主要内容

- 端到端设计原理
- Internet的设计原则
- Internet的网络层（复习）

- 两篇经典论文
  - Saltzer, Jerome H., David P. Reed, and David D. Clark. "End-to-end arguments in system design." ACM Transactions on Computer Systems (TOCS) 2.4 (1984): 277-288.
  - Clark, David. "The design philosophy of the DARPA Internet protocols." Symposium proceedings on Communications architectures and protocols. 1988.

端到端设计原理

- 端到端原则（ End-to-end ）
  - 数据通信网络（data communication network）是计算机系统的一个重要组成部分
  - 如何划分计算机系统各个组成部分的功能边界？

Saltzer J H, Reed D P, Clark D D. End-to-end arguments in system design[J]. ACM Transactions on Computer Systems (TOCS), 1984, 2(4): 277-288

End-to-end Arguments in System Design

Saltzer J H, Reed D P, Clark D D. End-to-end arguments in system design[J]. ACM Transactions on Computer Systems (TOCS), 1984, 2(4): 277-288.

争论什么？ (The Argument)

- 应用的需求：
  - Define when it is applicable

  "The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system..." *应用需求+通信端点*
- 主机端（Endpoint）的优势：
  - 不考虑网络通信的细节
  - 在主机端进行验证：*correct operation can only be verified by endpoints.*

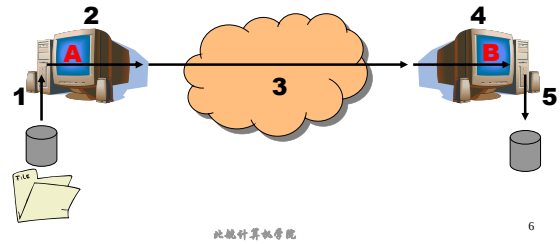
The Argument

- 结论
  - *"... Therefore, providing that questioned function as a feature of the communication system itself is not possible..."*
- 例外情况：Define the exception
  - *"... (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)"*

增强性能

实例1：文件传输

- Copy/Move file from HD on Computer A to HD on Computer B



传输中可能出现的问题

- 例如:
  1. Disk error
  2. Software error (OS, File transfer program, Network driver)
  3. Hardware error
  4. Communication system
  5. System crash

如何解决?

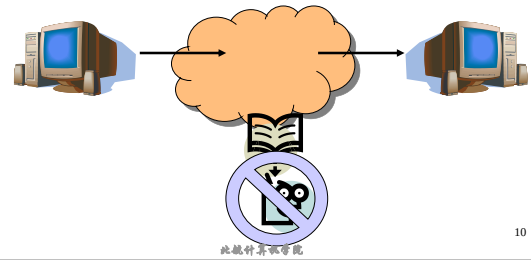
- Solution 1: Point-to-Point点到点
  - Reinforce each step of process (duplicate copies , timeout, retry, etc.)
  - 目标: Reduce probability of each threat to an acceptably small value
  - Could be hard to do, each step must be full-proof（充分证明）
  - Could be inefficient, extra checking
- Solution 2: End-to-End 端到端
  - "end-to-end check and retry"
  - Checksum → transfer file → receive the file → compute checksum → send checksum to originator to compare the two checksums.
  - If check fails, redo from beginning

如何解决？（续）

- Solution 3: Both
  - Point-to-Point checks in communication system (such as link level, IP, and/or TCP)
  - End-to-End checks must still be performed, since only one of the threats is handled
  - 不能降低应用的开销，但可能降低故障概率
- 教训Lesson:
  - Application must supply the guarantee in the end
    应用系统必须提供端到端的保证

安全的数据传输

- 加密Encryption：Goal, move data from one machine to another such that the data on the wire is secure (encrypted)



安全的数据传输（续）

- 由通信系统在入口进行加密，在出口进行解密
- 存在问题：密钥分发？
  - Communication system needs the key
  - Data is in the clear（明文） when entering/exiting
  - Authenticity must still be checked by application
- End-To-End argument wins here

性能

- 考虑一些特殊情况：
  - communication system is very unreliable, file transfer could keep retrying for ever because one packet got lost!
- 折中trade-off：Providing more reliability at the lower layers is a tradeoff between *cost* & *engineering effort* vs. *reliability*
- Not a simple decision
- *应用无法感知底层系统状态*

识别主机（End Hosts）

- Maybe not so easy...
- Consider voice over IP
  - 端系统是计算机？
    - Could introduce long delays
  - 端系统是人？
    - Retry = "repeat that"
- *End-to-End argument is not an absolute, but a design tool（guideline）*

思考

- **端到端原则**：In layered design, the E2E principle provides guidance on where functions belong.
  - "Dumb, minimal" network and "intelligent" endpoints.
- **对网络发展的影响**：E2E principle allowed the Internet to grow rapidly because innovation took place at the edge, in applications and services.
  - **新型应用**：Ex. WWW, Skype, BitTorrent, Bitcoin
  - **网络功能**：NATs, firewalls, VPN tunnel endpoints
  - **网络核心**： multicast, mobility, QoS

The Design Philosophy of the DARPA Internet Protocols

ACM SIGCOMM Computer Communication Review 18.4 (1988): 106-114.

David D. Clark (MIT)

- Since the mid 70s, Dr. Clark has been leading the development of the Internet;
- from 1981-1989 he acted as Chief Protocol Architect in this development, and chaired the Internet Activities Board.

- At the time of writing (1987)...
  - (Almost) no commercial Internet
  - 1 yr after Cisco's 1st product, IETF started
  - Number of hosts reaches 10,000
  - NSFNET backbone 1 year old; 1.5Mb/s

https://www.csail.mit.edu/person/david-clark

Internet 体系结构（Architecture）

- 基本目标: Effective network interconnection
- 二级目标（优先级）：
  1. 可生存性：Continue despite loss of networks or gateways
  2. Support multiple types of communication service
  3. Accommodate a variety of networks
  4. Permit distributed management of Internet resources
  5. Cost effective
  6. Host attachment should be easy
  7. Resource accountability

优先级

- 相关技术
  - 分组交换 Packet switching
  - 共享Fate Sharing/软状态Soft state
- 早期的设计目标对目前的Internet仍产生影响
  - E.g., resource accounting（计费） is a hard, current research topic

基本目标

- 网络互连

  “technique for multiplexed utilization of existing interconnected networks”
  - 信道复用（共享sharing）
    - Multiplexing (Shared use of a single communications channel
      - TDMA, FDMA, CDMA，statistical multiplexing（统计复用）
  - 互连现有网络 (互连 interconnection)
    - Tries to define an “easy” set of requirements for the underlying networks to support as many as possible

数据报交换（Datagram Switching）

- 分组交换
  - 分组包含转发所需的信息：地址
- 无状态：No state established ahead of time (helps fate sharing)
- 协议：Basic building block – must build things like TCP on top
- Pretty much implies statistical multiplexing（统计时分复用）
- 其他技术:
  - 电路交换Circuit Switching
  - 虚电路Virtual Circuits
  - 源路由Source routing

连接各种网络

- 不同类型的网络
  - ARPANET, X.25 networks, LANs, satellite networks, packet networks, serial links...
- 网络之间的差异
  - Address formats
  - Performance – bandwidth/latency
  - Packet size
  - Loss rate/pattern/handling
  - Routing

Goal 1: Internet communication must continue despite loss of networks or gateways.

1. "Entities should be able to continue communicating without having to re-establish or reset the high level state of their conversation."
2. "The architecture [should] mask completely any transient failure."

Leads to:

1. "Fate-sharing" model - only lose communication state if the end-host is lost.
2. Stateless packets switches => datagrams

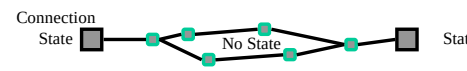
可生存性

可生存性Survivability

- If network disrupted and reconfigured
  - Communicating entities should not care!
  - No higher-level state reconfiguration
- How to achieve such reliability?
  - Where can communication state be stored?

	Network	Host
Failure handing	Replication	"Fate sharing"
Net Engineering	Tough	Simple
Switches	Maintain state	Stateless
Host trust	Less	More

Fate Sharing



- 当且仅当端节点（实体）失败，才能丢失状态信息
- 例如:
  - 主机崩溃：TCP状态丢失
    - 路由器重启：无状态
  - 目前网络现状
    - NATs and firewalls
- 折中：Survivability compromise
  - 异构网络Heterogeneous network：端主机存储较少信息；网络恢复机制

软状态Soft-state

- Soft-state
  - Announce state
  - Refresh state
  - Timeout state
- 计时器超时：性能急剧下降
- 流标识：Robust way to identify communication flows
  - Possible mechanism to provide non-best effort service
- 改善可生存性（survivability）

Goal 2: 异构服务Heterogeneous Services

- TCP/IP 的传输层
  - TCP for flow control, reliable delivery
  - IP for forwarding
- 异构的服务：可靠 vs. 不可靠的传输服务
  - Example: Voice and video over networks
  - Example: DNS
  - Why don't these applications require reliable, in-order delivery?
  - Narrow waist: allowed proliferation of transport protocols

服务类型（Types of Service， TOS）

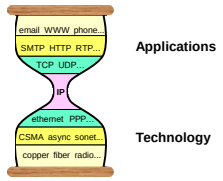
- TCP vs. UDP
  - 弹性应用需要可靠性保证
    - remote login or email
  - 非弹性应用：Inelastic, loss-tolerant apps
    - real-time voice or video
  - 其他应用：不同需求
  - 时延变化对可靠传输的影响
    - Today's net: ~100ms RTT
    - Reliable delivery can add *seconds*.
- 早期Internet 模型："TCP/IP" one layer
  - First app was remote login...
  - But then came debugging, voice, etc.
  - These differences caused the layer split, added UDP

Goal 3: 各种网络Varieties of Networks

- 多种网络互连
  - ARPANET, X.25 networks, LANs, satellite networks, packet networks, serial links...
- 最小集合假设
  - 包大小： Minimum packet size
  - 可靠性： Reasonable delivery odds, but not 100%
  - 寻址： Some form of addressing unless point to point
- Important non-assumptions:
  - Perfect reliability
  - Broadcast, multicast
  - Priority handling of traffic
  - Internal knowledge of delays, speeds, failures, etc.
- Much engineering then only has to be done once

体系结构设计?

- Need to interconnect many existing networks
- Hide underlying technology from applications
- Decisions:
  - Network provides minimal functionality
  - *"Narrow waist"*



Tradeoff: No assumptions, no guarantees.

局限性

- IP over anything, anything over IP
  - Has allowed for much innovation both above and below the IP layer of the stack
  - An IP stack gets a device on the Internet
- 缺点:
  - difficult to make changes to IP
  - But...people are trying (下一代互联网 GENI)
  - Only a small amount of information available about lower levels. (如 无线网络，wireless)

Goal #4: 分布式管理

- Independently managed as a set of independent "Autonomous Systems" 自治系统AS
  - ISPs
- BGP (Border Gateway Protocol) connects ASes together
  - Completely (well...) decentralized routing
  - Is this a good thing?

管理的问题

- 缺乏有效的管理工具
  - "Some of the most significant problems with the Internet today relate to lack of sufficient tools for distributed management, especially in the area of routing."
- Internet 的庞大规模
  - 18,000 constituent networks
  - Routing tables with 1,000,000+ entries
  - Gajillions of $$.
- 管理开销
  - Management and operational expenses becoming increasingly important
  - 流量，用户，链路

Goal #5: 成本效益Cost Effectiveness

- 分组交换的开销：Packet headers introduce high overhead
  - but so does circuit setup（电路交换）
- 重传开销：End-to-end retransmission of lost packets
  - Potentially wasteful of bandwidth by placing burden on the edges of the network
- 有争议的折中
  - 存储冗余：Current trends are to exploit redundancy even more.
  - 带宽：Bandwidth is becoming cheaper in many environments

Goal #6: Ease of Attachment

- IP： Anything with a working IP stack can connect to the Internet (hourglass model)
- 不断支持新型应用
  - 移动通信，物联网...

代价：

Tradeoff: Burden on end systems/programmers.

Goal #7:可计量 Accountability

- Huge problem
  - 可管理，可计量
- Accounting 记账
  - Billing? (mostly flat-rate 固定费率. But phones are moving that way too - people like it!)
  - Inter-provider payments（ISP之间）
    - 各种复杂因素
- 网络安全：Accountability and security
  - 病毒 Worms, viruses, etc.
    - Partly a host problem. But hosts very trusted.
  - 认证Authentication
    - privacy vs. security.

作者的结论

- 数据报的适用性
  - “Datagram” good for most important goals, but poor for the rest of the goals.

  思考：处理效率，应用语义，策略
- 数据包（分组）处理
  - Processing packets in isolation, resource management, accountability all hard.

  其他数据结构：“cell”，“flow”，“class”.....
- Anticipates flows and “soft-state” for the future.

未来的互联网?

- 数据报（Datagram）是否是合适的抽象？
  - resource management, accountability, QoS
- 流 flow 的作用 (IPv6)
  - 如何定义流？
  - 流状态：routers require to maintain per-flow state
- 状态管理state management
  - recovering lost state is hard
- 软状态"soft state"!
  - soft-state: end-hosts responsible to maintain the state

二十年后.....

Blumenthal M S, Clark D D. Rethinking the design of the Internet: the end-to-end arguments vs. the brave new world[J]. ACM Transactions on Internet Technology (TOIT), 2001, 1(1): 70-109.

- 在端到端原则提出的20年后， David D. Clark 总结了互联网发展中遇到的主要问题

Rethinking Internet Design

What's changed?

- 安全性：operation in untrustworthy world
  - endpoints can be malicious
  - If endpoint not trustworthy, but want trustworthy network -> more mechanism in network core
  - Trust and security a big issue today!
- 应用多样性：more demanding applications
  - end-end best effort service not enough
  - new service models in network (Intserv, Diffserv)?
  - new application-level service architecture built on top of network core (e.g., CDN, p2p)?
  - wireless and mobility

Rethinking Internet Design …

What's changed? 违反端到端原则

- 服务提供商：ISP service differentiation
  - ISP doing more (than other ISPs) in core is competitive advantage
- 第三方监管：rise of third party involvement
  - Interposed between endpoints (even against will)
  - e.g., Chinese government, US recording industry
- 新技术：new technologies (wireless, optical ...)
- 新设备：limited capability devices (e.g., PDA, smart phones, sensors, ...), or perhaps also less "sophisticated" users

技术变革

- Add functions to the network core ("middleboxes" 中间盒技术，网络功能NF):
  - filtering firewalls
  - application-level firewalls, web caches and proxies
  - NAT boxes
  - active networking
  - ...
- 基础服务：Add "infrastructure services"
  - e.g., DNS,
  - (application-specific) content distribution networks (CDNs)

思考

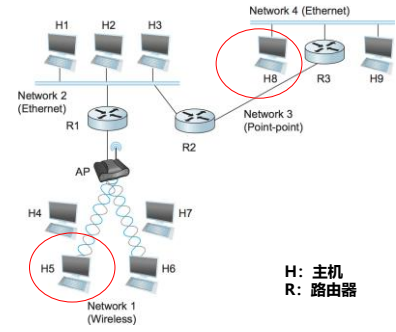
- Internet成功的原因？
- 商业网络设计如何考虑优先级？
- 端到端原则（E2E）设计的网络的特点？
  - 网络："Dumb, minimal" network；
  - 主机端："intelligent" end-points.
- 端到端原则为什么能支持应用创新？
  - WWW，即时通信，流媒体，P2P，Bitcoin
  - 云计算，物联网
- 端到端原则对目前网络应用有什么影响？
  - NATs, firewalls, VPN tunnel
  - 可信端用户→ 网络自我保护

基础知识回顾

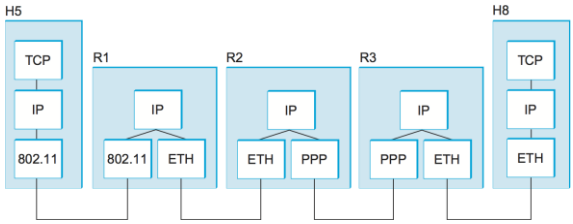
- 网络互连
- IP分组转发
- 子网划分
- CIDR

网络互连

- 场景：

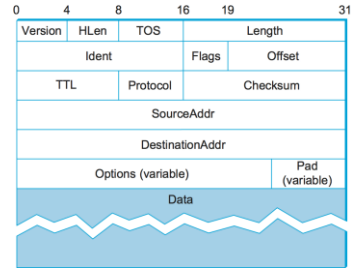


主机H5和主机H8之间通信



IPv4分组格式

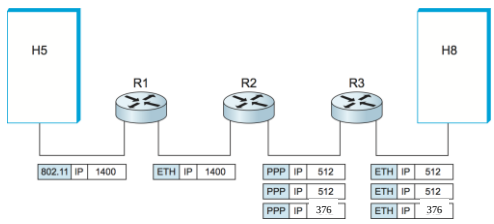
◆各字段的含义是什么？

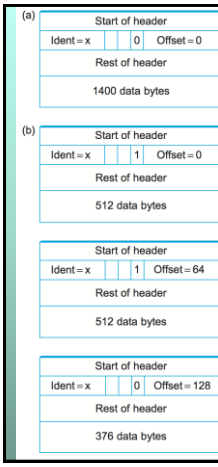


IP分组的传输过程

◆IP分组分片

- 假设以太网和802.11的MTU是1500字节，PPP的MTU是532字节，H5发送一个1420字节的IP分组，如何分片？
- 分片在哪里进行重组？

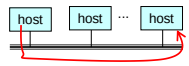




Header fields used in IP fragmentation:
(a) unfragmented packet;
(b) fragmented packets

如何转发帧?

- 共享传输介质的网络（物理层广播）
  - Forward all frames on the shared media
  - Adapter grabs frames with matching dest address



- 多跳交换的网络（第二层转发）
  - Flood every frame over every link?
  - Learn where the MAC address is located?

MAC 地址

- 平面名字空间：48 bits
  - Typically written in six octets in hex
  - E.g., 00-15-C5-49-04-A9 for my Ethernet
- 组织唯一标识符（Organizationally unique identifier）
  - Assigned by IEEE Registration Authority
  - Determines the first 24 bits of the address
  - E.g., 00-15-C5 corresponds to "Dell Inc"
- 地址分配：
  - Allocated by the manufacturer
  - E.g., 49-04-A9 for my Ethernet card

MAC 地址

- 特点
  - Persistent identifier (well, except for spoofing)
  - Mobile hosts are easy to handle
  - Forwarding-table look-up is a simple match
- 限制
  - Large forwarding tables in the data plane
  - Flooding overhead to learn location information
  - Lack of privacy

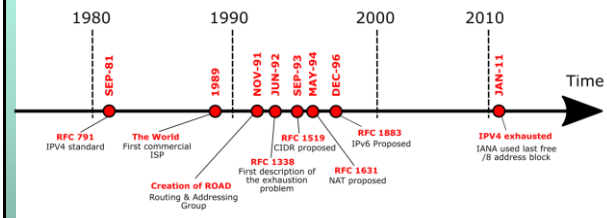
如何实现寻址的可扩展性?

- MAC addresses are flat（扁平地址）
  - Multiple hosts on the same network
  - No relationship between MAC addresses
- 数据平面 Data plane
  - Forwarding based on MAC address
  - Table size? Look-up overhead?
- 控制平面 Control plane
  - Determining where the host is located
  - Keeping the information up-to-date

IP分组转发需要解决的问题

- 是否Internet上的每个主机都有任意的、唯一的地址标识？
  - Would it scale?
- 分层结构（hierarchy）是否可扩展？
  - Tying the addressing to the topology & routing?
- 移动主机的地址?
- 谁来分配地址?
  - Network provider? Device manufacturer?
- 发送方认证自己，还是接收方？
  - What about spoofing and impersonation?

IPv4 历史

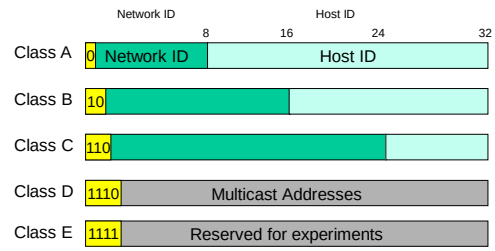


IP 地址

- 固定长度: 32 bits
- 有结构：Initial classful structure (1981)
- IP地址空间: 4 billion
  - Class A: 128 networks, 16M hosts
  - Class B: 16K networks, 64K hosts
  - Class C: 2M networks, 256 hosts

High Order Bits	Format	Class
0	7 bits of net, 24 bits of host	A
10	14 bits of net, 16 bits of host	B
110	21 bits of net, 8 bits of host	C

IP 地址类别

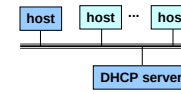


IP地址类别

- Class A: 0*
  - Very large /8 blocks (e.g., MIT has 18.0.0.0/8)
- Class B: 10*
  - Large /16 blocks (e.g,. Princeton has 128.112.0.0/16)
- Class C: 110*
  - Small /24 blocks (e.g., AT&T Labs has 192.20.225.0/24)
- Class D: 1110*
  - Multicast groups
- Class E: 11110*
  - Reserved for future use

如何获得IP地址?

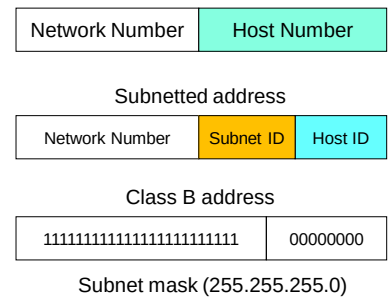
- ISP如何得到IP地址块?
  - From Regional Internet Registries (RIRs)
    - ARIN (North America, Southern Africa), APNIC (Asia-Pacific), RIPE (Europe, Northern Africa), LACNIC (South America)
- 主机如何获得IP地址?
  - 手工配置：Hard-coded by system admin in a file
  - 动态配置：DHCP（Dynamic Host Configuration Protocol）：dynamically get address: "plug-and-play"
    - Host broadcasts "DHCP discover" msg
    - DHCP server responds with "DHCP offer" msg
    - Host requests IP address: "DHCP request" msg
    - DHCP server sends address: "DHCP ack" msg



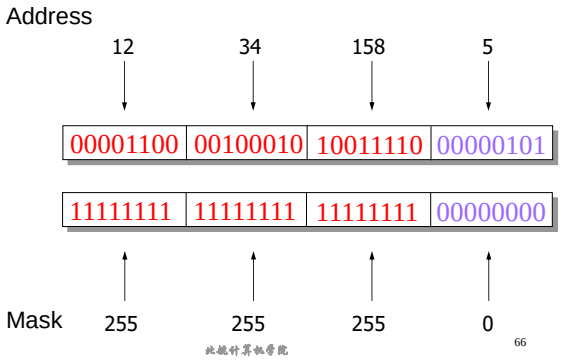
子网划分 Subnet Addressing -RFC917 (1984)

- Class A & B networks too big
  - Very few LANs have close to 64K hosts
  - For electrical/LAN limitations, performance or administrative reasons
- Need simple way to get multiple "networks"
  - Use bridging, multiple IP networks or split up single network address ranges (subnet)
- 子网划分的基本思想：
  - 将一个网络号分配给多个物理网络，每个物理网络为一个子网。

子网划分

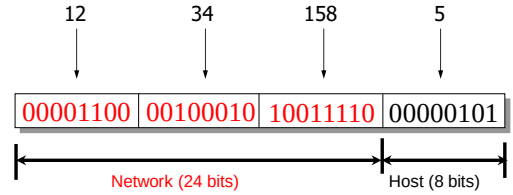


例：IP地址和子网掩码



IP地址的前缀（Prefixes）表示

- IP地址：网络号+主机号
- 例：12.34.158.0/24表示
  - 地址前缀（prefix）：24-bit
  - 地址空间：2^8

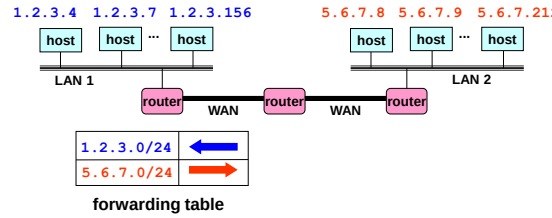


比较MAC地址和IP地址

	MAC	IP
分配	Hard-coded in the adaptor	Configured or learned
大小	48 bits	32 bits (in v4)
结构	Flat	Hierarchical
可移植性	Constant over life of the adapter	Changes with time and location
用途	Delivery within a single network	Delivery across an inter-network

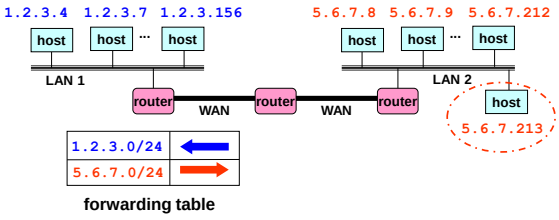
寻址：子网内

- 子网中的主机具有相同的网络前缀，例如：
  - 1.2.3.0/24 on the left LAN
  - 5.6.7.0/24 on the right LAN



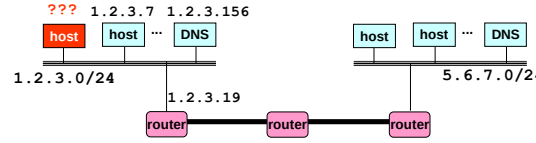
增加新主机

- 不需要更新路由器配置
  - E.g., adding a new host 5.6.7.213 on the right
  - Doesn't require adding a new forwarding entry



如何寻址?

- 问题
  - 问题1：分组如何到达主机？
  - 问题2：如何在链路上发送分组？
- 相关协议
  - ARP协议：根据IP地址确定主机的MAC地址（Link Layer addressing）
  - DHCP协议：动态获得IP地址？

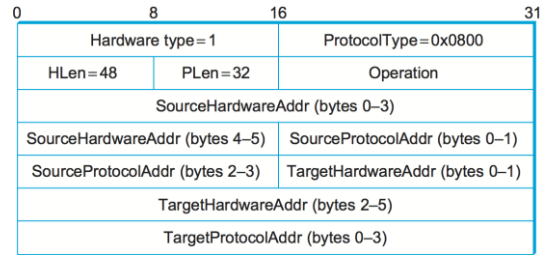


基本思想

- 广播Broadcasting：ARP，DHCP...
  - Broadcast query to all hosts in the local-area-network
  - ... when you don't know how to identify the right one
- 缓存Caching：ARP，DNS
  - Store the information you learn to reduce overhead
  - Remember your own address & other host's addresses
- 软状态Soft state：Timer
  - Associate a time-to-live field with the information
  - ... and either refresh or discard the information
  - Key for robustness in the face of unpredictable change

ARP：Address Resolution Protocol

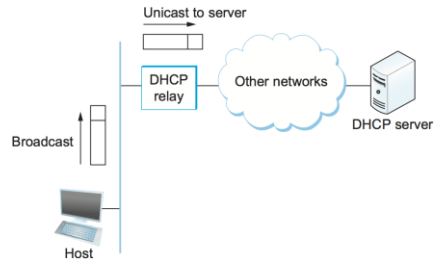
- ARP：IP地址和MAC地址之间的映射
  - 请求和响应的过程？



DHCP：Dynamic Host Configuration Protocol

◆**动态主机配置协议** (DHCP)

❖ 如何配置主机地址信息？



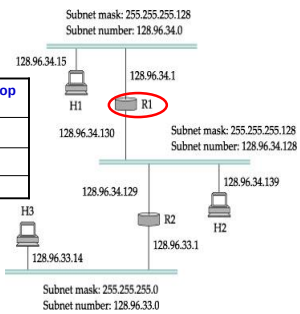
DHCP分组格式

Operation	HType	HLen	Hops
Xid			
Secs		Flags	
ciaddr			
yiaddr			
siaddr			
giaddr			
chaddr (16 bytes)			
sname (64 bytes)			
file (128 bytes)			
options			

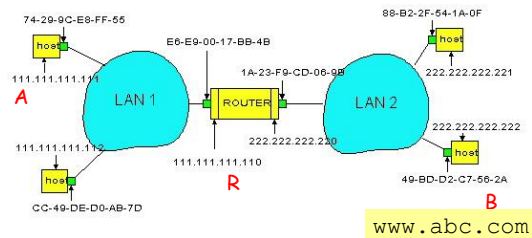
https://www.ietf.org/rfc/rfc2131.txt

例1：路由器R1的转发表

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	接口0
128.96.34.128	255.255.255.128	接口1
128.96.33.0	255.255.255.0	R2



例2: A向B发送分组的过程

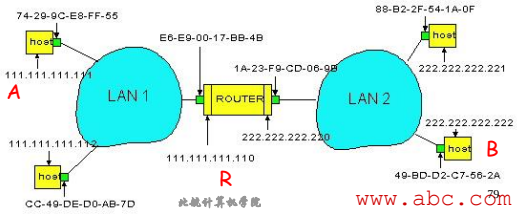


www.abc.com

A sends packet to R, and R sends packet to B.

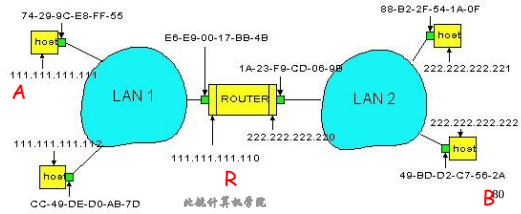
基本步骤

- 主机A获得主机B的IP地址：通过 DNS
- 主机A利用路由器R与外部主机通信
- 主机A发送帧到R的MAC地址
- 路由器 R 转发IP分组到输出端口（outgoing interface）
- 路由器R学习B的MAC地址，并转发帧



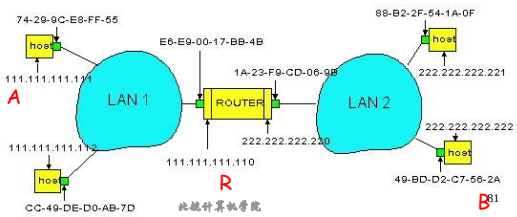
主机A获得主机B的IP地址

- Host A does a DNS query to learn B's address
  - Suppose gethostbyname() returns 222.222.222.222
- Host A constructs an IP packet to send to B
  - Source 111.111.111.111, destination 222.222.222.222



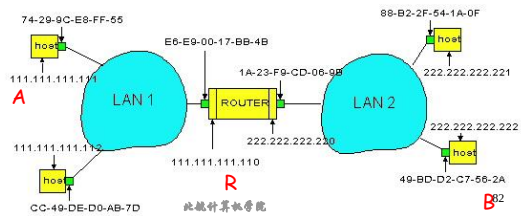
主机A决定发送分组

- IP header
  - From A: 111.111.111.111
  - To B: 222.222.222.222
- Ethernet frame
  - From A: 74-29-9C-E8-FF-55
  - To gateway: ????



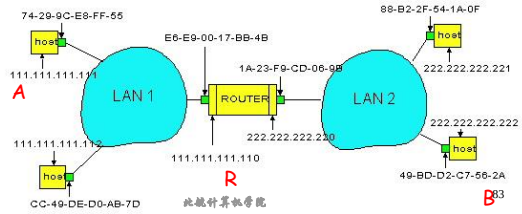
主机A准备向路由器R发送分组

- Host A has a gateway router R（主机A的路由配置）
  - Used to reach destinations outside of 111.111.111.0/24
  - Address 111.111.111.110 for R learned via DHCP
- 如何得到路由器的MAC地址？



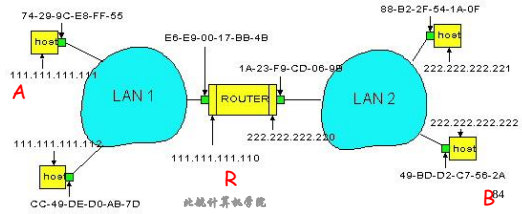
主机A 获得路由器R的MAC地址

- Host A learns the MAC address of R's interface
  - ARP request: broadcast request for 111.111.111.110
  - ARP response: R responds with E6-E9-00-17-BB-4B
- Host A encapsulates the packet and sends to R



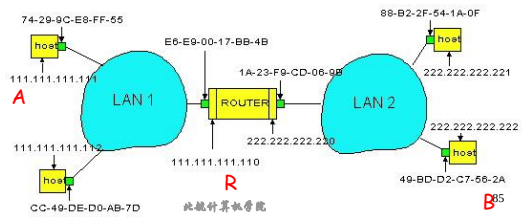
主机A向路由器R发送分组

- IP header
  - From A: 111.111.111.111
  - To B: 222.222.222.222
- Ethernet frame
  - From A: 74-29-9C-E8-FF-55
  - To R: E6-E9-00-17-BB-4B



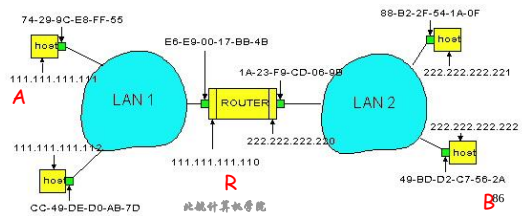
路由器R 决定如何转发分组

- Router R's adapter receives the packet
  - R extracts the IP packet from the Ethernet frame
  - R sees the IP packet is destined to 222.222.222.222
- Router R consults its forwarding table
  - Packet matches 222.222.222.0/24 via other adapter



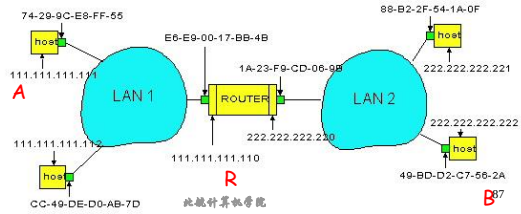
路由器R 决定如何转发分组

- IP header
  - From A: 111.111.111.111
  - To B: 222.222.222.222
- Ethernet frame
  - From R: 1A-23-F9-CD-06-9B
  - To B: ???



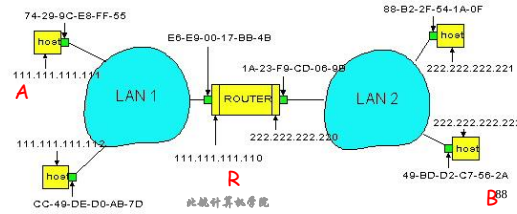
R 向 B发送分组

- Router R's learns the MAC address of host B
  - ARP request: broadcast request for 222.222.222.222
  - ARP response: B responds with 49-BD-D2-C7-56-2A
- Router R encapsulates the packet and sends to B



R 向 B发送分组

- IP header
  - From A: 111.111.111.111
  - To B: 222.222.222.222
- Ethernet frame
  - From R: 1A-23-F9-CD-06-9B
  - To B: 49-BD-D2-C7-56-2A



IP分组转发算法

```
D= destination IP address
for each forwarding table entry (SubnetNumber,
   SubnetMask, NextHop)
   D1=SubnetMask & D
   If D1 = SubnetNumber
        if NextHop is an interface
              deliver datagram directly to destination
        else
              deliver datagram to NextHop (a router)
```


几点说明

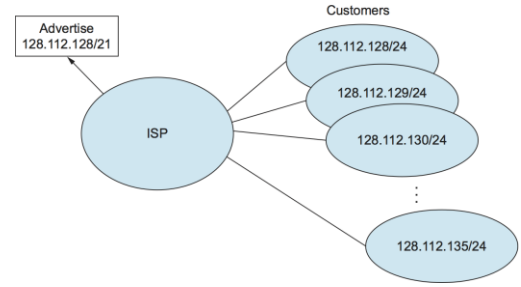
- 转发算法的问题
  - 通常应包含默认路由器（default router）
  - 重复进行目的地址与子网掩码的按位“与”运算
    - 线性表搜索，效率低
    - 子网掩码可能相同
- 子网划分的作用：解决可扩展性问题
  - 提高地址的分配效率
  - 地址信息汇聚：使多个物理网络共享一个地址
- 如何将多个物理网络的地址合并为一个地址？
  - CIDR（Classless Inter-Domain Routing）

Classless Inter-Domain Routing (CIDR) – RFC1338

- 更灵活的地址分配机制
  - 不使用类别（classes）确定网络号（network ID）
  - 使用一组地址的公共部分作为网络号（network number）
  - 例如：
    - 地址段：192.4.16 - 192.4.31可以表示为：192.4.16/20
      （16：00010000）
      （31：00011111）
- 问题：如何支持更高效的地址空间（和路由表）利用？
  - 多地址聚合为单一表项
  - 转发机制

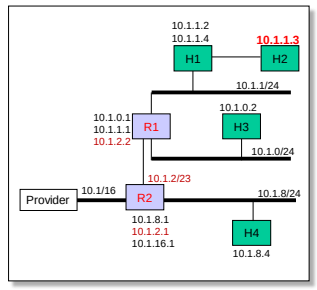
CIDR

- 如何用CIDR进行路由聚合？



案例：路由到网络

- Packet to 10.1.1.3 arrives
- Path is R2 – R1 – H1 – H2

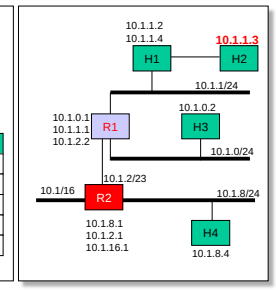


子网内的路由-1

- Packet to 10.1.1.3
- Matches 10.1.0.0/23

Routing table at R2

Destination	Next Hop	Interface
127.0.0.1	127.0.0.1	lo0
Default or 0/0	provider	10.1.16.1
10.1.8.0/24	10.1.8.1	10.1.8.1
10.1.2.0/23	10.1.2.1	10.1.2.1
10.1.0.0/23	10.1.2.2	10.1.2.1

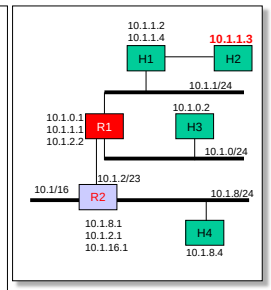


子网内的路由-2

- Packet to 10.1.1.3
- Matches 10.1.1.2/31
  - **最长前缀匹配（Longest prefix match）**

Routing table at R1

Destination	Next Hop	Interface
127.0.0.1	127.0.0.1	lo0
Default or 0/0	10.1.2.1	10.1.2.2
10.1.0.0/24	10.1.0.1	10.1.0.1
10.1.1.0/24	10.1.1.1	10.1.1.4
10.1.2.0/23	10.1.2.2	10.1.2.2
10.1.1.2/31	10.1.1.2	10.1.1.2

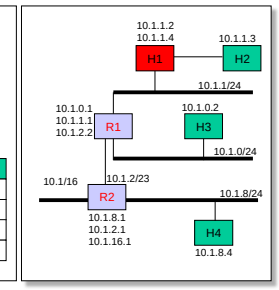


子网内的路由-3

- Packet to 10.1.1.3
- Direct route
  - **最长前缀匹配（Longest prefix match）**

Routing table at H1

Destination	Next Hop	Interface
127.0.0.1	127.0.0.1	lo0
Default or 0/0	10.1.1.1	10.1.1.2
10.1.1.0/24	10.1.1.2	10.1.1.1
10.1.1.3/31	10.1.1.2	10.1.1.2



CIDR 例子

- 例1 地址聚合：将8个C类地址块分配给某个网络：200.10.0.0 ~ 200.10.7.255
  - 地址块的前缀表示：201.10.0.0/21
- 例2 地址分配：ISP的地址块： 200.23.16.0/20，分配给8组织机构： 11001000 00010111 0001**0000** 00000000 (16个C地址块)

Organization 0 11001000 00010111 0001**0000** 00000000
200.23.16.0/23

Organization 1 11001000 00010111 0001**0010** 00000000
200.23.18.0/23

Organization 2 11001000 00010111 0001**0100** 00000000
200.23.20.0/23

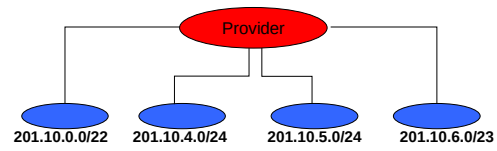
... .

Organization 7 11001000 00010111 0001**1110** 00000000
200.23.30.0/23

可扩展性: 地址聚合

Provider is given 201.10.0.0/21

可分配的C类地址空间：
201.10.0.0~201.10.7.0

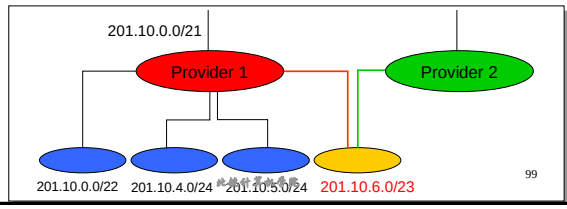


外部路由器仅需要知道如何到达201.10.0.0/21，由路由器（provider）直接将IP分组转发给适当的子网（customer）。

CIDR 对分组转发的影响

- 开销
  - CIDR 支持对有限地址空间的高效使用
  - 但是，增加了分组转发的复杂性
- 转发表（Forwarding table）可能有很多匹配项
  - 例如，表项为 201.10.0.0/21 和 201.10.6.0/23

IP地址 201.10.6.17 均可以匹配（为什么？）



99

IPv6

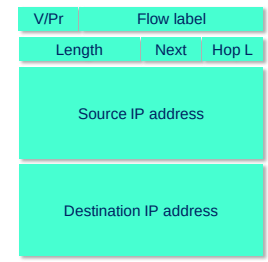
IPv6

- 动机：IETF从1991年开始看到IP地址空间耗尽问题，研究解决方法
- 下一代IP（IPng）→ IPv6，增加新特性
  - 地址空间：128bit
  - 支持实时服务
  - 安全性
  - 自动配置（IP地址和域名）
  - 增强路由功能（如移动主机等）
- 从IPv4到IPv6的过渡
  - 双栈操作
  - 隧道技术

101

IPv6

- 目标："Next generation" IP.
- 解决主要问题:增加地址空间
- 主要特点
  - 128位地址空间（16字节）
  - Multicast
  - Real-time service
  - Authentication and security
  - Auto-configuration
  - End-to-end fragmentation
  - Enhanced routing functionality, including support for mobile hosts



102

IPv6 的变化

- TOS字段： replaced with traffic class octet
- 流Flow
  - Help soft state systems
  - Maps well onto TCP connection or stream of UDP packets on host-port pair
- 容易配置
  - Provides auto-configuration using hardware MAC address to provide unique base
- 其他
  - Support for security
  - Support for mobility

103

103

IPv6 的变化

- 协议字段：replaced by next header field
  - Support for protocol demultiplexing as well as option processing
- 选项处理
  - Options are added using next header field
  - Options header does not need to be processed by every router
    - Large performance improvement
    - Makes options practical/useful

104

IPv6 地址

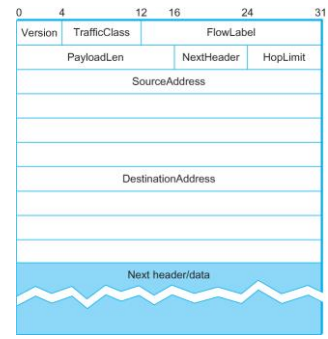
- Classless addressing/routing (similar to CIDR)
- 表示: x:x:x:x:x:x:x:x (x = 16位地址段的16进制表示)
  - 压缩连续的0: 47CD::A456:0124
  - 与IPv4 地址兼容: ::128.42.1.87
- 地址分配
  - provider-based
  - geographic

010	Registry	Provider	Subscriber	Sub Net	Host

105

IPv6 Header

- 基本头部：40字节
- 扩展头部
  - fragmentation
  - source routing
  - authentication and security
  - other options



106

IPv6 自动配置

- 无状态自动配置：Serverless ("Stateless")
  - Only configures addressing items, NOT other host things
    - If you want that, use DHCP.
- Link-local address
  - 1111 1110 10 :: 64 bit interface ID (usually from 48bit Ethernet addr)
    - (fe80::/64 prefix)
  - 唯一性测试 ("anyone using this address?")
  - 请求路由器 (solicit, or wait for announcement)
    - Contains globally unique prefix
    - Usually: Concatenate this prefix with local ID → globally unique IPv6 ID
- DHCP took some of the wind out of this, but nice for "zero-conf" (many OSes now do this for both v4 and v6)

从IPv4向 IPv6迁移

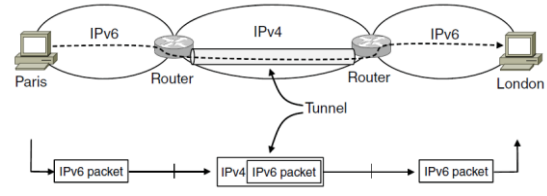
- Interoperability with IP v4 is necessary for gradual deployment.
- 几种机制
  - 双协议栈 Dual stack operation: IP v6 nodes support both address types
  - 转换 Translation:
    - Use form of NAT to connect to the outside world
    - NAT must not only translate addresses but also translate between IPv4 and IPv6 protocols
  - 隧道Tunneling: tunnel IP v6 packets through IP v4 clouds

ICMP协议

- Internet Control Message Protocol (ICMP)
  - 有哪些报文类型？
  - 作用？
- 有用的工具
  - Ping：ses ICMP echo messages to determine if a node is reachable and alive.
  - Traceroute：uses a slightly non-intuitive technique to determine the set of routers along the path to a destination

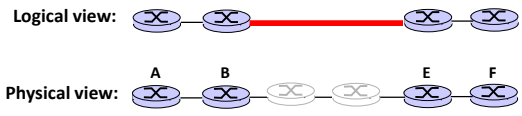
隧道技术（Tunneling）

- 如何处理两个不同网络互联？
- 场景：源主机和目的主机是同类型网络，中间隔着一个不同类型的网络



隧道技术（Tunneling）--续

- IP 隧道是一条虚拟的点到点链路
  - 增加IP头部，封装新的目的地址



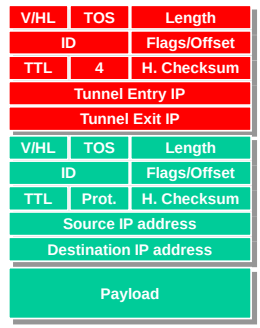
- 封装在IP分组中
  - 节点B向节点E发送一个分组
  - 数据字段封装另一个分组

应用

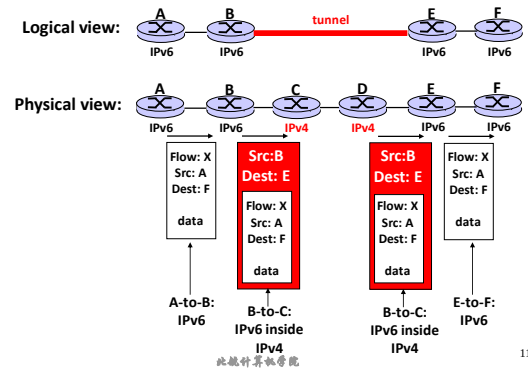
- VPN：虚拟专用网，Virtual private networks.
  - Connect subnets of a corporation using IP tunnels
  - Often combined with IPSec
- 支持新的协议
  - Routers that support the protocols use tunnels to “bypass” routers that do not support it
  - E.g. multicast
- 指定分组的路由
  - Routing is based on outer-header
  - E.g. mobile IP

IP-in-IP 隧道

- 协议：RFC 1993.
- IP 源和目的地址标识隧道的两个端点
- Protocol id = 4.
  - IP
- 字段复制：Several fields are copies of the inner-IP header.
  - TOS, some flags, ..
- TTL处理：Inner header is not modified, except for decrementing TTL.



6Bone: Deploying IPv6 over IP4



总结

- 虚电路和数据报的基本概念
- IP协议和IP地址
- 子网及前缀表示
- 最长前缀匹配
- 与IP相关的其他知识点
  - ICMP，ARP, DHCP，DNS，NAT，IPv6

作业要求

作业提交时间待定（课程中心提交）

完成小作业（1）

- 专题1 “网络体系结构”

1.任意选择1篇论文进行阅读

2.每人独立完成论文评论（paper review），评论内容要求：

- 作者主要观点和要解决的问题
- 研究方法评论（关键技术，优点和局限性）
- 论文的主要贡献
- 其他
- 注意：不是翻译，篇幅不限

3.作业提交（两个文档）

- .docx文件
- .pptx文件（约 10 页左右，请勿超过15页，课堂讨论用）

说明

小作业共五个专题：

1. 网络体系结构
2. SDN
3. 数据中心网络
4. 拥塞控制
5. 应用层网络与网络安全

作业提交说明：

- 论文可以提前阅读，但按专题要求提交作业。
- 每个专题有5~6篇论文，可在课程中心下载，每次作业任选1篇论文。
- 整个学期，每个同学至少选择3个专题完成小作业（鼓励多读论文）。

提交作业注意事项

1. 小作业命名格式：姓名+学号+论文文件名.docx (.pptx)
2. 小组提交的大作业文档命名格式：小组成员姓名+题目.docx（.pptx）
3. 请在截止期之前提交。提交成功后查看确认。
4. 如果逾期无法在网站提交，请在课前（周五之前）尽快发邮件至（liw@buaa.edu.cn）提交并说明理由，作业收到后会回复
5. 作业评分标准上传到课程中心

可用工具

- ◆网络模拟工具
  - ❖如 ns-2，ns-3，mininet
- ◆抓包工具：如 tcpdump，Wireshark
- ◆网络程序设计API
- ◆其他开源工具

大作业选题范围

◆课程项目选题范围：课程项目选题方向可以从论文阅读（小作业）的五个专题中选择，具体题目自拟：

1. 网络体系结构
2. SDN
3. 数据中心网络
4. 拥塞控制
5. 应用层网络与网络安全

大作业选题建议

- ◆ 阅读高质量学术论文并复现论文中的研究成果
  - ❖ Software Defined Networks (SDN), Network Function Virtualization (NFV), Quality of Experience (QoE), the Internet of Things (IoT), or the Cloud Computing
- ◆ SDN与网络虚拟化技术研究
  - ❖ 教程实现及应用开发
- ◆ 基于网络模拟平台的协议和算法研究
  - ❖ 模拟平台Mininet, ns2, or ns3..
  - ❖ 有线/无线路由协议实现
  - ❖ 拥塞控制等
  - ❖ 协议性能分析
- ◆ 网络测量和流量分析：网络性能参数/流量等数据的采集与分析
- ◆ 面向应用的传输协议实现与优化
- ◆ 其他

大作业过程管理

1. 分组：
   - 自由组合进行分组，每组2-3人（不超过3人）
   - 根据给定专题方向确定课程项目选题
2. 确定大作业选题（任选一种类型，必须包括网络程序设计部分）
   - 系统实现 design/implementation
   - 基于模拟平台的研究 measurement, analysis, and simulation
   - 网络测量和分析
3. 小组提交大作业开题报告（提交截止时间参见网站说明）
   - 包括相关背景分析，技术路线和实施方法等。成员分工和主要参考资料
   - 各个小组的成员针对项目需求，确定小组各个成员的分工，要求工作量相当。
4. 小组提交中期报告（抽查）
5. 大作业讨论（期末）
   - 各个小组制作演讲PPT（15页左右），约8分钟
   - 按小组进行演讲和讨论(Peer Review)。
6. 小组成员总结各自承担的工作，独立完成技术报告，期末考试前按时提交
   - 课程论文字数不限，按照期刊论文格式进行书写

大作业开题报告说明

- 注意作业的截止时间
- 作业内容

  以小组为单位提交大作业开题报告。 必须包括以下内容：
  - 问题提出和研究目标：为什么选择该题目？需要解决什么问题？
  - 研究现状与相关技术分析
  - 技术路线
  - 小组成员及其分工说明（必需）
  - 主要参考资料
- 要求：各个小组的成员针对项目需求，确定小组各个成员的分工，工作量相当。
- 提交作业（两个文档）
  - .docx文件
  - .pptx文件（约 10 页左右，请勿超过15页，课堂讨论用）