Ch4.Network Layer: The Data Plane

4.0 Background

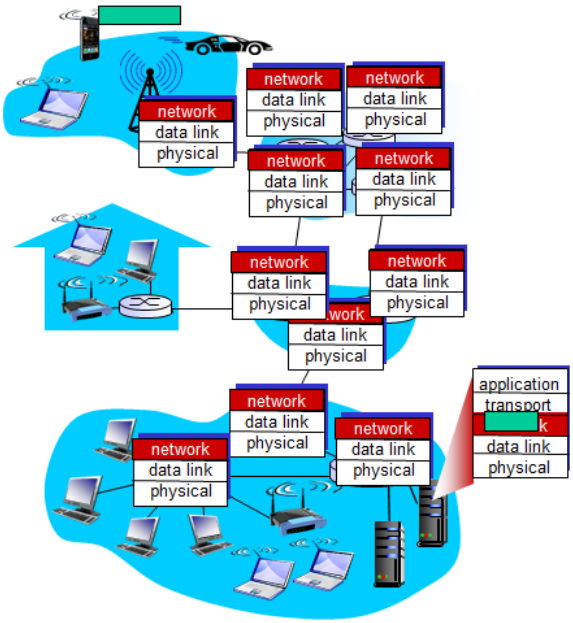
understand principles behind network layer services, focusing on data plane:

- network layer service models
- forwarding versus routing
- how a router works
- generalized forwarding

4.1 Overview of Network layer

提供服务

- 从发送端到接收端传输segment
- 在发送端将segment封装为datagram
- 在接收端，将segment发送到传输层
- 网络层协议在每个主机、路由器都有
- 路由器检查所有通过它的IP datagram的报头字段



Intetnet service model只提供best effort，不做任何保证

Network Architecture	Service Model	Bandwidth	Loss	Order	Timing	Congestion feedback
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

核心功能

- forwarding

- routing

网络层功能:

- *forwarding*:将datagram从路由器的输入移动到适当的路由器输出
- *routing*:确定数据包从源到目的地所采取的路由
  - *routing algorithms*

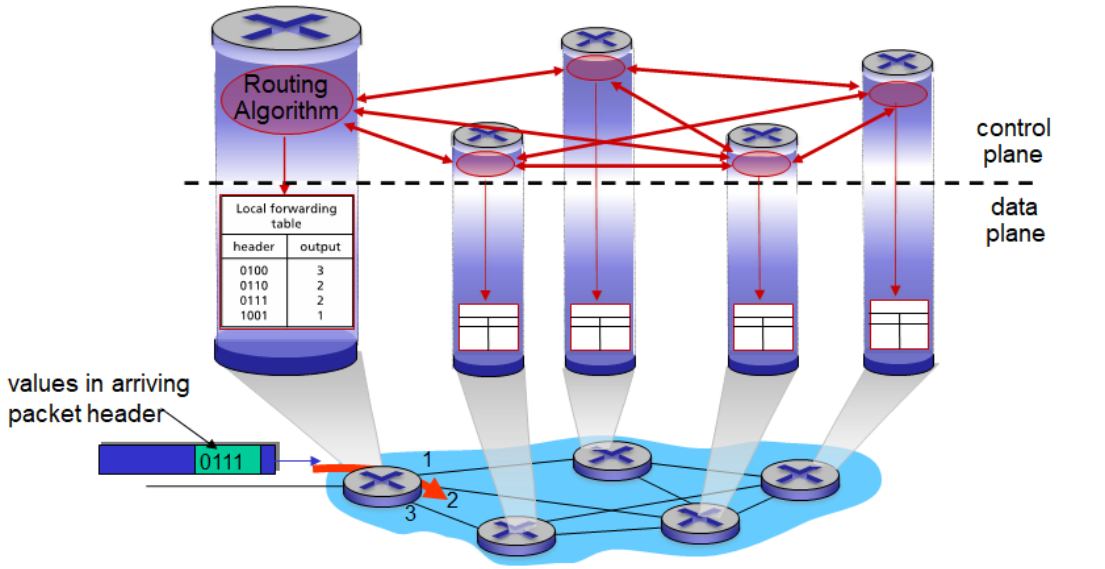
Data Plane 和 Control Plane

Control Plane负责计算forwarding table

Data Plane负责按照forwarding table上的内容去转发

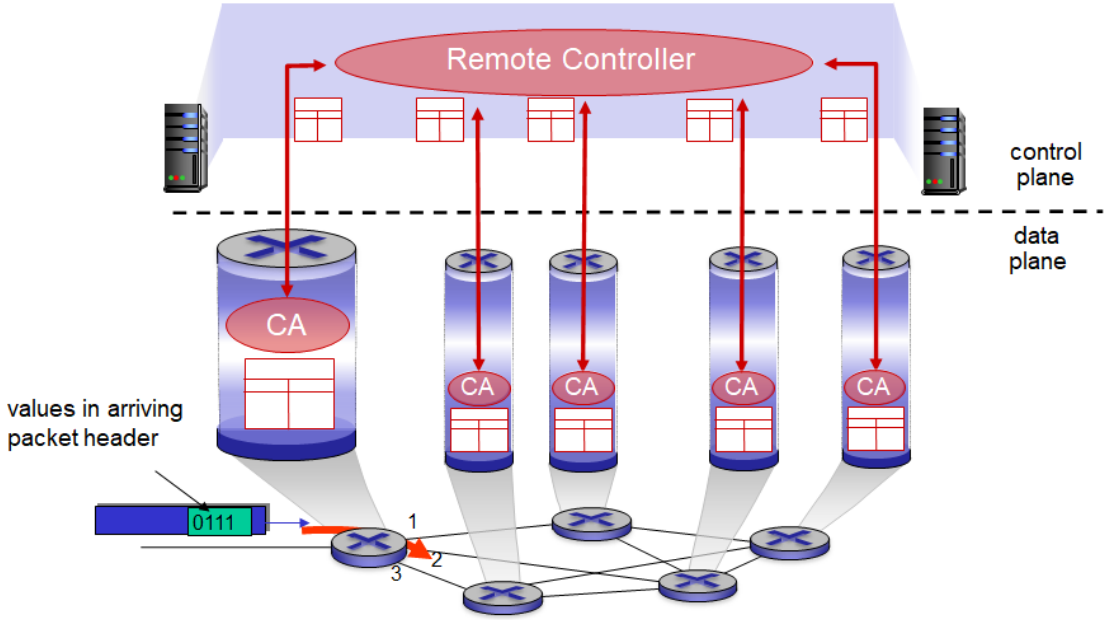
Per-router control plane

每个路由器中的单个路由算法组件在control plane上交互



SDN

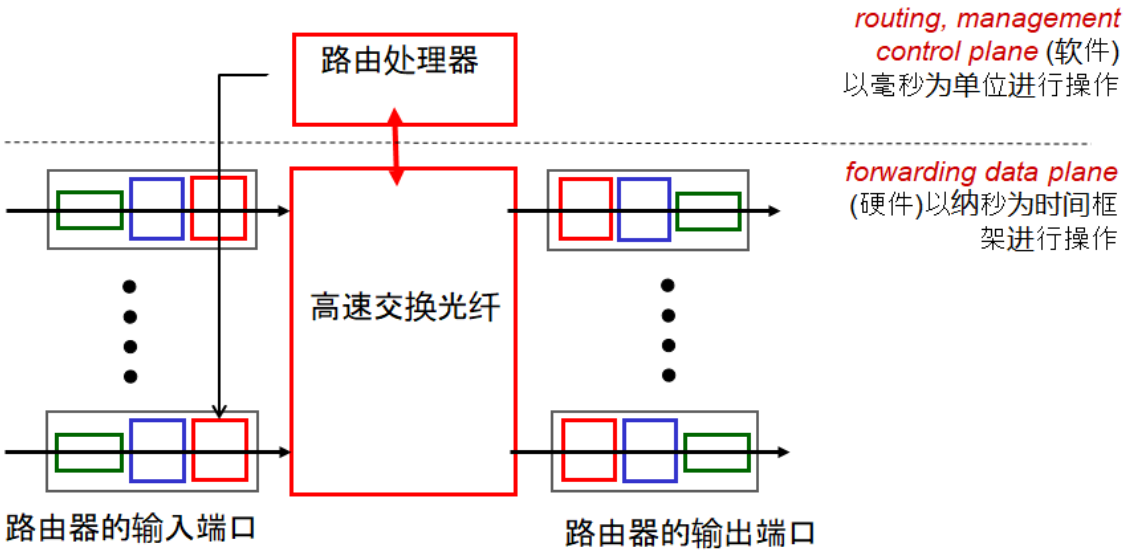
一个独立的(通常是远程的)控制器与本地控制代理(CAs)交互



4.2 What's inside a router

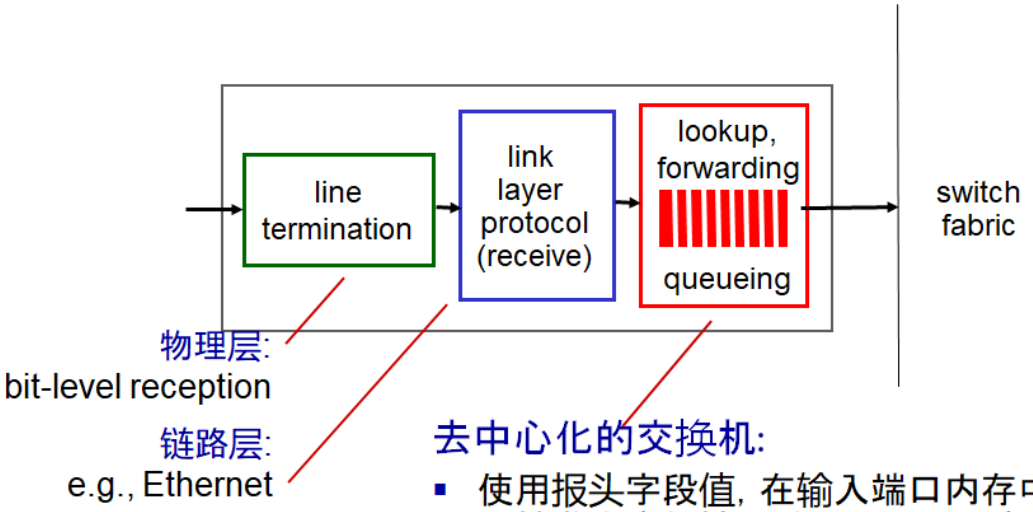
路由器架构

- 通用路由器架构的高级视图:



输入端口

介绍



去中心化的交换机:

- 使用报头字段值，在输入端口内存中使用转发表查找输出端口(“匹配加动作”)
- 目标:以' line speed '完成输入端口处理
- 排队:如果数据报到达交换机结构的速度快于转发速率
- *基于目的地址的forwarding:*仅基于目的IP地址转发(传统)
- *广义forwarding:* 基于任何一组报头字段值转发

forwarding table转发表与最长前缀匹配

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

longest prefix matching

当查找给定目的地址的转发表项时，使用与目的地址匹配的最长地址前缀

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

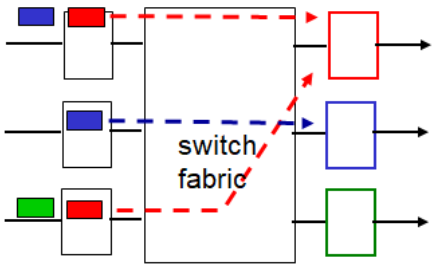
示例:

DA: 11001000 00010111 00010110 10100001 interface 1

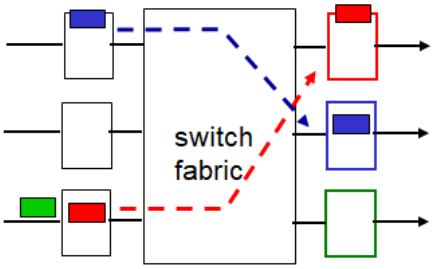
DA: 11001000 00010111 00011000 10101010 interface 3

排队，HOL阻塞

- fabric传输比输入端口加起来慢→输入队列可能发生>排队
  - 由于输入缓冲区溢出可能导致的排队延迟和丢失!
- Head-of-the-Line (HOL)阻塞:队列前面的排队datagram阻止其他队列中的数据报前进



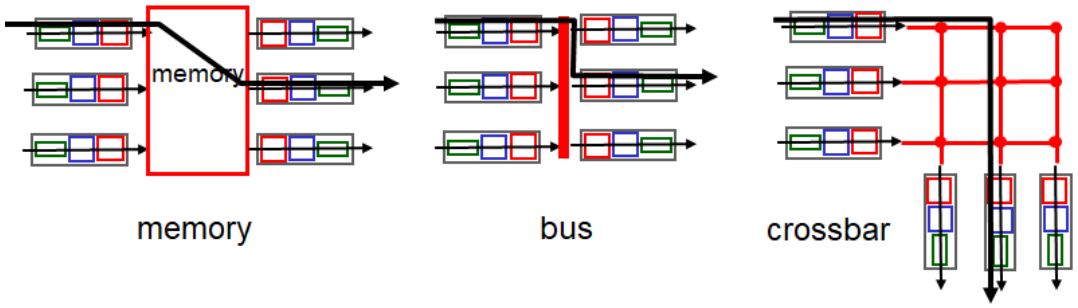
输出端口争用:只能传输一个红色数据报
下方的红色包被阻塞



经过一个包后:绿色包经历HOL阻塞

交换网络switching fabrics

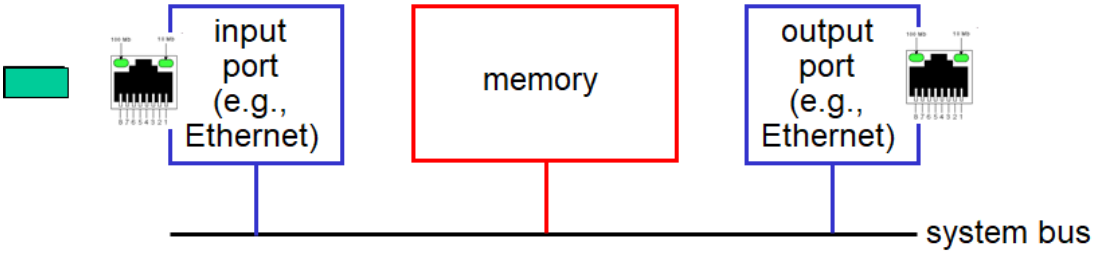
- 将数据包从输入缓冲区传送到适当的输出缓冲区
- 交换速率switching rate:数据包从输入传输到输出的速率
  - 通常测量为输入/输出线速度的倍数
  - N输入：交换速率N乘以倍行率
- 三种类型的switching fabrics



Memory

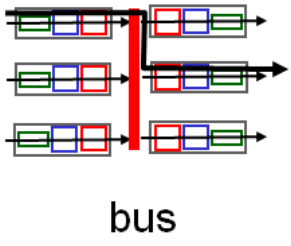
第一代路由器:

- 传统计算机在中央处理器的直接控制下进行切换
- 包被复制到系统内存中
- 速度受内存带宽限制(每个数据报2个总线交叉点)



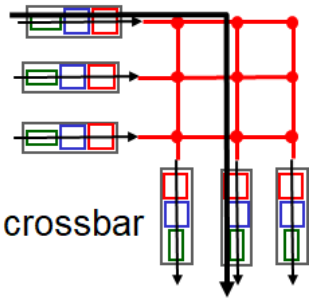
Bus

- 来自输入端口存储器的数据报通过共享总线输出端口存储器
- *bus contention:* 交换速度受bus带宽限制
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers



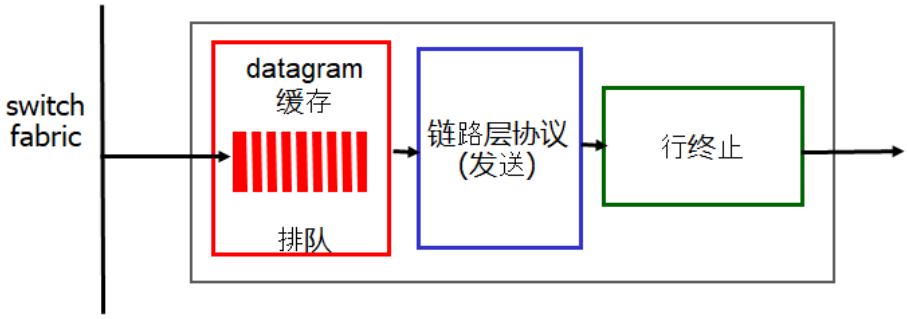
Cross Bar

- 克服bus带宽限制
- banyan networks, crossbar,其他互连网最初是为连接多处理器中的处理器而开发的
- advanced design:将datagram分割成固定长度的单元，通过fabric转发单元.
- Cisco 12000: switches 60 Gbps through the interconnection network



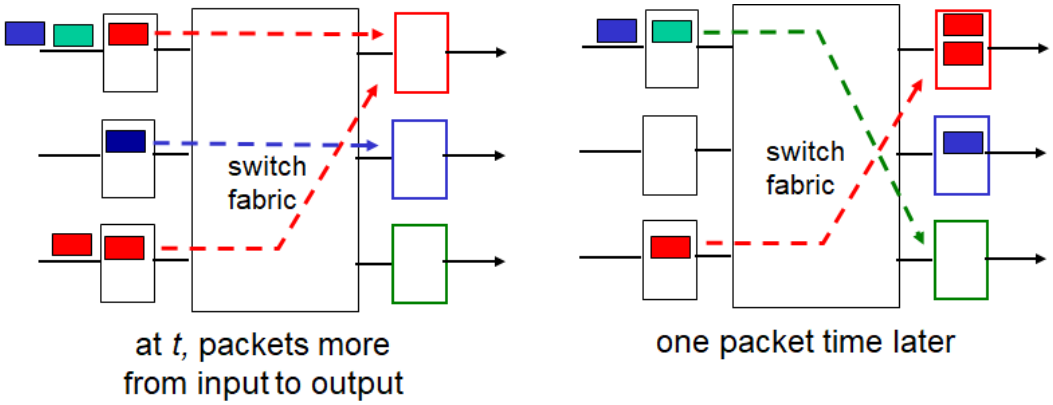
输出端口

介绍



- *缓存buffering*当数据报从fabric到达比传输速率快时(数据报(包)可能由于拥塞、缺乏缓冲区而丢失)
- *调度规则scheduling discipline*在等待传输的数据报中进行选择(优先级调度-谁获得最佳性能，网络中立性)

排队



- 当通过switch的到达率超过输出的行速度时，缓存
- *由于输出端口缓冲区溢出而导致的排队(延迟)和丢失!*

Buffer大小

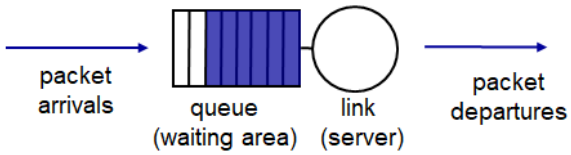
- 如果有N个流，RTT (比如说 250 msec)，link容量为C，则

$$\text{Buffer} = \frac{RTT \cdot C}{\sqrt{N}}$$

调度机制

丢掉谁?

- *scheduling:* 在缓存中选择下一个要在链路上发送的包
- *FIFO (first in first out) scheduling:* 按到达顺序发送到队列
  - real-world example?
  - *discard policy:* 如果数据包到达一个满了的队列:丢弃谁?
    - *tail drop:* 丢掉到达的包
    - *priority:* 按优先级删除/删除
    - *random:* 随机丢掉



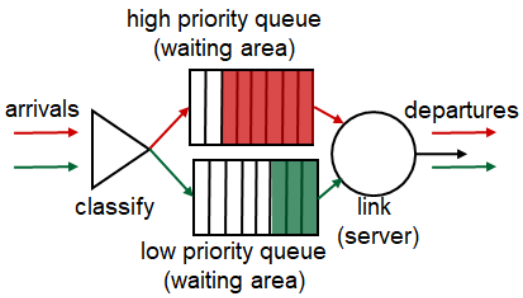
发送谁?

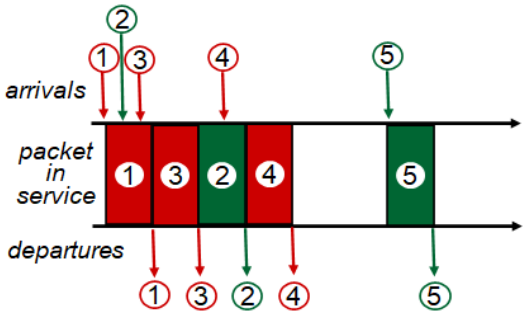
高优先级的队列必须全部发完后才发低优先级的队列

priority scheduling:

发送最高优先级的队列数据包

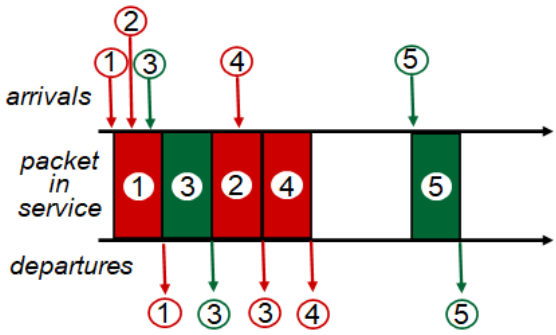
- 多个类，具有不同的优先级
  - 类可能取决于标记或其他头信息，如IP源/目的地，端口号等





Round Robin (RR) scheduling:

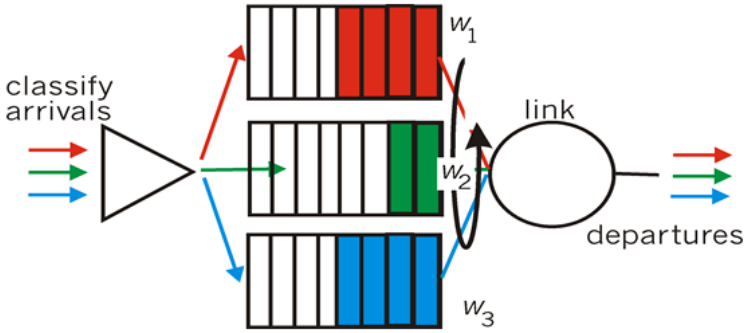
- 多种类型的
- 循环扫描类队列，从每个类发送一个完整的包(如果有的话)



加权发送

Weighted Fair Queuing (WFQ):

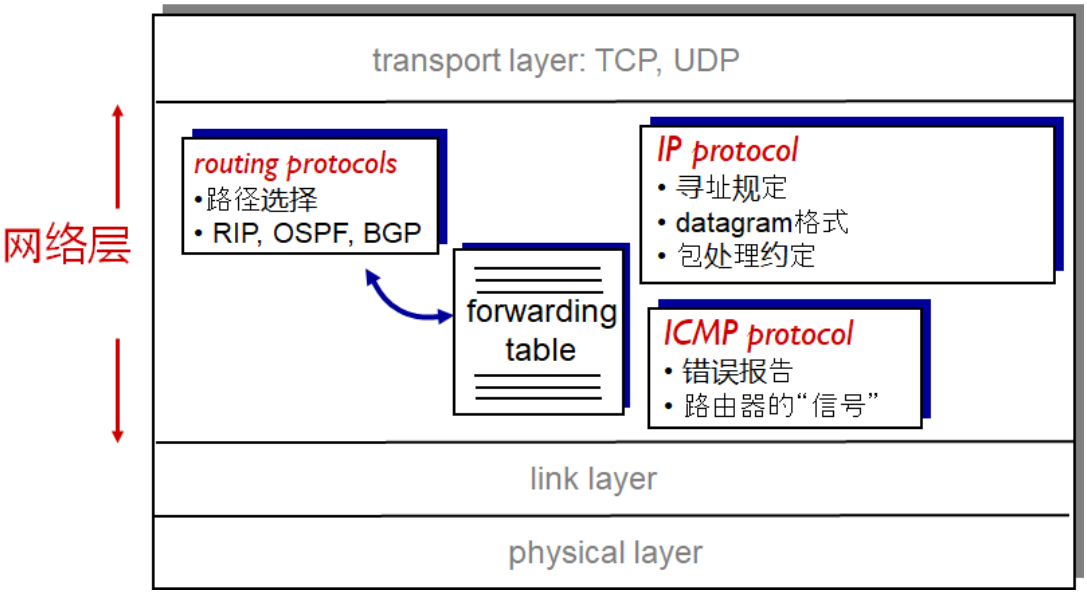
- 广义的Round Robin
- 每个类在每个周期中得到加权的服务量



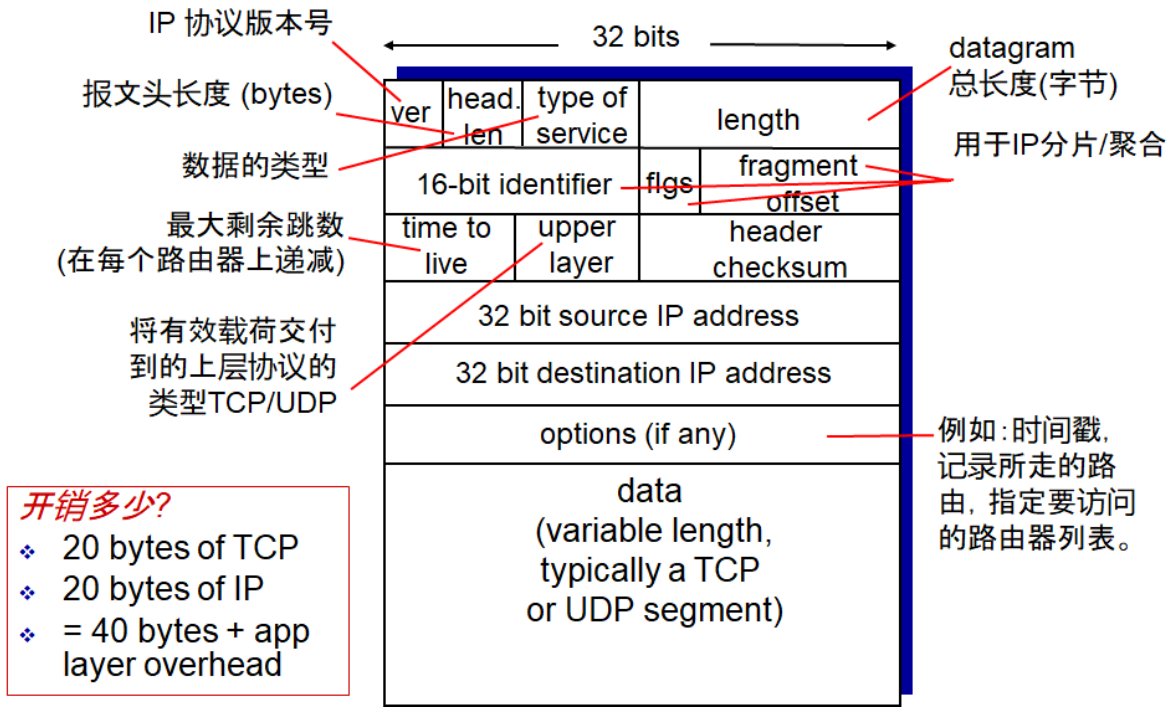
4.3 IP: Internet Protocol

因特网网络层常见协议

主机、路由器网络层功能:



IP报文格式

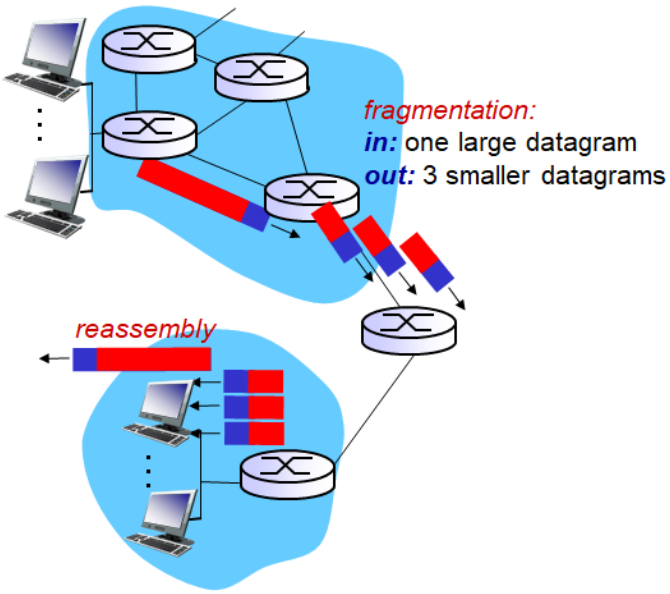


开销多少?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

IP fragmentation, reassembly

- 网络链路的MTU (max.传输大小)-可能的最大链路层frame容量
  - 不同的链路类型，不同的MTU
- 在网络中划分的大型IP datagram("分片")
  - 一个datagram变成多个datagram
  - 只在最终目的地"重新组装"
  - IP报头位有用来识别、排序相关的片段

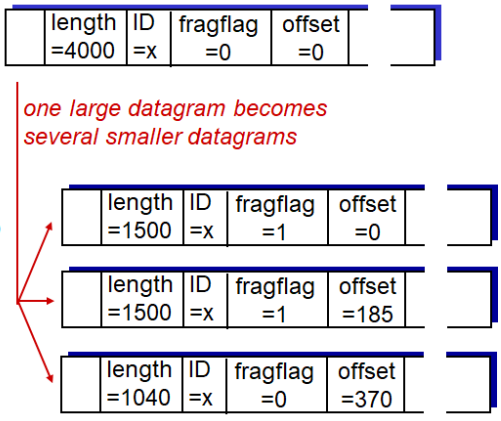


IP分片示例

示例:

- 4000 byte(total datagram)
- 4000 byte – 20 byte(header) = 3980 byte(payload)
- 3980 / 1500 = 3 (fragment) (向上取整)

- MTU = 1500 bytes
- 1500 byte – 20 byte(header) = 1480 byte(fragment payload)
- offset:
- fragment1 = 0
- fragment2 = 1480 byte / 8 = 185
- fragment3 = 2960 byte / 8 = 370

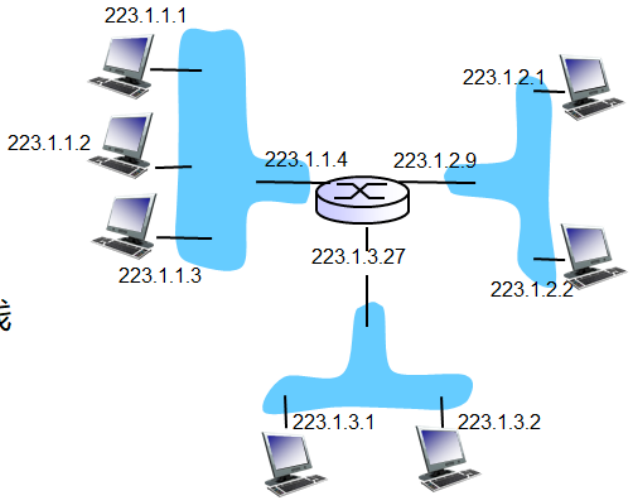


4.4 IPv4 32bit

IP地址介绍

写IP地址的时候，将每8bit转换成10进制数，中间用点隔开

- *IP address:* 32-bit 主机、路由器接口的标识符
- *interface:* 主机/路由器与物理链路之间的连接
  - 路由器通常有多个接口
  - 主机通常有一个或两个接口(例如，有线以太网，无线802.11)
- *IP地址与每个接口关联*

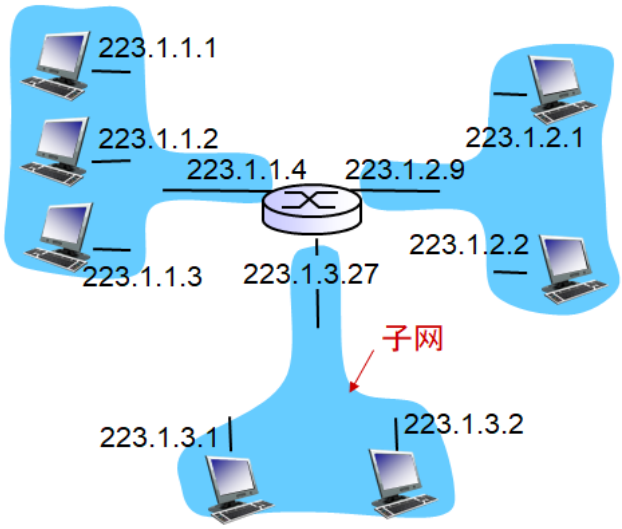


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

子网Subnet

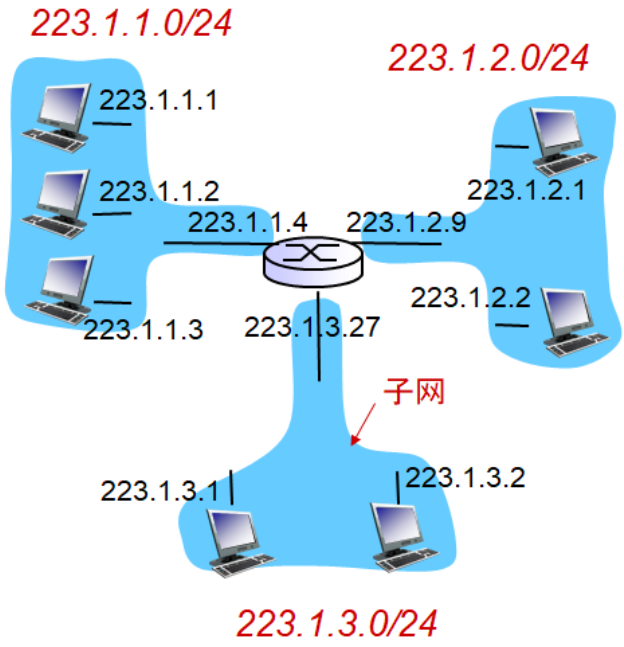
- IP address:
  - 子网部分-高bit位
  - 主机部分-低bit位
- *什么是子网?*
  - IP地址子网部分相同的设备接口
  - 可以在没有路由器的情况下能物理地到达对方



由3个子网组成的网络

如何判断子网:

- 为了确定子网，将每个接口与它的主机或路由器分离，创建孤立的网络孤岛
- 每个隔离的网络称为子网

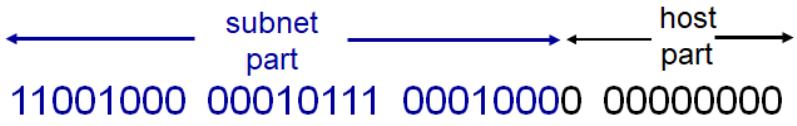


子网掩码subnet mask: /24

CIDR划分子网

CIDR: Classless Inter Domain Routing

- 任意长度的地址的子网部分
- 地址格式:a.b.c.d/x，其中x是地址子网部分的#比特位



200.23.16.0/23

如何获取IP地址

Q: 主机如何获得IP地址?

- 系统管理员在文件中硬编码
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol: 从服务器动态获取地址
  - “即插即用的”

DHCP (Dynamic Host Configuration Protocol)

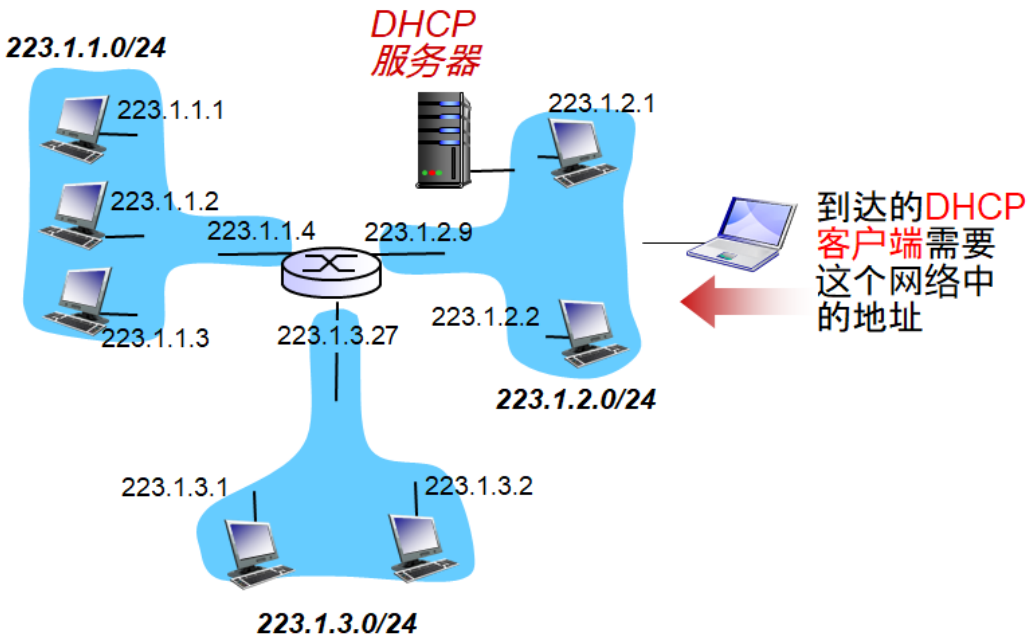
介绍

目的: 允许主机在加入网络时动态地从网络服务器获取IP地址

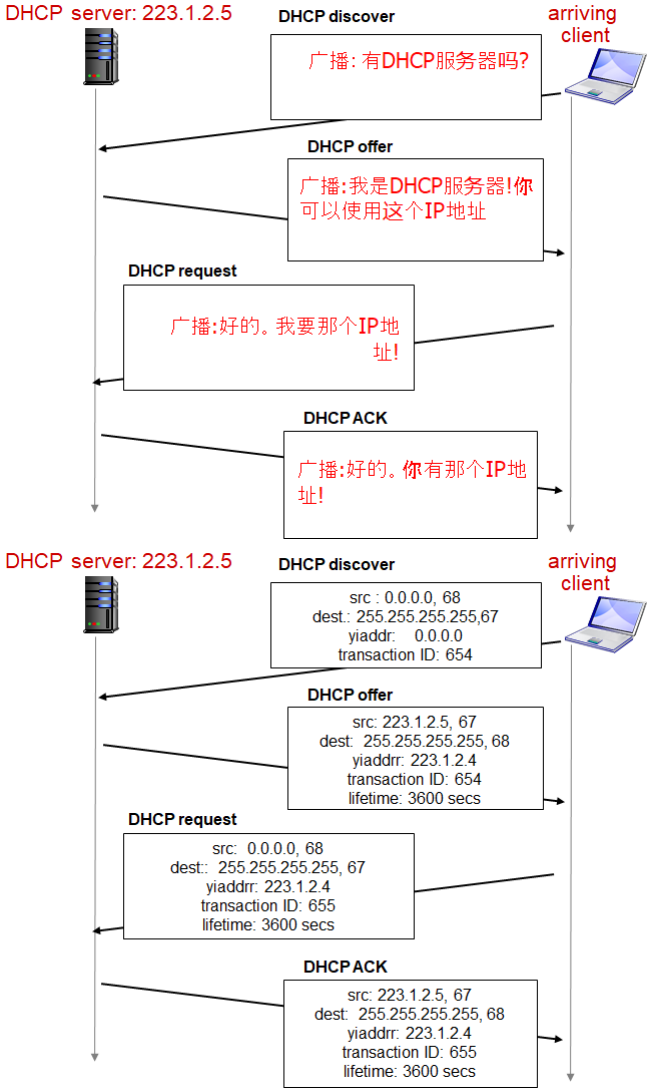
- 按使用地址续租
- 允许重用地址(只有在连接/"on"时保持地址)
- 支持想要加入网络的移动用户

DHCP 概要:

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg



DHCP地址分配原则



DHCP其他功能

DHCP不仅可以在子网中分配IP地址:

- 给出客户端的第一跳路由器地址
- 给出DNS服务器的名称和IP地址
- 给出网络掩码(指示网络与地址的主机部分)

如何获得子网的分配

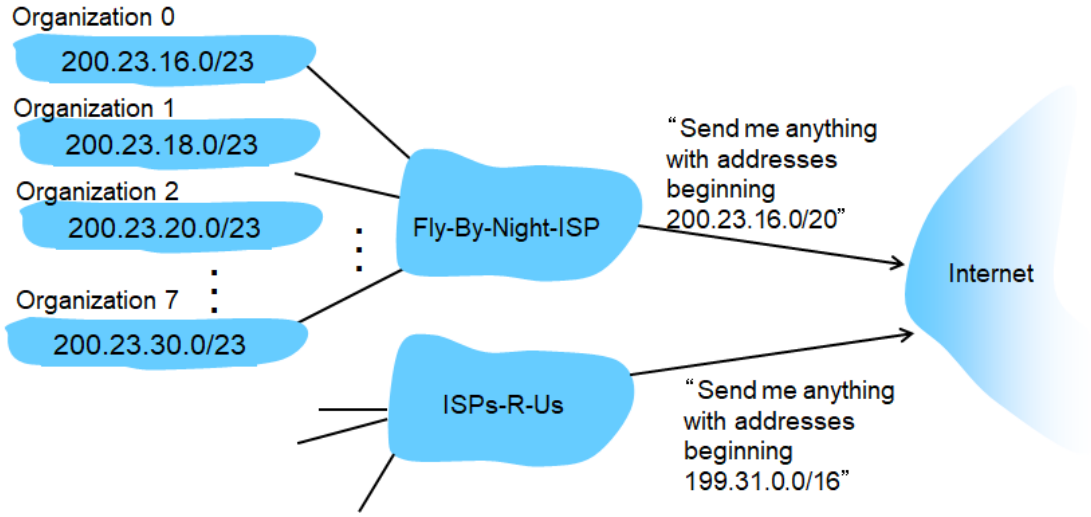
Q:网络如何得到IP地址的子网部分?

A: 获取其提供者的ISP地址空间的一部分

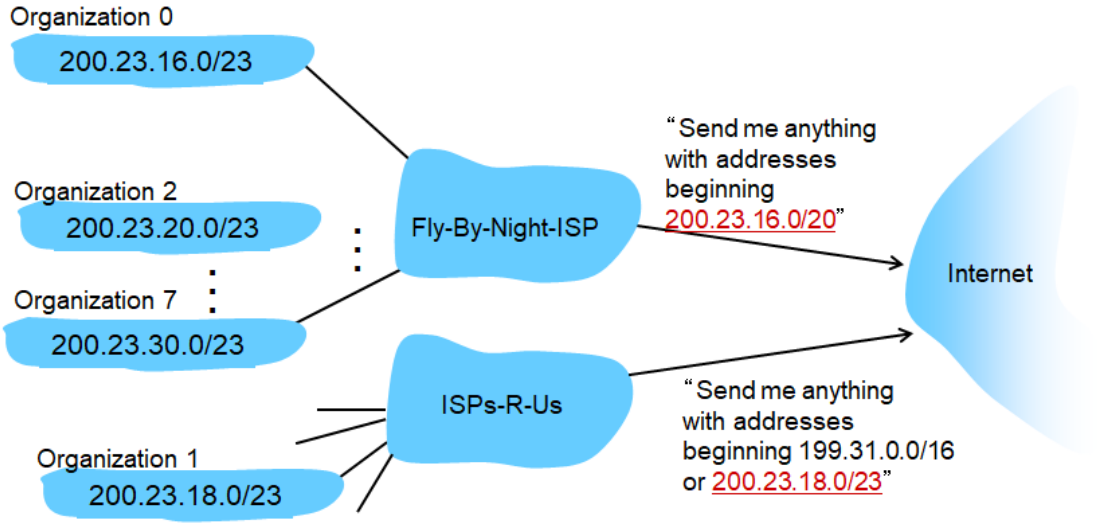
ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
…	……	….	….
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

路径聚合route aggregation

分层寻址允许有效地发布路由信息:



ISPs-R-Us 有一条到Organization 1更具体的路线



NAT: network address translation

介绍

*motivation:*就外界而言，局部网络只使用一个IP地址:

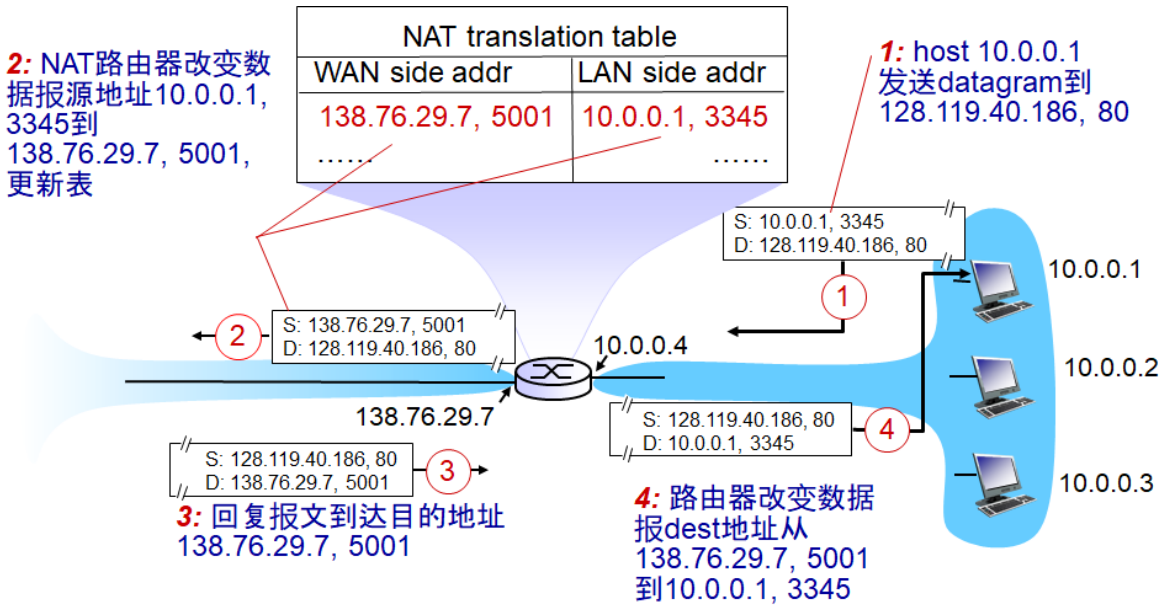
- 不需要ISP提供的地址范围:所有设备只需要一个IP地址
- 可以在本地网络不通知外部世界的情况下改变设备地址
- 可以在不改变本地网络设备地址的情况下改变ISP
- 局部网络内的设备不能被显式寻址，但可以被外部世界看到(安全因素)

功能

implementation: NAT 路由器必须做到:

- *outgoing datagrams: replace* 每一个来自(source IP address, port #)的datagram转换成 (NAT IP address, new port #)
  . . . 远端客户端/服务器将响应使用(NAT IP address, new port #)作为它的目的地址
- *remember (in NAT translation table)* 每一个(source IP address, port #) 到(NAT IP address, new port #) 的键值对
- *incoming datagrams: replace* (NAT IP address, new port #) 每个传入数据报的目的地字段与相应的(source IP address, port #) 存储在NAT表里

示例



争议性

- 16-bit port-number field:
  - 用一个局域网地址同时连接60000个!
- NAT是有争议的:
  - 路由器应该只处理到第三层，因为用到了DHCP上升到了第五层应用层
  - IPv6可以解决地址短缺的问题
  - 违反了端到端参数
    - 应用设计者必须考虑NAT的可能性，例如P2P应用
  - NAT穿越:如果客户端想要连接到NAT后的服务器怎么办?

4.5 IPv6 128bit

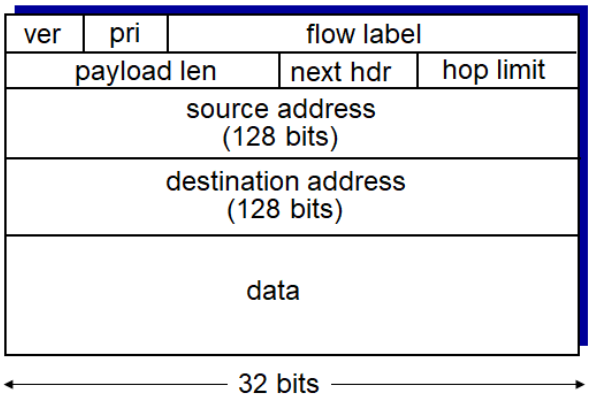
IPv6地址介绍

- *原始动机*: 32位IPv4地址空间已经完全分配
- 其它动机:
  - 报头格式有助于加速处理/转发
  - 报文头更改以促进QoS

IPv6 datagram格式:
- 固定长度的40byte报文头
- 不允许分片

IPv6报文格式

priority:确定流中datagram的优先级
flow Label: 识别相同“流”中的data(“流”的概念没有很好的定义)
next header: 识别上层协议(TCP/UDP)

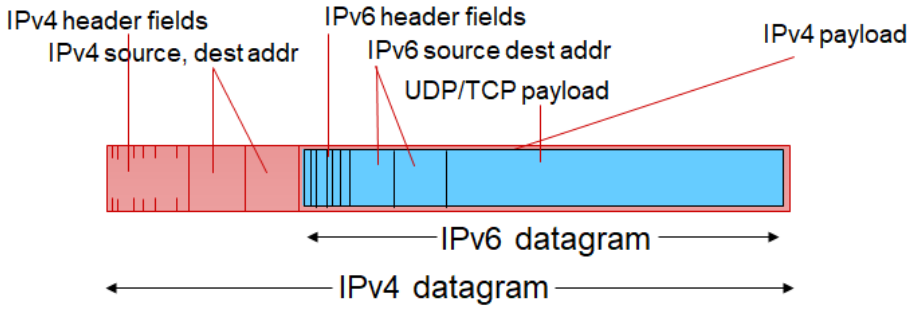


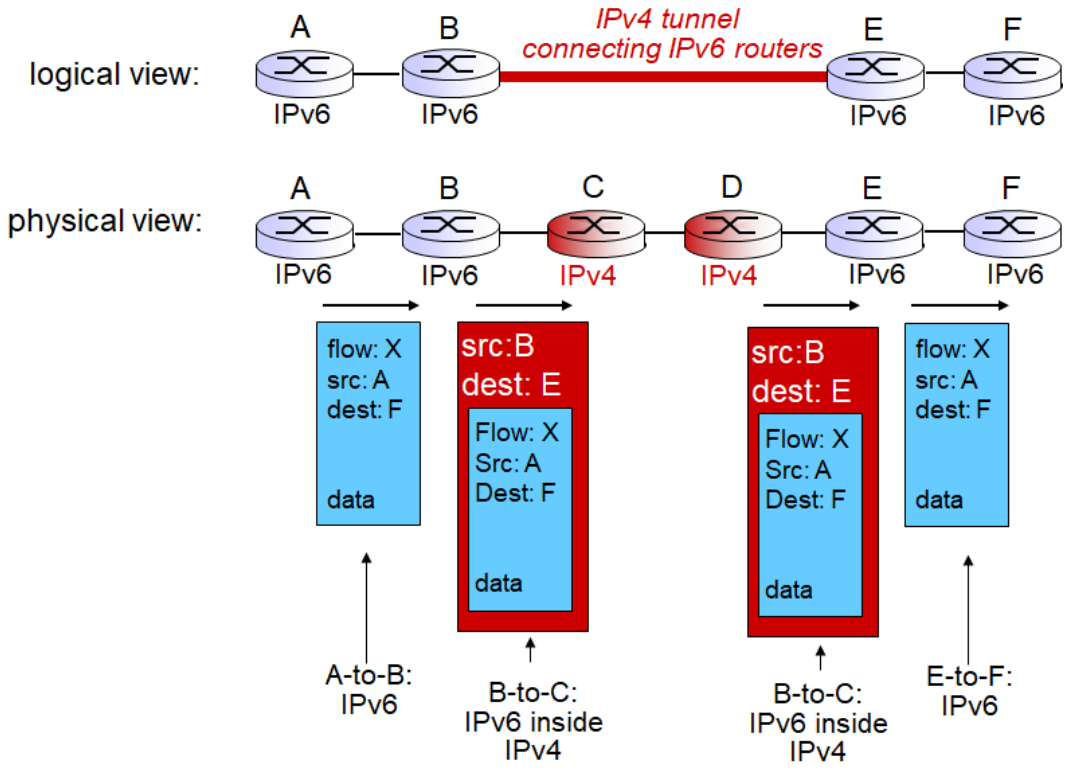
与IPv4的不同

- *checksum*:完全删除，以减少每一跳的处理时间
- *options*: 允许，但在header之外，由" Next header "字段指示
- *ICMPv6*: ICMP的新类型
  - 附加消息类型，例如“包太大”
  - 组播组管理功能

IPv4与IPv6共存（Tunneling）

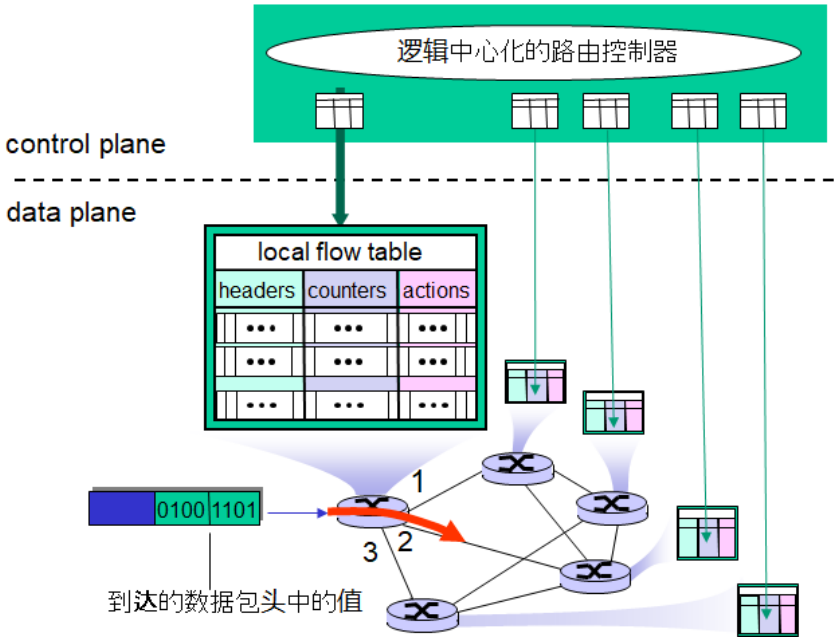
- 并不是所有的路由器都能同时升级
  - 混合IPv4和IPv6路由器的网络将如何运作?
- *tunneling*: IPv6 datagram作为payload在IPv4 datagram中在IPv4路由器之间携带





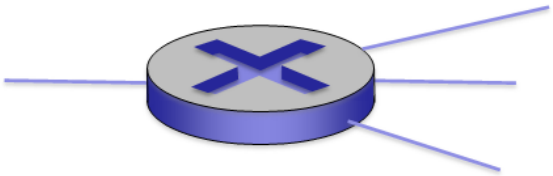
4.6 Generalized Forward and SDN

每个路由器包含一个由逻辑集中的路由控制器计算和分发的flow table



Openflow data plane概述

- *flow*:由报头字段定义
- 广义转发:简单的包处理规则
  - *Pattern:* 匹配报头字段中的值
  - *Actions: for matched packet:* 丢弃、转发、修改、匹配的数据包或将匹配的数据包发送到控制器
  - *Priority*: 消除歧义重叠的pattern
  - *Counters:* #bytes and #packets

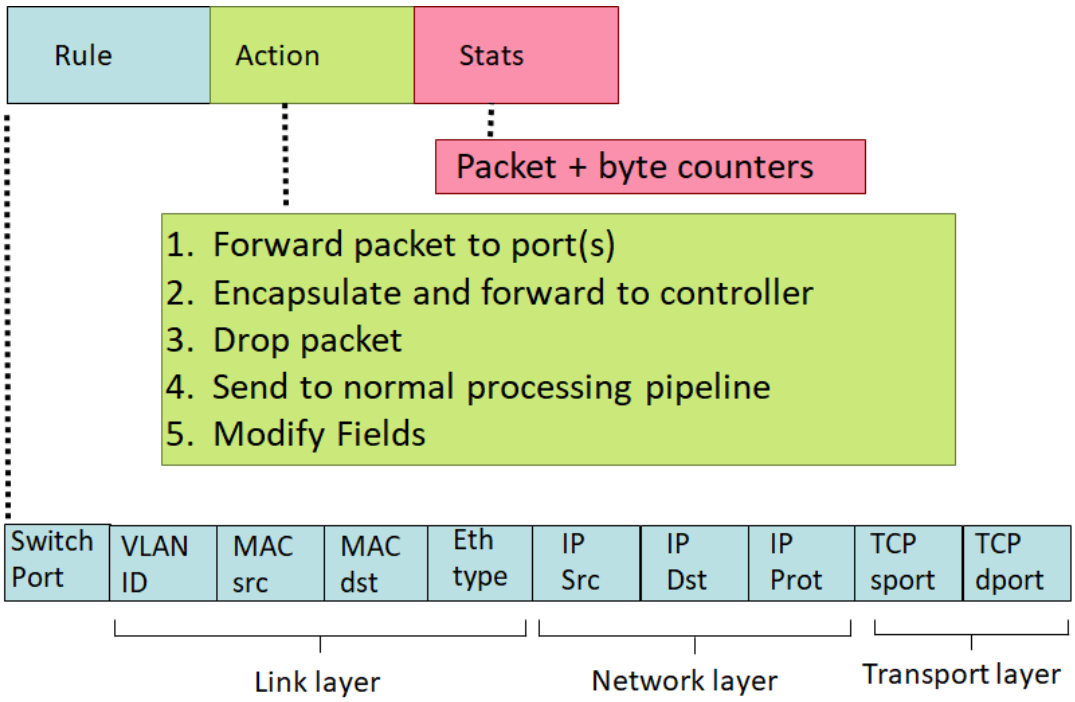


路由器中的flow table(由控制器计算和分发)定义了路由器的匹配+动作规则

* : wildcard

1. src=1.2.*.*, dest=3.4.5.* → 丢弃
2. src = *.*.*.*, dest=3.4.*.* → 转发(2)
3. src=10.1.2.3, dest=*.*.*.* → 发送给控制端

Flow table 表项



示例

基于目的地的转发:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP datagrams destined to IP address 51.6.0.8 should be forwarded to router output port 6

防火墙:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	*	*	*	*	22	drop

do not forward (block) all datagrams destined to TCP port 22

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Forward
*	*	*	*	*	128.119.1.1	*	*	*	*	drop

do not forward (block) all datagrams sent by host 128.119.1.1

理想愿景

- *match+action:* 统一不同的设备
- 路由器
  - *match:* 最长目的IP前缀
  - *action:* 转发链路
- 交换机
  - *match:* 目的地MAC地址
  - *action:* 转发或者广播
- 防火墙
  - *match*: IP地址和TCP/UDP端口号
  - *action:* 允许或拒绝
- NAT
  - *match:* IP地址和端口号
  - *action:* 重写地址和端口