

# 计算机网络

---

## 1. Introduction

- Networks: exchange something from one site to the others
- Why computer networks? 【s1 p20】
  - Communication, resource sharing, collaboration.

## Connectivity

- What do we need to connect two computers? 【s1 p27】
  - computers
  - links
- Building blocks 【s1 p28】
  - Nodes
    - Hosts: work for the users
    - Switches: work for the network
  - Direct links
    - point to point
    - **multiple access** (常用)
- Device at the center
  - Hub ( 集线器 ), a box
  - Switch ( 交换机 ), a computer
  - Router ( 路由器 )
- Switched networks 【s1 p30】
  - suitable for scaling up
  - circuit switched (tel.), packet switched (computer)
  - can be defined recursively
- Network scales 【s1 p32】
  - SAN, LAN, WAN, MAN, Internet

## Networking resource sharing

- Fundamental resource sharing concept: **multiplexing** ( 复用 )
- Solutions 【s1 p35】
  - Synchronous Time-Division Multiplexing (STDM)
    - 分时间段使用 · 这段时间归谁 · 那段时间归谁 · 不能乱用。
  - Frequency-Division Multiplexing (FDM)

- 分出多个lane，每个lane分配一台电脑，同样是专属。这样效率还是不高，可能一条道没用，另一条道却堵。
- Packet switching (Statistical Multiplexing, On-demand Time-division)
  - 将各个电脑的数据包分成小块，小块混杂着传输。
  - 各个小块过程中可能线路不同，但最终到达的目的地相同。
- Circuit vs Packet switching 【s1 p37】

## Architecture

- Support for common services 【s1 p42】
    - request/reply channel
    - message stream channel: videos, sending message continuously
  - Layering 【s1 p43】
    - Application programs, Process-to-process channels, Host-to-host connectivity, Hardware
  - **Layers of network applications** 【s1 p44】
    - Users
    - Applications
    - Network services
    - Network hardware
  - Simplified file transfer architecture
- 

## 2. Network architecture

### Layering and protocols

- **Protocols** 【s2 p7】
  - Standards at a layer
  - a library function, or a running process
  - protocol stack: set of consecutive layers
- **Interfaces** 【s2 p8】
  - service interface
  - peer-to-peer interface
- Protocol graph
  - most peer-to-peer communication is indirect, direct only at hardware level.
  - RRP: request/reply protocols
  - MSP: message stream
  - HHP: host-to-host
- Encapsulation (封装)

- Data  $\rightarrow$  (RRP, Data)  $\rightarrow$  (HHP, RRP, Data)  $\rightarrow$  (RRP, Data)  $\rightarrow$  Data

## Architecture

### ISO/OSI Model

- 7 layers
  - 7: Application
  - Presentation
  - Session
  - Transport
  - Network
  - Data link
  - 1: Physical
- 太麻烦了，仅供参考，实际并未使用。
- Nodes 至少要有3个layer: physical, data link, network
- Brief introduction of the 7 layers 【s2 p14~p16】

### TCP/IP architecture

- 漏斗形 hourglass shape 【s2 p17】
- Does not imply strict layering
- 4 layers 【s2 p19】
  - 4: Application
  - Transport
  - Network
  - 1: Physical

### Application programming interface (API)

- How can we build network applications? 【s2 p24】
  - applications: inter-process communication
  - common services
  - A **network architecture** is a layered arrangement of common services, which support inter-process communication.
- The most widely used network API: **socket interface**
  - A socket is defined as an endpoint for communication
  - Host num: Port num, 161.25.19.8: 1625
- The client/server paradigm 【s2 p28】

- steps
  - Further notes about socket API 【s2 p37】
- 

### 3. Direct connection

#### Network performance

- Performance metrics 【s3 p4】
  - Bandwidth
    - data transmitted per time unit
    - Mbps =  $10^6$  bits per second
  - Latency
    - time to send message from point A to point B
    - RTT: round-trip time.  $RTT = \text{Propagation delay} \times 2$
    - **Latency = Propagation + Transmit + Queue**
  - Throughput =  $\frac{\text{TransferSize}}{\text{TransferTime}}$
- Latency 和 Bandwidth 可以看作 fix cost 和 unit cost。
  - For high speed networks, latency dominates.
- Delay  $\times$  Bandwidth product
  - amount of data "in the pipe"
  - **the amount of data the sender can send before the receiver receives.**
- 当算大小时，用2的次方换算，如  $1\text{KB} = 1024\text{ Bytes}$ ；当算速度时，用10的次方换算，如  $1\text{Kbps} = 1000\text{ bits/s}$

#### Direct link

- One can send message to the other without the third computer.
- Direct link networks 【s3 p14】
  - connecting
  - encoding
  - framing
  - error detection
  - reliable delivery
  - media access control

#### Connecting hardware

- Access technologies 【s3 p17】
  - Fiber, cable, Wi-Fi, cellular
- Node architecture, network adapter, links 【s3 p18~p20】

- Typical cables 【s3 p22】
  - twisted pairs
  - coaxial cables
  - fiber optic cables
- Wireless links 【s3 p26】
  - Mobile phone system
  - Wireless LAN/WAN
  - Short range device connection (infrared, bluetooth)

## Encoding, framing, and error detection (NOT required)

### Reliable transmission

- Transmitted frames may be delayed or lost.
- Solution: Automatic Repeat reQuest (ARQ)
  - Correctness: each packet is released to the network layer once and only once, without error
  - Efficiency
- Stop and wait 【s3 p51】
  - Acknowledgement and timeouts
  - It does not keep the pipe full
- Improvements 【s3 p54】
  - Concurrent logical channels
- **Sliding window** 【s3 p55】
  - allow multiple outstanding un-ACKed frames
  - upper bound: window. 【s3 p57】
    - Sender window size (SWS): Delay  $\times$  Bandwidth product.
    - Receiver window size (RWS): = 1 or = SWS

---

## 4. Direct link networks

- Ethernet, Token rings, Wireless 【s4 p4】
- IEEE 802.x specifications 【s4 p6】
- Media Access Control (MAC)
  - Ethernet: CSMA/CD
  - Token ring
  - Wireless: CSMA/CA

### Ethernet

- Physical connection 【s4 p9】
- Physical topologies 【s4 p10】
  - Linear bus topology
    - not so reliable, many single failure points
    - difficult to identify the problem if the entire network shuts down
  - Star topology
    - require more cable length, but more reliable
- Ethernet transmission 【s4 p14】
  - Broadcast
    - all stations receive all transmissions
  - CSMA/CD medium access control (MAC) scheme 【s4 p15】
    - all hosts compete for the same link. possibility of collisions
  - Transmission algorithms
    - 1-persistent protocol
    - p-persistent protocol
- Ethernet addresses 【s4 p18】
  - addresses are assigned to network adapters
  - 48-bits
  - address recognition 【s4 p19】
  - Possible destinations 【s4 p20】
    - unicast, broadcast, multicast
  - MAC address is used to distinguish between the destinations
  - Promiscuous mode

## Token rings

- Ring topology 【s4 p26】
  - prevent collisions before occurring
- Token ring transmission 【s4 p27】
- Token passing paradigm
  - travel in a unidirectional fashion
- When the node fails, the relay closes bypassing the node.
  - 在物理角度与 Ethernet 并没有本质区别，同样可以用 Hub
- Token ring maintenance 【s4 p30】
  - Ring can elect a new monitor. Highest MAC address wins.
- FDDI (Fiber Distributed Data Interface) dual ring operation
  - no single failure point

- 但 token ring 太复杂了，现在更多用 Ethernet 的结构。

## Wireless

- Wireless links 【s4 p33】
    - similar to Ethernet structure, sometimes called "wireless ethernet".
  - Access point (AP)
  - Ad hoc (mesh structure) 【s4 p35】
  - Most radios in wireless networking can't transmit and receive at the same time, so we can't detect collisions. Instead, we'll do CSMA/CA (collision avoidance).
  - Problems
    - The hidden node problem 【s4 p37】
    - The exposed node problem
    - Solution: Multiple Access with Collision Avoidance (MACA) 【s4 p39】
- 

## 5. Packet switching

- Problems of direct link networks 【s5 p5】
  - limitation: scalability
  - key problem: inter-connecting networks
  - solution: packet switching
- Packet switching
  - connection-oriented (switched WAN, Frame-Relay, ATM)
  - connectionless = datagram switching

## Inter-networking

- General working mechanism: switching and forwarding
- Inter-connecting links of the same type: Bridges/LAN switches
- Inter-connecting links of different types: Gateways/routers, the Internet Protocol

## Switching and forwarding

- Ethernet 【s5 p9】
  - shared medium: Bus
  - Hub
- Switched LAN
  - 用的是 switch 而不是 Hub

- NO shared medium
- no direct link between hosts. 因为switch算是一台电脑
- 现在实际上只用 switch 而不用 hub 了，因为成本也不高。
- Switch
  - Each one runs a **data-link protocol**.
  - Each input or output is a port (interface). 这里的port指的是物理意义上的接口，而不是数字。
  - Layer 3 switch (**router**): links may run different data link protocols
- Hub, Switch/Bridge, and Router/Gateway 【s5 p13~14】

### How does a switch decide what output port to use?

- Datagrams
  - connectionless
  - forwarding table
  - For connectionless networks 【s5 p17】
    - routing is important
    - it is not known whether the network can deliver the packet
- Virtual circuit 【s5 p18】
  - connection-oriented
  - Two part process:
    - connection setup
    - data transfer
  - virtual circuit table
  - connection establishment 【s5 p20】
    - Permanent Virtual Circuit (PVC)
    - Switched Virtual Circuit (SVC)
  - Typical virtual circuit networks
    - Frame relay (support VPN)
    - Asynchronous transfer mode (ATM)
- Source routing 【s5 p22】

### Bridges/LAN Switches 【s5 p24】

- LAN Bridge operation
- Learn table entries based on source address.
  - 一开始是空表，例如A从p1来，就知道它在p1，此时broadcast，因为不知道目标在哪，如此反复直到填出全表。
- Extended LANs can have loops
  - solution: minimum spanning tree



- Limitations 【s5 p30】
    - do not scale
    - do not accommodate heterogeneity
- 

## 6. Basic internetworking

### The network layer

- Connecting heterogeneous networks 【s6 p8】
  - requirements 【s6 p6】
  - Router/Gateway
- Key functions of the network layer 【s6 p12】
  - global addressing
  - data conversion
  - fragmentation
  - routing

### The Internet Protocol (IP) 【s6 p14】

- IP datagram delivery 【s6 p18】
  - connectionless
  - Best-effort ( 尽力而为 )
    - unreliable service
- Packet format
- Datagram size
  - Maximum Transmission Unit (MTU)
  - How to choose the max datagram size for IP?
    - fragment IP datagram into multiple MTUs
- Fragmentation and assembly 【s6 p21】

### Address and datagram forwarding

- IP addressing
  - independent of MAC addressing
  - used by: higher layer protocols, applications
  - virtual
  - 32-bit (IPv4)
  - unique value for each host/interface
    - prefix identifies the network (assigned by global authority), suffix identifies the host/interface (assigned by local administrator). 【s6 p28】

- Dotted decimal notation
    - 4 decimal values per 32-bit address
    - each number represents 8 bits, the value is between 0 and 255.
  - Classes of IP addresses 【s6 p30】
    - class A: large networks, host 24
    - class B: medium, host 16
    - class C: small, host 8
  - Special address
    - 0, booting
    - 255, broadcast
  - Router addresses
    - need one address per router connection
  - Datagram forwarding/routing
    - next hop
  - Subnetting 【s6 p38】
    - subnet masks 分割 · 后面为0的部分为host可以使用的位置。
- 

## 7. IP related protocols

- Address translation (ARP)
- Host configuration (DHCP)
- Error reporting (ICMP)
- Virtual networks and tunnels

### ARP

- Address translation 【s7 p7】
  - map IP address into physical (MAC) addresses
  - ARP (Address Resolution Protocol)
- ARP 【s7 p8】
  - ARP Request: who has this IP address?
  - ARP Reply: I have xxx. My MAC address is xxx.
  - Reverse ARP Request (RARP): who has this MAC address?
  - RARP Reply: I have the MAC. My IP address is xxx.
- ARP details 【s7 p11】

- ARP spoofing attacks 【s7 p16】
  - Denial of Service (DoS)
  - Man in the middle
  - MAC flooding
  - mitigating factor
    - only local attackers can exploit ARP's insecurities
  - How to prevent? 【s7 p20】
    - cannot be completely fixed

## Host configuration (DHCP)

- Dynamic Host Configuration Protocol (DHCP) 【s7 p22】
  - IP addresses bound to workstations dynamically
  - Benefits and limitations 【s7 p28】

## Error reporting (ICMP)

- Internet Control Message Protocol (ICMP) 【s7 p31】
  - Route test
    - Echo (ping), trace route
  - Error reporting
  - Redirect

## Virtual networks and channels

- Private networks 【s7 p35】
- Virtual Private Networks (VPN)
  - sharing some of the same physical links, but make it private. 因为单独买physical的线太贵了.....
  - 在R1信息加密 ( 包括地址也加密, 别人拿到也看不懂 ) , 传到R2解密, 然后再按地址传。( 于是就可以躲过某些地址筛查 )

---

# 8. Routing and the global Internet

## Distance vector routing

- Routing table
- Network as a graph
  - all-pairs shortest path problem
- Static solution 【s8 p9】
  - problem
- Distance vector routing 【s8 p11】
  - algorithm 【s8 p13】

- deal with link failures 【s8 p15】
  - Periodic updates
  - Triggered updates
- count-to-infinity problem (2-node loops)
  - solution 【s8 p18】
- Routing Information Protocol (RIP) 【s8 p19】

## Link state routing 【s8 p21】

- spread the local knowledge of each node to all other nodes in the network, so that every node can construct a network weighted graph
- Strategy 【s8 p22】
  - send to all nodes (not just neighbors) information about directly connected links (not entire routing table)
  - Link State Packet (LSP)
- Link state routing mechanism
  - reliable flooding 【s8 p24】
  - Dijkstra
- Properties 【s8 p26】
- Open Shortest Path First Protocol (OSPF)

## Global Internet

- Multi-backbone Internet 【s8 p30】
  - challenges 【s8 p32】
- Related topics 【s8 p33】
  - Subnetting
  - Network address translation (NAT) 【s8 p35】
  - IPv6 【s8 p37】
    - 128-bit addresses
    - notation 【s8 p38】
    - address space allocation
    - packet format

---

## 9. End to End Protocols

### Transport protocols 【s9 p6】

- Provide application-to-application (process-to-process) communication. Called "end-to-end"
- Optionally provide:

- Reliability, Flow control, Congestion control
- End to end protocols
  - Simple demultiplexer (UDP)
  - Reliable byte stream (TCP)

## UDP

- User Datagram Protocol 【s9 p9】
  - **Unreliable** message delivery
  - Connectionless
  - No flow control
  - No error recovery (**no ACKs**)
  - Application multiplexing
- Addresses for applications: protocol ports 【s9 p11】
  - server: lower port numbers
  - client: higher port numbers
- Simple asynchronous demultiplexing 【s9 p13】
- Basic firewalls 【s9 p16】

## TCP

- Transmission Control Protocol 【s9 p18】
  - Most popular layer 4 protocol
  - Connection-oriented protocol
- TCP feature summary
  - provides a stream transport service
  - allows two application programs to...
- Reliable byte stream 【s9 p20】
  - reliable, in-order delivery
- Segment format 【s9 p21】
- An apprent contradiction? 【s9 p22】
  - IP and TCP
  - How is this possible? ACK mechanism.
- Achieving reliability
  - reliable connection setup
    - must be reliable

- reliable data transmission
- reliable connection shutdown
  - must be graceful
- Why startup/shutdown is difficult? 【s9 p25】
- TCP Startup solution: **3-way handshake** 【s9 p26】
  - use three-message exchange
  - SYN, SYN+ACK, ACK
  - Segments are labeled with a **sequence number**. 【s9 p29】
    - Protect from out-of-order delivery.
    - Initial Sequence Numbers (ISN)
- TCP Shutdown: 四次挥手 【s9 p30】
- Reliable data transmission 【s9 p31】
  - Guarantees a reliable delivery of data: **ARQ**
  - Ensures that data is delivered in order: **SeqNum**
  - Enforces flow-control between sender and receiver: **Sliding window**
- Reliable delivery of data
  - Positive Acknowledgement
  - **Retransmission** 【s9 p33】
- Retransmission
  - Problem: how long should wait?
    - distance to destination, current traffic condition. 在TCP层面上，不知道中间的路是怎样的（不像data link layer）
  - Solution: **adaptive retransmission**
    - User current estimate to set retransmission timer
    - $\text{EstimatedRTT} = \alpha \text{EstimatedRTT} + (1-\alpha) \text{SampleRTT}$
    - $\text{TimeOut} = 2 \times \text{EstimatedRTT}$
    - 这里的SampleRTT指的是最近一次检测到的数据
- Pipeline operation
  - 由 **receiver** 决定 window size
- TCP flow control 【s9 p38】
  - sliding window protocol
  - receiver: **advertise available buffer space**
  - sender: can send up to entire window before ACK arrives
- Window advertisement 【s9 p40】
  - Each acknowledgement carries new window information

- Sliding window 【s9 p41】
    - Sending side: buffer bytes between LastByteAcked and LastByteWritten
    - Receiving side: buffer bytes between LastByteRead and LastByteRcvd
  - Flow control 【s9 p42】
    - Receiving side
    - Sending side
- 

## 10. End-to-End Data

- End-to-End data
  - Presentation formatting
  - Markup language (XML)
  - Multimedia data

### Presentation formatting

- Problem 【s10 p7】
  - at the transport layer (TCP/UDP), a message is sent as an uninterpreted string of bytes.
  - How to encode the different kinds of data into byte strings?
- Encoding/Decoding
- Byte order: big-/little-endian
- Data types
- Formatting approaches 【s10 p11】
  - Argument marshalling ( 参数编组 )
    - Remote Procedure Call (**RPC**) 【s10 p12】
  - Markup documents

### Markup languages

- XML 【s10 p19】
- Difference between XML and HTML 【s10 p22】
  - XML is designed to carry data, not to display data
  - HTML is designed to display data
- Flexibility of XML 【s10 p23】
- XML element, prolog, syntax, tree
- Displaying XML 【s10 p28】
  - CSS

- **XSL** (eXtensible Stylesheet Language) preferred!
- XML validation 【s10 p32】
  - DTD (Document Type Definition)
- JSON
  - a lightweight data-interchange format
- YAML (Yet Another Markup Language)
- Application of XML
  - From HTML to XHTML
  - General document formats
    - docx
  - MathML
  - SVG
  - Synchronized Multimedia Integration Language (SMIL)
  - RSS (Really Simple Syndication)

## Multimedia data

- The amount of data is very large
- Compression methods 【s10 p53】
  - Image compression: JPEG
    - 记一次颜色 · 然后标出哪些坐标对应哪些颜色
  - Video compression: MPEG
    - I frames (intrapicture): self-contained, 保存完整的图片
    - P frames (predicted picture): depending on earlier I frames, 只存储不同之处
    - B frames (bidirectional predicted picture): depending on earlier and later frames (可以是I也可以是P)

---

## 11. Applications: World Wide Web

- Separation of duties 【s11 p4】
  - network
  - application
- Typical application protocols 【s11 p7】
  - HTTP (WWW)
  - FTP (file transfer)
  - SMTP, POP, IMAP (emails)
  - SSH, Telnet (remote shell)
  - RDP, VNC (remote desktop)



## World Wide Web (HTTP)

- HTML (HyperText Markup Language) 【s11 p11】
  - URL (Uniform Resource Locator) 【s11 p12】
  - URL and MIME types 【s11 p15】
  - HTTP (HyperText Transfer Protocol) 【s11 p16】
    - default server port: 80
    - HTTP request message 【s11 p18】
    - HTTP request methods 【s11 p21】
    - HTTP response codes 【s11 p23】
  - HTTPS 【s11 p25】
  - Cookies 【s11 p26】
    - to identify the client
    - private mode
  - Documents 【s11 p28】
    - Static: fast, but not customized
    - Dynamic: slower, but more flexible
  - Client side elements
    - HTML5
  - AJAX (Asynchronous Javascript and XML)
  - Web services: XML + HTTP 【s11 p31】
    - XML-RPC
    - example: Google search API
  - HTTP server operation
  - Multi-tier servers
  - Browser operation 【s11 p37】
    - browser kernels (layout engines)
- 

## 12. Applications: DNS and Email

### Domain name service (DNS)

- DNS functionality 【s12 p5】
  - given name of a computer, return computer's IP address

- Example 【s12 p6】
- Configure of the network connection
  - IP address
  - netmasks
  - gateway
  - name server
- Domain name syntax 【s12 p7】
  - DNS: domain name  $\rightarrow$  IP address
  - ARP: IP address  $\rightarrow$  MAC address
- Top-level domains
- DNS client/server interaction 【s12 p14】
  - client known as resolver
- Name resolution 【s12 p15】
  - local name server不需要知道全世界所有的name所对应的IP地址，按照图中的顺序问就行，问过的存为cache，更快。
- Root servers 【s12 p18】
  - 13 root name servers
  - multiple server hosts
    - load balancing 【s12 p20】
    - content distribution
- Global server load balancing (GSLB) 【s12 p22】

## Internet Email (SMTP, POP, IMAP)

- Internet Email 【s12 p28】
  - Email address
  - Mail message format
- MIME (Multi-purpose Internet Mail Exchange) 【s12 p29】
  - sender inserts additional header lines to indicate the encoding method
- MIME encoding
  - quoted-printable
  - base64
- SMTP (Simple Mail Transfer Protocol) 【s12 p34】
- POP/IMAP 【s12 p35】

- POP is simpler than IMAP
  - POP3: one way, only transfers email to client
  - IMAP4: two way
  - Webmail example 【s12 p38】
- 

## 13. Other applications

- Modern applications
  - multimedia
  - peer-to-peer
  - blockchain
  - storage networks

### Multimedia

- Key issues 【s13 p4】
  - data volume, realtime transmission
- Solutions
  - compression, realtime scheduling, p2p

### Peer-to-Peer

- 对等 · 参与者本质上无区别 · different from client/server. 【s13 p5】
- Tracker and torrent
- Example: BitTorrent 【s13 p7】
  - tit-for-tat
  - optimistically unchoke
- Peer-to-Peer applications 【s13 p10】
  - application areas: content distribution, document management, multimedia
- Example: Bitcoin and Blockchain 【s13 p12】
  - de-centralized ledger
  - reliability requirements and solutions 【s13 p15】
    - voting based on computation power
    - ledger structure: blockchain
  - Hash function 【s13 p16】
  - Blockchain 【s13 p17】
    - Nonce value, "mining"
    - keep longer branch, discard the shorter one

### Storage networks

- Network Attached Storage (NAS) 【s13 p21】
- Storage Area Networks (SAN) 【s13 p22】

## Cloud computing

- Cloud computing service model 【s13 p24】
    - SaaS
    - PaaS
    - IaaS
  - Typical services
  - Key technologies 【s13 p26】
    - Storage
    - Distributed computing
    - Virtualization
- 

## 14. Security

- Network security 【s14 p3】
  - confidentiality
  - authentication
  - message integrity
  - access and availability
- Security threats and methods 【s14 p5】
  - Information interception: Encryption
  - Virus/Worms: Pattern scanning
  - Trojan Horses: Firewalls, pattern scanning
  - Intrusion: Passwords, detection
  - DoS/DDoS attacks: No good solutions
  - Session Hijacking: Authentication/Encryption
  - Natural Disasters: Redundancy
- Information interception
- Asymmetric encryption 【s14 p8】
  - Public/Private key
  - Alice存有自己的private key · 给Bob发Alice的public key, private key可以解密用对应public key加密的信息 · 而Bob反过来要给Alice签名 · 则需要用Bob自己的private key。
  - Email encryption 【s14 p12】
- Secure Sockets Layer (SSL) 【s14 p15】

- uses asymmetric encryption to privately share the session key; uses **symmetric** encryption to encrypt data
  - SSH 【s14 p18】
    - SSH Tunnel 【s14 p20】
- 

## 15. Review

- How to connect two or more computers? 【s14 p25】
  - How to connect networks? 【s14 p26】
  - How can so many computers share the same link resources? 【s14 p27】
    - Multiplexing
    - Packet switching
  - How can we build the applications 【s14 p28】
- 

## Questions

s1

- What is packet switching? What are the advantages of packet switching?
  - Packet Switching transmits data across digital networks by breaking it down into blocks or packets for more efficient transfer using various network devices. Each time one device sends a file to another, it breaks the file down into packets so that it can determine the most efficient route for sending the data across the network at that time. The network devices can then route the packets to the destination where the receiving device reassembles them for use.
- What is layering?
  - Layering is the practice of arranging functionality of components in a system in a hierarchical manner such that lower layers provide functions and services that support the functions and services of higher layers. The end user in a layered system interacts only with the top-most layer. The lowest layer typically interacts with physical world to perform the task requested by the end user.
- Why is layering useful?
  - Using layering means using abstractions to hide the complexity of the computer networks. Keep networks simple.
  - Layering is useful in computer networking because it provides an effective way to organize all the functionality required for data communication. Each layer is responsible for a specific function in data communication and developers of each layer of the software can focus on improving how that specific functionality is delivered.
  - A side benefit of layering is that it enables up gradation of each layer independently of other layers. For example, you can upgrade a version of your web browser or switch to a different web

browser without changing any other network software. You can also easily move between wired and wireless computer networks without changing browsers or email clients. Again, this is possible because the network access functionality is defined in a layer independent of the application layer.

- What are the primary functions of each layer?
  - Application layer: commands to support common end user network needs
  - Transport layer: reliable end-to-end data delivery
  - Network layer: routing and network addressing
  - Data-link layer: error-free transmission over 1 link
  - Physical layer: converts data to signals for transmission over physical media

s3

- What is the significance of the delay  $\times$  bandwidth product?
  - The delay  $\times$  bandwidth product is important to know when constructing high-performance networks because it corresponds to the amount of data the sender can send out before receiving a response.
- If you want to connect your home to the Internet, what are the common access technologies available for you to choose from?
  - Fiber, cable, Wi-Fi, Cellular...
- Why is the sliding window transmission "reliable"?
  - reliability: delivery, order.
  - The key idea is to limit the amount of packets that can be sent ahead of an acknowledgment. So for a Window Size of  $W$ , the transmitter is limited to send up to  $W$  packets before it needs to wait for acknowledgments.
  - In addition, the Sliding Window acts as a simple flow control and prevents network overload by bounding the number of packages sent. Although that is not the main motivation for this modification.
- Why should the sender window size (SWS) be set to bandwidth $\times$ delay product estimate?
  - To maximize the utilization (throughput). The throughput that this protocol achieves depends on the window size

s4

- Why is Ethernet more popular than Token Ring in LANs?
  - Ethernet is simpler and less costly.
  - Ethernet was the first widely deployed high-speed LAN. Therefore, network administrators were reluctant to switch over to other LAN technologies.
  - Ethernet was always producing versions that operated at equal data rates or higher than substitutes.

- because Ethernet has been so popular that Ethernet hardware has become a commodity and is remarkably cheap.
- Differences between Ethernet and Token Ring.
  - Token Ring: In the token ring a token ring passes over a physical ring. Token ring is defined by IEEE 802.5 standard. In token ring, there is a station and a special frame called token. A station in token ring can transmit data frame if it contains a token. After the successful transmission of data frame token are pointed(issued). Token ring is a Star shaped topology and handles priority in which some nodes may give priority to the token.
  - Ethernet: IEEE 802.3 defines the Ethernet. It uses CSMA/CD mechanism. It means that if many stations exist at the same time to talk, all stations will be closed. To resume them, wait for a random time. Unlike token ring it doesn't employ any priorities. And it is less costly than token ring network.
- Why might a mesh (ad-hoc) topology be superior to a base station topology for communication in a natural disaster?
  - Because the mesh structure is easier to deploy and has better performance in changing environments.
  - Because there is no special base station node in a wireless mesh,a natural disaster could hardly affect the nodes.Meshes also offer fault tolerance by providing multiple routes for a message to get from point A to point B.
- How can an ethernet communicate with an FDDI network?
  - Inter-connecting links of different types: Gateways/routers, the Internet Protocol
  - Connecting heterogeneous networks 【s6 p8】
  - This relates to the internet protocol at the network layer. For inter-connecting different data links, we need the switch to convert between different data frame format. A switch that can play this role is called a router. Routers use the internet protocol to conduct the conversion.

s5

- What are the advantages and disadvantages of datagram switching?
  - connectionless networks 【s5 p17】
  - Ads: No connection establishment required. A switch or link failure is no big deal.
  - Disads: When a packet is sent, it is not known whether the network can deliver it (reliable transmission is not guaranteed). Each packet is forwarded independently.
- What are the advantages and disadvantages of virtual circuit switching?
  - Ads: Data are delivered in order. Reliable. Packets don't contain destination address and hence headers are very small.
  - Disads: If one of the device does not work properly or fails, the complete connections are lost. In case of crises, it won't be able to handle a lot of traffic. Load is not distributed in the network, some routers may be overused, while others unused.

- How can broadcast and multicast of messages be implemented in the extended LAN (connected with bridges)?
  - Broadcast is simple—each bridge forwards a frame with a destination broadcast address out on each active (selected) port other than the one on which the frame was received.
  - Multicast can be implemented in exactly the same way, with each host deciding for itself whether or not to accept the message. This is exactly what is done in practice.
- What is ATM? What do you think about the future of ATM?
  - Asynchronous transfer mode (ATM) is a switching technique used by telecommunication networks that uses asynchronous time-division multiplexing to encode data into small, fixed-sized cells.
  - This is different from Ethernet or internet, which use variable packet sizes for data or frames. ATM is the core protocol used over the synchronous optical network (SONET) backbone of the integrated digital services network (ISDN).
  - An ATM network is less adaptable to a sudden network traffic surge.
  - The ATM provides data link layer services that run on the OSI's Layer 1 physical links. It functions much like small-packet switched and circuit-switched networks, which makes it ideal for real-time, low-latency data such as VoIP and video, as well as for high-throughput data traffic like file transfers.
  - ATM is a core protocol used in the SONET/SDH backbone of the public switched telephone network (PSTN) and in the Integrated Services Digital Network (ISDN), but has largely been superseded in favor of next-generation networks based on Internet Protocol (IP) technology, while wireless and mobile ATM never established a significant foothold.

s6

- Why is IP able to connect various link technologies to form a larger internetwork?
  - Because data is contained in IP datagram, which is of uniform format. Moreover, the philosophy used in defining the IP service model was to make it undemanding enough that just about any network technology that might turn up in an internetwork would be able to provide the necessary service.
- What aspect of IP addresses makes it necessary to have one address per network interface, rather than just one per host?
  - IP addresses consist of two parts, usually referred to as a network part and a host part. Since a host may be connected to different networks, it is necessary to have one address per network interface.
- What are the challenges in connecting two physical networks into a single logical network?
- IP works as a "best-effort" service, that is, if a packet fails to reach its intended destination, the network does nothing. This indicates that IP packet delivery is not fully reliable. However, we know the data link layer has already addressed the reliable transmission issue with the "sliding window" algorithm. How does the data transmission become unreliable again when it comes to the IP protocol?



- Sliding window at the data link layer provides reliable delivery in a direct link network. When it comes to IP, packets are forwarded across multiple direct link networks, where the forwarding is not guaranteed to be reliable.
- IP addressing and physical (MAC) addressing?
  - The main difference between MAC and IP address is that, MAC Address is used to ensure the physical address of computer. It uniquely identifies the devices on a network. While IP address are used to uniquely identifies the connection of network with that device take part in a network.
- What aspect of IP addresses makes it necessary to have one address per network interface, rather than just one per host?
  - IP addresses include the network/subnet, so that interfaces on different networks must have different network portions of the address. Alternatively, addresses include location information and different interfaces are at different locations, topologically.
  - Point-to-point interfaces can be assigned a duplicate address (or no address), because the other endpoint of the link doesn't use the address to reach the interface; it just sends. Such interfaces, however, cannot be addressed by any other host in the network.

s8

- What are the differences between distance vector and link state routing approaches?
  - See the picture in word file.
- What are the limitations of the Network Layer?
  - See word.
- What is NAT?
  - To access the Internet, one public IP address is needed, but we can use a private IP address in our private network. The idea of NAT is to allow multiple devices to access the Internet through a single public address. To achieve this, the translation of private IP address to a public IP address is required. Network Address Translation (NAT) is a process in which one or more local IP address is translated into one or more Global IP address and vice versa in order to provide Internet access to the local hosts. Also, it does the translation of port numbers i.e. masks the port number of the host with another port number, in the packet that will be routed to the destination. It then makes the corresponding entries of IP address and port number in the NAT table. NAT generally operates on router or firewall.

s11

- Why the client's port numbers are different but the server's are the same? [s11 p6]
  - two clients can connect to same server port because for each client we can assign a different socket (as client IP will definitely differ). Same client can also have two sockets connecting to same server port - since such sockets differ by SRC-PORT.
  - A server socket listens on a single port. All established client connections on that server are associated with that same listening port on the server side of the connection. An established

connection is uniquely identified by the combination of client-side and server-side IP/Port pairs. Multiple connections on the same server can share the same server-side IP/Port pair as long as they are associated with different client-side IP/Port pairs, and the server would be able to handle as many clients as available system resources allow it to.

- It is a common question and quite mind-boggling, which is – How does one server handle multiple connections? Let's suppose that multiple clients have been simultaneously connected to port 80. This signifies that the "connection" actually represents a packet that consists of a header field containing the destination port, which is 80 in this case. There are many other ports that have different serial numbers to get identified. It is possible that multiple clients get connected to the same port depending on which client has been assigned which port. The sockets signify the exact end-point. It is quite an obvious fact that the IP address of each client will certainly differ, and so does the server's IP address. Ports are just for general source and destination purpose and will hence allow multiple connections, that in turn will be diverted to the exact servers where they are supposed to reach.
- On a server, a process is listening on a port. Once it gets a connection, it hands it off to another thread. The communication never hogs the listening port. Connections are uniquely identified by the OS by the following 5-tuple: (local-IP, local-port, remote-IP, remote-port, protocol). If any element in the tuple is different, then this is a completely independent connection.
- What made WWW so successful?
  - Web operates on demand
  - Hyperlinks and search engines help a lot
  - 信息获取、传输、共享
  - user interaction
- Why does HTTP run over TCP (rather than UDP)?
  - HTTP, FTP, SMTP and POP3 being application layer (layer 7) protocols can run on any of the underlying protocols, be it TCP/IP, IPX/SPX, UDP, or even on FCP.
  - TCP is the most sought after transmission protocol due to its reliability and ubiquity, such that it has now become the underlying protocol for almost all network transmissions. UDP being a connectionless transport protocol, there can be no guarantee of reliable data transmission.
  - Because TCP is more reliable, and HTTP, FTP, SMTP, and POP3 cannot be afforded using UDP while UDP cannot transmit packet and guarantee a well order delivery. TCP is a connected-oriented network so packet will be delivered to the destination.
- Why do HTTP, FTP, SMTP and POP3 run on top of TCP rather than UDP? from chegg.com
  - The protocols (HTTP...) require the application data to be received without any gaps in the **correct order**.
  - The TCP is more **reliable** than the UDP because TCP is a **connected-oriented network** where there is a guarantee of the transmitted packet in reaching the destination.
  - UDP sends only the datagram and it does not stand in managing retransmission, data sequencing or the connection.
  - Here, to run on their data and for assured delivery, all the data streams want the reliable protocol. The packet loss or failure during the transmission cannot be decided in the UDP protocol.

## s12

- Is it possible to map multiple domain names to one IP address? If yes, what kind of problems should be addressed?
  - Yes. It is called a Shared Web Hosting:
    - In name-based virtual hosting, also called shared IP hosting, the virtual hosts serve multiple hostnames on a single machine with a single IP address. This is possible because when a web browser requests a resource from a web server using HTTP/1.1 it includes the requested hostname as part of the request. The server uses this information to determine which web site to show the user. When you register/purchase your domain name on a particular "registrars name server", your DNS settings are kept on their server, and in most cases point your domain to the Name Server of your hosting provider. This Name Server is where the IP number (currently associated with your domain name) resides.

## s13

- Why are distributed computing and virtualization important in cloud computing?
  - 云计算的核心是资源的网络化共享、应用，对应用者来讲实际上就是一种资源映射，透明的资源映射就是虚拟技术（就是你能像用本地资源一样用网络资源）。拟化是指计算机元件在虚拟的基础上而不是真实的基础上运行。虚拟化技术可以扩大硬件的容量，简化软件的重新配置过程。CPU的虚拟化技术可以单CPU模拟多CPU并行，允许一个平台同时运行多个操作系统，并且应用程序都可以在相互独立的空间内运行而互不影响，从而显著提高计算机的工作效率。
  - 极大地降低了云计算系统应用程序的开发难度且提高了云计算系统的并行度和运行效率，这就是云计算的分布式计算系统
  - 分布式的优势：更高的可扩展性、高容错和更高的冗余利用率、更低延时、更高的成本效益、高效