
  - Synchronization, e.g., provide check points for long transfers

◆ **Interface:** depends on service

◆ **Protocols:** token management; insert checkpoints, implement roll-back functions
the presentation layer is concerned with the syntax and semantics of the information transmitted

**Service:** convert data between various representations

**Interface:** depends on service

**Protocol:** define data formats, and rules to convert from one format to another
Application Layer (7)

- **Service**: any service provided to the end user
- **Interface**: depends on the application
- **Protocol**: depends on the application
- **Examples**: FTP, Telnet, WWW browser
**OSI vs. TCP/IP**

- **OSI:** conceptually define: service, interface, protocol
- **Internet:** provide a successful implementation

<table>
<thead>
<tr>
<th>OSI Layers</th>
<th>TCP/IP Layers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Telnet, FTP, DNS</td>
</tr>
<tr>
<td>Presentation</td>
<td>Transport</td>
</tr>
<tr>
<td>Session</td>
<td>Internet</td>
</tr>
<tr>
<td>Transport</td>
<td>Data Link</td>
</tr>
<tr>
<td>Network</td>
<td>Physical</td>
</tr>
<tr>
<td>Datalink</td>
<td>Radios</td>
</tr>
<tr>
<td>Physical</td>
<td>LAN</td>
</tr>
</tbody>
</table>

- Telnet, FTP, DNS | TCP, UDP |
- IP | LAN |
- Packet, radio | Network |
Andrew S. Tanenbaum 在 “Computer Networks” 第三版中评价 OSI:
- Bad timing (too late)
- Bad technology (both the model and the protocol are flawed)
- Bad politics (government and organizations bureaucrats)

Internet 标准化名言 (David Clark of MIT):
- “We reject kings, presidents, and voting; we believe in rough consensus and running code”
提纲

◆ 网络基本概念
◆ Internet简介
◆ 协议分层和OSI参考模型
◆ End-to-End Arguments
◆ Internet设计原则
◆ 基本机制回顾
Two Aspects to Architecture

- **Layering**
  - how to break network functionality into modules

- **The End-to-End Argument**
  - where to implement functionality
Internet & End-to-End Argument

- network layer provides one simple service:
  - best effort datagram (packet) delivery
- transport layer at network edge (TCP) provides end-to-end error control
- all other functionality ...
  - all application layer functionality
  - network services: DNS
  implemented at application level
Discussion: congestion control, flow control: why at transport, rather than link or application layers?

- Congestion control needed for many applications
- Many applications “don’t care” about congestion control - it’s the network’s concern
- Consistency across applications - you *have* to use it if you use TCP (social contract - everybody does)
- Why do it at the application level
  - Flow control - application knows how/when it wants to consume data
  - Congestion control - application can do tcp-friendly
Internet & End-to-End Argument

Why not at the link layer

1: not every application needs/want it
2: lots of state at each router (each connection needs to buffer, need back pressure) - it's hard
E2E Argument: Interpretations

◆ One interpretation:
  - A function can only be completely and correctly implemented with the knowledge and help of the applications standing at the communication endpoints

◆ Another: (more precise …)
  - A system (or subsystem level) should consider only functions that can be completely and correctly implemented within it
Alternative interpretation: (also correct ...)

- Think **twice** before implementing a functionality that you believe that is useful to an application at a lower layer

- If the application can implement a functionality correctly, implement it at a lower layer **only** as a performance enhancement
End-to-End Argument: Critical Issues

Philosophy: if application can do it, don’t do it at a lower layer -- application best knows what it needs
- add functionality in lower layers iff
- (1) used by and improves performances of many applications
- (2) does not hurt other applications

allows cost-performance tradeoff
提纲

◆ 网络基本概念
◆ Internet简介
◆ 协议分层和OSI参考模型
◆ End-to-End Arguments
◆ Internet设计原则
Internet Design Philosophy (Clark’88)

In order of importance:

0. Connect existing networks
   - initially ARPANET, ARPA packet radio, packet satellite network

1. Survivability
   - ensure communication service even with network and router failures

2. Support multiple types of services

3. Must accommodate a variety of networks

4. Allow distributed management

5. Allow host attachment with a low level of effort

6. Be cost effective

7. Allow resource accountability
What About the Future

- Datagram not the best abstraction for:
  - resource management, accountability, QoS
- new abstraction: flow (see IPv6)
  - but no one knows what a flow is
- routers require to maintain per-flow state
- state management: recovering lost state is hard
- here (1988) we see the first proposal of “soft state”!
  - soft-state: end-hosts responsible to maintain the state
Summary: Internet Architecture

- packet-switched datagram network
- IP is the glue (network layer overlay)
- IP hourglass architecture
  - all hosts and routers run IP
- stateless architecture
  - no per flow state inside network
**Summary: Minimalist Approach**

- **Dumb network**
  - IP provide minimal functionalities to support connectivity
  - addressing, forwarding, routing

- **Smart end system**
  - transport layer or application performs more sophisticated functionalities
  - flow control, error control, congestion control

- **Advantages**
  - accommodate heterogeneous technologies (Ethernet, modem, satellite, wireless)
  - support diverse applications (telnet, ftp, Web, X windows)
  - decentralized network administration