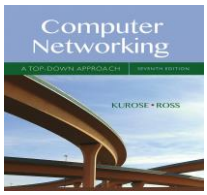
Chapter 4
网络层：数据平面
Network Layer: The Data Plane



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

Chapter 4: 网络层

章节目标:

- 理解网络层服务背后的原理，关注数据平面:
  - 网络层服务模型
  - 转发与路由
  - 路由器工作原理
  - 通用转发
- 实例化，在网络中的实现

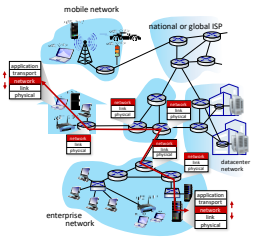
网络层：“数据平面”路线图

- 网络层：概述
  - 数据平面data plane
  - 控制平面(control plane)
- 路由器内部工作原理
  - 输入端口处理、交换、输出端口处理
  - 缓冲区管理、调度
- 网络协议IP(Internet Protocol)
  - 数据报格式
  - 编址
  - 网络地址转换
  - IPv6

- 通用转发和SDN
  - 匹配和动作
  - OpenFlow：匹配加动作
- 中间盒子(Middleboxes)

网络层服务和协议

- 负责从发送主机到接收主机的报文段传输
  - 发送方:将报文段封装为数据报，然后传递到链接层
  - 接收方:将报文段传送到传输层协议
- 每个网络设备均有网络层协议:主机、路由器
- 路由器:
  - 检查通过它的所有IP数据报中的头部字段
  - 将数据报从输入端口移动到输出端口，以端到端路径传输数据报



两种重要的网络层功能

- 转发(forwarding):将数据包从路由器的输入链路移动到适当的路由器输出链路。
- 路由选择(routing):确定数据包从源到目的地所采用的路由或路径
  - 路由选择算法(routing algorithm)

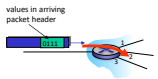
类比: 旅行

- 转发: 通过单个立交桥的过程
- 路由选择: 规划旅途从出发地到目的地行程的过程

网络层：数据平面和控制平面

数据平面(Data plane):

- 本地(local), 每个路由器功能
- 确定到达路由器输入端口的数据报如何转发到路由器输出端口

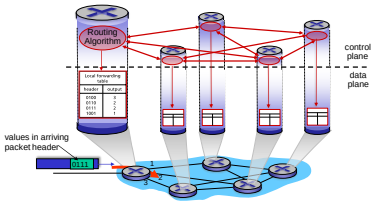


控制平面(Control plane):

- 全网(network-wide) 逻辑
- 确定数据报如何沿从源主机到目标主机的端到端路径，确定路由器之间路由
- 两种控制平面方法:
  - 传统路由算法: 在路由器中实现
  - 软件定义网络 (SDN):在（远程）服务器中实现

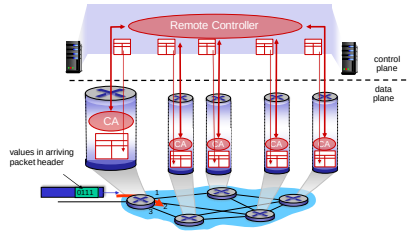
每个路由器的控制平面

每个路由器中的各个路由算法组件都在控制平面中进行交互



软件定义网络(SDN) 控制平面

远程控制器计算和分发转发表以供每台路由器所使用。



网络层服务模型

问：“通道”将数据报从发送方传输到接收方的服务模型是什么？

单个数据报的示例服务：

- 确保交付
- 具有时延上界（小于40 msec）的确保交付

数据报流的示例服务：

- 有序分组交付
- 确保最小流量带宽
- 数据包间距变化的限制

网络层服务模型

网络架构	服务模型	服务质量（QoS）保证？			
		带宽	无损	有序	实时
因特网	尽力而为	无	无	无	无

因特网“尽力而为（best effort）”服务模型无法保证：

i. 成功将数据报传送到目的地
ii. 交付的时延和顺序
iii. 端到端流量的可用带宽

Network-layer service model

网络架构	服务模型	服务质量 (QoS) 保证？			
		带宽	无损	有序	实时
因特网	尽力而为	无	无	无	无
ATM	恒定比特率(CBR)	恒定速率	有	有	有
ATM	可用比特率(ABR)	保证最小值	无	有	无
因特网	综合服务Intserv (RFC 1633)	有	有	有	有
因特网	区分服务Diffserv (RFC 2475)	可能	可能	可能	无

关于尽力而为服务的思考：

- 简单的机制使互联网得以广泛部署和采用
- 充足的带宽供应允许实时应用程序（例如，交互式语音、视频）的性能在“大部分时间”内“足够好”
- 重复的、应用层的分布式服务（数据中心，内容分发网络）连接到客户端的网络附近，从而允许从多个位置提供服务
- “弹性”服务的拥塞控制有一定作用

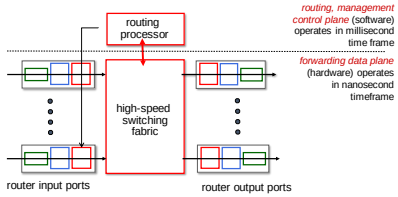
尽力而为服务模式的成功是有目共睹的

网络层：“数据平面”路线图

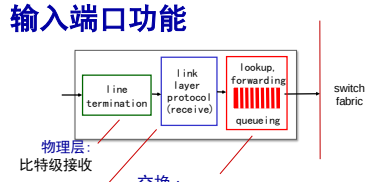
- 网络层：概述
  - 数据平面data plane
  - 控制平面(control plane)
- 路由器内部工作原理
  - 输入端口处理、交换、输出端口处理
  - 缓冲区管理、调度
- 网络协议IP(Internet Protocol)
  - 数据报格式
  - 编址
  - 网络地址转换
  - IPv6
- 通用转发和SDN
  - 匹配和动作
  - OpenFlow：匹配加动作
- 中间盒子(Middleboxes)

路由器架构概述

通用路由器体系结构的总体视图：



输入端口功能

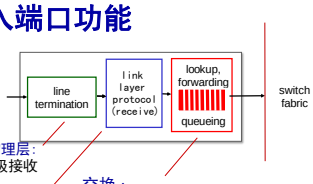


物理层：
比特级接收

链路层：
例如，以太网
（第6章）

交换：
- 使用头字段值，使用输入端口内存中的转发表查找输出端口（“*匹配加操作*”）
- 目标：以“线速”完成输入端口的处理，纳秒级执行
- 输入端口排队：如果数据报到达的速度快于进入交换结构的转发速度

输入端口功能



物理层：
比特级接收

链路层：
例如，以太网
（第6章）

交换：
- 使用头字段值，使用输入端口内存中的转发表查找输出端口（“*匹配加操作*”）
- 基于目的地转发：仅基于目标IP地址（传统）进行转发
- 通用转发：基于任何一组头字段值进行转发

基于目的地转发

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010000 00000100	0
11001000 00010111 00010000 00000101 through 11001000 00010111 00010000 00000111	3
11001000 00010111 00011000 11111111	
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

问：但如果范围划分得不太好会怎么样？

最长前缀匹配

最长前缀匹配(longest prefix match)

在查找给定目标地址的转发表条目时，使用与目标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

examples:

11001000 00010111 00010110 10100001 which interface?

11001000 00010111 00011000 10101010 which interface?

最长前缀匹配

最长前缀匹配(longest prefix match)

在查找给定目标地址的转发表条目时，使用与目标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

match!

examples:

11001000 00010111 00010110 10100001	which interface?
11001000 00010111 00011000 10101010	which interface?

最长前缀匹配

最长前缀匹配(longest prefix match)

在查找给定目标地址的转发表条目时，使用与目标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

match!

examples:

11001000 00010111 00010110 10100001	which interface?
11001000 00010111 00011000 10101010	which interface?

最长前缀匹配

最长前缀匹配(longest prefix match)

在查找给定目标地址的转发表条目时，使用与目标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

match!

examples:

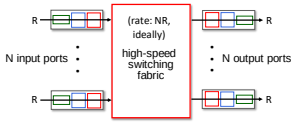
11001000 00010111 00010110 10100001	which interface?
11001000 00010111 00011000 10101010	which interface?

最长前缀匹配

- 为什么使用最长前缀匹配
- 最长前缀匹配：通常使用三态内容可寻址存储器（Tenary Content Address Memory, TCAM）来查找
  - 内容可寻址：向TCAM寻找地址：在一个时钟周期内检索地址，与表大小无关
  - Cisco Catalyst: 能够保存100多万TCAM转发表项

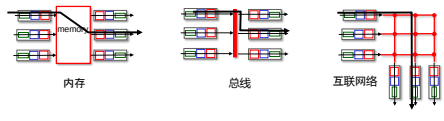
交换结构

- 将数据包从输入链路传输到适当的输出链路
- 交换速率：数据包从输入传输到输出的速率
  - 通常测量为输入/输出线路速率的倍数
  - N个输入：交换速率需要是线路速率的N倍



交换结构

- 将数据包从输入链路传输到适当的输出链路
- 交换速率：数据包可以从输入传输到输出的速率
  - 通常测量为输入/输出线路速率的倍数
  - N个输入：交换速率需要是线路速率的N倍
- 交换结构的三种主要类型：



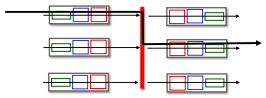
经内存交换

第一代路由器：

- 传统的计算机在CPU的直接控制下进行交换
- 分组被复制到处理器内存中
- 速度受到内存带宽的限制（每个分组需2次穿越系统总线）

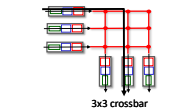
经总线交换

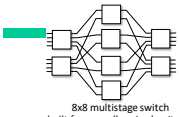
- 分组通过一条共享的总线从输入端口的内存传递到输出端口的内存
- 总线竞争：交换速率受限于总线的带宽
- 32 Gbps总线，Cisco 5600：对于接入路由器该速度足够了



经互联网络交换

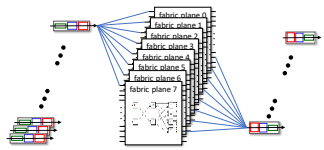
- Crossbar与Clos networks，在发展初期是用来连接多处理器系统中的处理器的
- 多级交换：多级小交换的nxn交换
- 利用并行性：
  - 在进入时将数据报分段为固定长度的单元格
  - 通过结构交换单元，在出口重新组装数据报





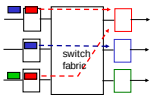
经互联网络交换

- 通过并行使用多个切换“平面”进行缩放：
  - 加速，并行缩放
- Cisco CRS路由器：
  - 基本单位：8个交换平面
  - 每个平面：3级内联网络
  - 高达100 Tbps的交换容量

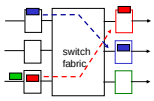


输入端口排队

- 交换网络的处理速度低于所有输入端口之和 -> 导致分组在输入端口的队列中排队
  - 由于输入缓冲区溢出导致的排队延迟和数据丢失！
- 线路前部/首部(Head-of-the-Line, HOL) 阻塞：在队列的排头上的分组挡住了其他分组的前移

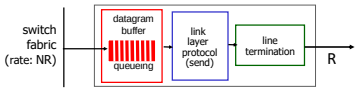


output port contention: only one red datagram can be transferred. lower red packet is *blocked*



one packet time later: green packet experiences HOL blocking

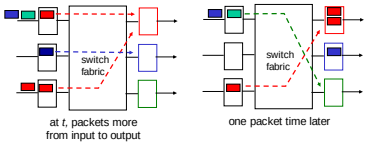
输出端口排队



This is a really important slide

- 缓存(Buffering)：当来自交换网络的分组到达速度高于传输速率时，需要进行缓存。
- 丢弃策略(Drop policy)：如果没有空闲的缓存，要丢弃哪些分组？ → 分组可能会由于拥塞、缺少缓存而丢失
- 调度原则(Scheduling discipline)：从队列中的分组中选择传输 → 优先级调度—谁获得最佳性能

输出端口排队



- 当交换速度*超过*输出线路的速率时，需要进行缓存
- 输出端口的溢出会造成排队（延迟）和数据丢失！

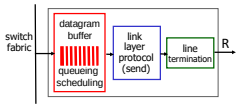
缓存容量多少合适？

- RFC 3439 经验法则：平均缓存等于“典型” RTT（例如250毫秒）乘以链路容量C
  - 例如，C = 10 Gbps 链路：2.5 Gbit 缓存
- 最近的理论和试验研究表明：在有N个流的情况下，缓存等于

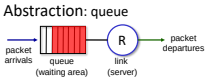
$$\frac{RTT \cdot C}{\sqrt{N}}$$

- 但是过多的缓存会增加延迟(特别是在家庭路由器中)
  - 较长的RTT：实时应用程序的性能较差，TCP响应缓慢
  - TCP协议中延迟的拥塞控制：“保持瓶颈链路刚满（忙），但不能再满。。”

缓存管理



Abstraction: queue



缓存管理：

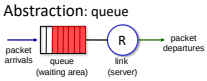
- 丢弃：缓存已满时要添加或丢弃哪些数据包
  - 弃尾(tail drop)：丢弃到达的分组
  - 优先级(priority)：按优先级丢弃/删除
- 标记：标记哪些分组以指示拥塞（ECN，RED）

分组调度：FCFS

分组调度：决定下一个在链路上发送哪个数据包

- 先来先服务(first come, first served)
- 优先权排队
- 循环排队
- 加权公平排队

Abstraction: queue

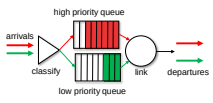


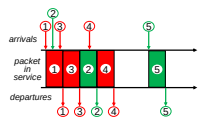
FCFS：分组按到达顺序传输到输出端口

- 也称为：先进先出(First-in-first-out, FIFO)
- 真实的例子？

分组调度：优先权排队

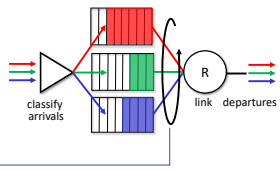
- 到达输出链路的分组被分类，按类别排队
  - 任何头部字段都可以用于分类
- 从具有缓存分组的最高优先权类中传输一个分组
  - 同一优先权类的分组之间的选择通常以FIFO方式完成





分组调度：循环排队 Round Robin Queuing

- 到达输出链路的分组被分类，按类别排队
  - 任何头部字段都可以用于分类
- 循环调度器周期性地重复扫描类队列，依次从每个类（如果可用）发送一个完整的数据包

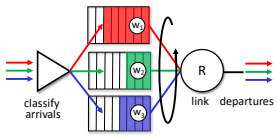


分组调度：加权公平排队
Weighted Fair Queuing

- 通用形式的循环排队
- 每个类 i 具有权重 w_i，并在每个周期中获得加权的服务部分：

$$\frac{w_i}{\sum_j w_j}$$

- 最小带宽保证



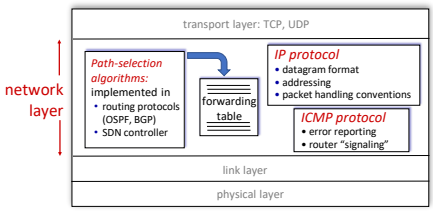
网络层：“数据平面”路线图

- 网络层：概述
  - 数据平面data plane
  - 控制平面(control plane)
- 路由器内部工作原理
  - 输入端口处理、交换、输出端口处理
  - 缓冲区管理、调度
- 网络协议IP(Internet Protocol)
  - 数据报格式
  - 编址
  - 网络地址转换
  - IPv6

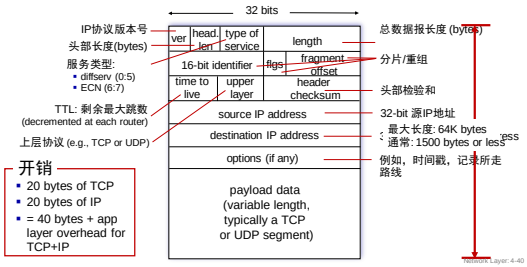
- 通用转发和SDN
  - 匹配和动作
  - OpenFlow：匹配加动作
- 中间盒子(Middleboxes)

网络层

主机、路由器网络层功能：

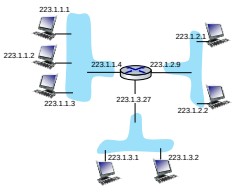


IP 数据报格式



IP寻址：简介

- IP 地址：与每个主机或路由器接口关联的32位标识符
- 接口：主机/路由器和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）



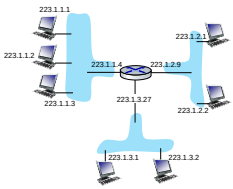
点分十进制IP地址表示法：

223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

IP寻址：简介

- IP 地址：与每个主机或路由器接口关联的32位标识符
- 接口：主机/路由器和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）



点分十进制IP地址表示法：

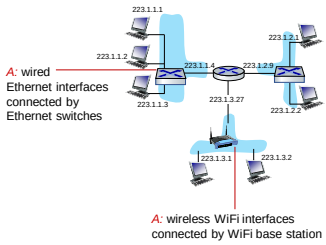
223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

IP寻址：简介

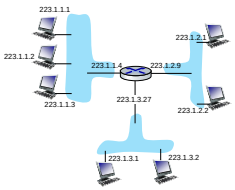
问：接口实际上是如何连接的？

答：我们将在第6、7章中了解到这一点



子网

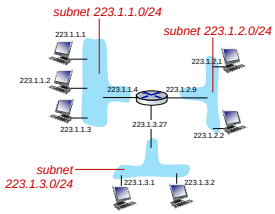
- 什么是子网？
  - 无需通过中间路由器即可物理互连的设备接口
- IP地址具有以下结构：
  - 子网部分：同一子网中的设备有公共高位
  - 主机部分：剩余低位



子网

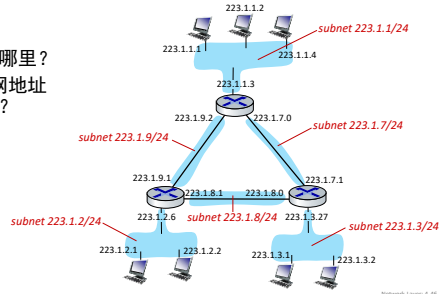
定义子网的方法：

- 将每个接口与其主机或路由器分离，创建隔离网络的“孤岛”
- 每个隔离的网络称为子网



子网

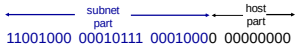
- 子网在哪里？
- /24子网地址是什么？



IP地址：CIDR

无类别域间路由选择（ Classless InterDomain Routing，CIDR）（发音为 “cider”）

- 任意长度地址的子网部分
- 地址格式：a.b.c.d/x，其中x是地址子网部分的位数



200.23.16.0/23

IP地址：如何获取？

这实际上是两个问题：

1. 问：主机如何在其网络中获得IP地址（地址的主机部分）？
2. 问：网络如何获取自身的IP地址（地址的网络部分）

主机如何获得IP地址？

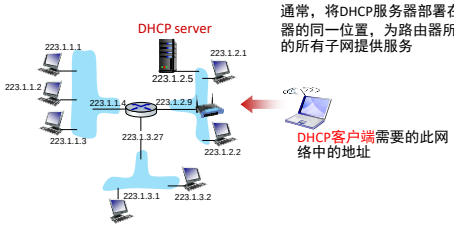
- 由sysadmin硬编码在配置文件中（例如，在UNIX中为/etc/rc.config）
- 动态主机配置协议（Dynamic Host Configuration Protocol，DHCP）：从服务器动态获取地址
  - “即插即用”

DHCP: 动态主机配置协议

目标：当主机“加入”网络时，动态地从网络服务器获取IP地址
- 可以更新其使用的地址
- 允许重复使用地址（仅在连接/打开时保留地址）
- 支持加入/离开网络的移动用户

DHCP 概述:
- 主机广播DHCP discover msg[可选]
- DHCP服务器以DHCP offer msg响应[可选]
- 主机请求IP地址：DHCP request msg
- DHCP服务器发送地址：DHCP ack msg

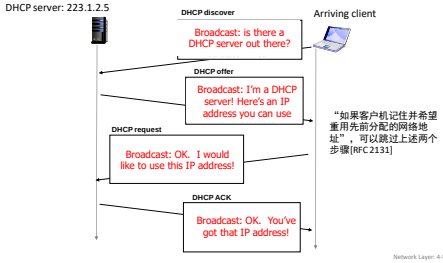
DHCP客户端-服务器方案



通常，将DHCP服务器部署在路由器的同一位置，为路由器所连接的所有子网提供服务

DHCP客户端需要的此网络中的地址

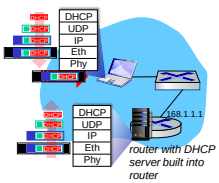
DHCP协议简要流程



DHCP可以返回的不仅仅是子网中分配的IP地址

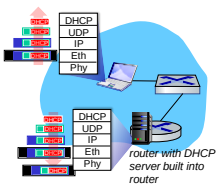
- 客户端的第一跳路由器地址
- DNS服务器的名称和IP地址
- 网络掩码（表示地址的网络和主机部分）

DHCP: 例子



- 连接笔记本电脑将使用DHCP获取IP地址、第一跳路由器地址、DNS服务器地址。
- DHCP请求消息封装在UDP中，再封装在IP中，再封装在以太网中
- 局域网上的以太网帧广播（dest:ffffffffffff），在运行DHCP服务器的路由器上接收
- 以太网多路分解到IP多路分解，UDP多路分解到DHCP

DHCP: 例子



- DHCP服务器给出包含客户端IP地址、客户端第一跳路由器IP地址、DNS服务器名称和IP地址的DHCP ACK
- 封装的DHCP服务器报文转发到客户端，在客户端多路分解到DHCP
- 客户机现在知道它的IP地址，DNS服务器的名称和IP地址，第一跳路由器的IP地址

IP地址：如何获取？

问：网络如何获取IP地址的子网部分？

答：ISP分配其地址空间的一部分

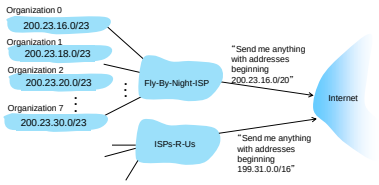
ISP's block	11001000 00010111 00010000 00000000	200.23.16.0/20

然后ISP可以按8个块分配其地址空间：

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

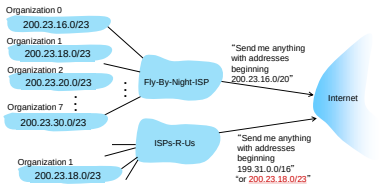
分层寻址：路由聚合

分层寻址可以有效地发布路由信息：



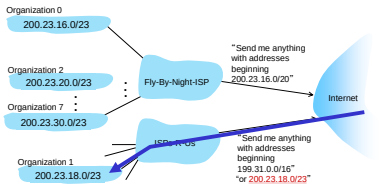
分层寻址：更具体的路由

- 组织1从Fly-By-Night-ISP迁移到ISPs-R-Us
- ISPs-R-Us发布了一条更为具体的到组织1的路由



分层寻址：更具体的路由

- 组织1从Fly-By-Night-ISP迁移到ISPs-R-Us
- ISPs-R-Us发布了一条更为具体的到组织1的路由



IP寻址：最后…

问：ISP如何获取地址块？

答：ICANN：互联网名称与数字地址分配机构http://www.icann.org/

- 通过5个区域注册管理机构(regional registries, RRs)分配IP地址（然后可以分配给本地注册表）
- 管理DNS根区域，包括单个TLD（.com, .edu, …）管理的委派

问：是否有足够的32位IP地址？

- ICANN在2011年将最后一部分IPv4地址分配给了区域注册机构
- NAT有助于缓解IPv4地址空间耗尽
- IPv6有128位地址空间

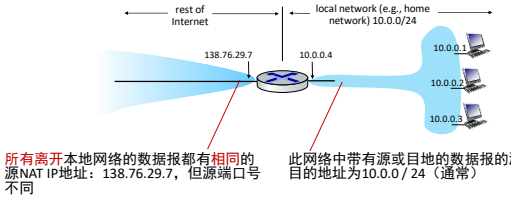
“谁知道我们需要多少地址空间？”
Vint Cerf（反思将IPv4地址设置为32位长的决定）

网络层：“数据平面”路线图

- 网络层：概述
  - 数据平面data plane
  - 控制平面(control plane)
- 路由器内部工作原理
  - 输入端口处理、交换、输出端口处理
  - 缓冲区管理、调度
- 网络协议IP(Internet Protocol)
  - 数据报格式
  - 编址
  - 网络地址转换
  - IPv6
- 通用转发和SDN
  - 匹配和动作
  - OpenFlow：匹配加动作
- 中间盒子(Middleboxes)

NAT: 网络地址转换

NAT(Network Address Translation)：就外部世界而言，本地网络中的所有设备仅共享一个IPv4地址



NAT: 网络地址转换

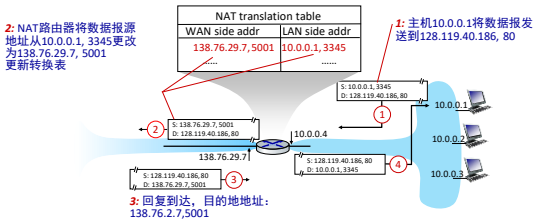
- 本地网络中的所有设备在“专用”IP地址空间（10/8、172.16/12、192.168/16前缀）中都有32位地址，这些地址只能在本地网络中使用
- 优点:
  - 供应商ISP只需为所有设备提供一个IP地址
  - 可以在不通知外界的情况下更改本地网络中主机的地址
  - 可以更改ISP而无需更改本地网络中设备的地址
  - 安全性：本地网络中的设备无法直接被寻址，外界看不到

NAT: 网络地址转换

实现：NAT路由器必须透明：

- 传出数据报：将每个传出数据报的（源IP地址，端口#）替换为（NAT IP地址，新端口#）
  - 远程客户端/服务器将使用（NAT IP地址，新端口#）作为目标地址进行响应
- 记住：在NAT转换表中每个（源IP地址，端口号）到（NAT IP地址，新端口号）的转换对
- 传入数据报：将每个传入数据报的目标字段中的（NAT IP地址，新端口号）替换为存储在NAT表中的相应的（源IP地址，端口号）

NAT: 网络地址转换



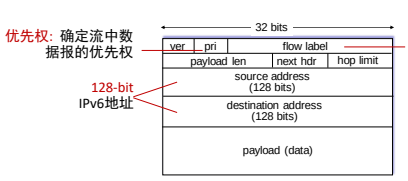
NAT: 网络地址转换

- NAT一直备受争议:
  - 路由器“应该”最多只能处理第3层
  - 地址“短缺”应通过IPv6解决
  - 违反端到端原则（网络层设备的端口操作）
  - NAT穿越：如果客户端要连接到NAT后面的服务器怎么办？
- 但是NAT仍然存在：
  - 广泛用于家庭网、机构网，4G / 5G蜂窝网

IPv6: 动机

- 最初的动机：32位IPv4地址空间将被完全分配完
- 其他动机：
  - 处理/转发速度：40字节固定长度的报头
  - 支持对“流”进行不同的网络层处理

IPv6 数据报格式



优先权：确定流中数据报的优先权

流标签：识别相同“流”中的数据报。（“流”的概念没有明确定义）。

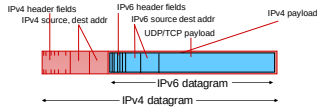
128-bit IPv6地址

缺少什么（与IPv4相比）：

- 没有校验和（以加快路由器的处理速度）
- 没有分片/重组
- 无选项（可能出现在 IPv6 首部中由 “下一个首部 ”指出的位置上）

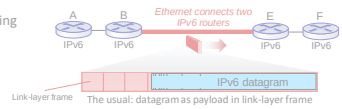
从 IPv4 到 IPv6 的迁移

- 并非所有路由器都可以同时升级
  - 没有“标志日”
  - 混合IPv4和IPv6路由器的网络将如何运行？
- 隧道：在IPv4路由器之间作为IPv4数据报中的有效载荷携带的IPv6数据报（“数据包中的数据包”）
- 在其他情况下广泛使用的隧道（4G / 5G）

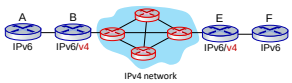


隧道和封装

Ethernet connecting two IPv6 routers:

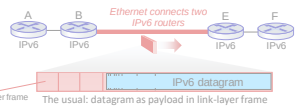


连接两个IPv6路由器的IPv4网络

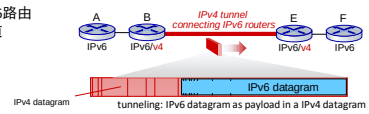


隧道和封装

Ethernet connecting two IPv6 routers:



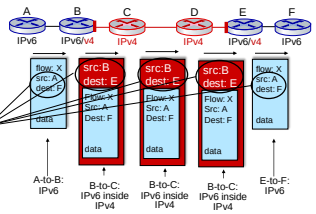
连接两个IPv6路由器的IPv4隧道



隧道

逻辑视图：

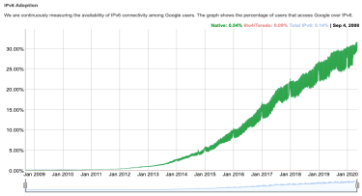
物理视图：



注意源地址和目的地址！

IPv6: 采用

- Google[1]: 约30％的客户端通过IPv6访问服务
- NIST: 1/3的美国政府域名支持IPv6



1 https://www.google.com/intl/en/ipv6/statistics.html

IPv6: 采用

- Google[1]: 约30%的客户端通过IPv6访问服务
- 长时间（很长！）的部署
  - 25年了！
  - 考虑一下过去25年中应用程序的变化：WWW，社交媒体，流媒体，游戏，远程呈现，…
  - 为什么？

[1] https://www.google.com/intl/en/ipv6/statistics.html

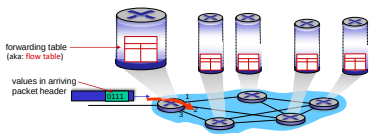
网络层："数据平面"路线图

- 网络层：概述
  - 数据平面data plane
  - 控制平面(control plane)
- 路由器内部工作原理
  - 输入端口处理、交换、输出端口处理
  - 缓冲区管理、调度
- 网络协议IP(Internet Protocol)
  - 数据报格式
  - 编址
  - 网络地址转换
  - IPv6
- 通用转发和SDN
  - 匹配和动作
  - OpenFlow：匹配加动作
- 中间盒子(Middleboxes)

通用转发：匹配加动作

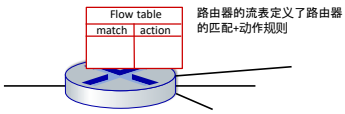
回顾：每个路由器包含一个转发表（又名：流表）

- "匹配加动作"抽象：匹配到达数据包中的位，执行操作
  - 基于目的地的转发：基于目的地转发， IP地址
  - 通用转发:
    - 许多头部字段可以决定操作
    - 可能有许多动作：删除/复制/修改/记录数据包



流表抽象

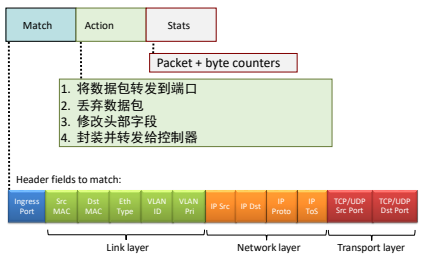
- 流：由头部字段值定义（在链接，网络，传输层字段中）
- 通用转发: 简单的数据包处理规则
  - 匹配：数据包头字段中的值
  - 动作：对于匹配的数据包：丢弃、转发、修改、匹配数据包或将匹配的数据包发送给控制器
  - 优先权：当多个匹配时，消除歧义
  - 计数器：#字节和#数据包



流表抽象

- 流：由头部字段值定义（在链接，网络，传输层字段中）
- 通用转发: 简单的数据包处理规则
  - 匹配：数据包头字段中的值
  - 动作：对于匹配的数据包：丢弃、转发、修改、匹配数据包或将匹配的数据包发送给控制器
  - 优先权：当多个匹配时，消除歧义
  - 计数器：#字节和#数据包

OpenFlow:流表条目



OpenFlow: 例子

基于目的地的转发：

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	51.6.0.8	*	*	*	*	port6

发往IP地址51.6.0.8的IP数据报应转发到路由器输出端口6

防火墙：

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	*	*	*	22	drop

阻止（不转发）发往TCP端口22（ssh端口号）的所有数据报

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	128.119.1.1	*	*	*	*	*	drop

阻止（不转发）主机128.119.1.1发送的所有数据报

OpenFlow: 例子

第2层基于目的地的转发：

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	22:A7:23:11:E1:02	*	*	*	*	*	*	*	*	*	port3

具有目标MAC地址为22:A7:23:11:E1:02的第2层帧应转发到输出端口3

OpenFlow 抽象

- 匹配+动作: 抽象统一了不同类型的设备

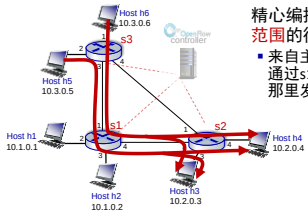
路由器
- 匹配: 最长目的IP前缀
- 动作: 转发链接

交换
- 匹配: 目的MAC地址
- 动作: 转发或泛洪

防火墙
- 匹配: IP地址和TCP / UDP端口号
- 动作: 允许或拒绝

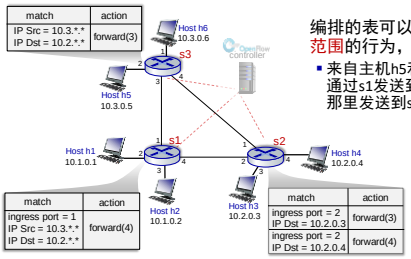
NAT
- 匹配: IP地址和端口
- 动作: 重写地址和端口

OpenFlow 例子



精心编排的表可以创建网络范围的行为，例如：
- 来自主机h5和h6的数据报应通过s1发送到h3或h4，再从那里发送到s2

OpenFlow 例子



编排的表可以创建整个网络范围的行为，例如：
- 来自主机h5和h6的数据报应通过s1发送到h3或h4，再从那里发送到s2

通用转发: 总结

- “匹配+动作”抽象：在任何层中匹配到达的数据包报头中的位，并采取动作
  - 匹配多个字段 (链路层，网络层，传输层)
  - 本地行动: 丢弃，转发，修改或将匹配的数据包发送到控制器
  - 在网络范围内“编程”行为
- “网络可编程性”的简单形式
  - 可编程的，按数据包进行“处理”
  - 历史：主动网络
  - 今天：更通用的编程：P4（请参阅P4.org）。

网络层：“数据平面”路线图

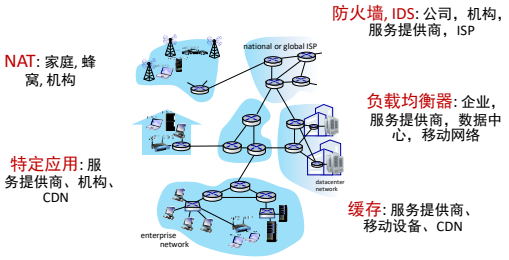
- 网络层：概述
- 路由器内部工作原理
- 网际协议IP(Internet Protocol)
- 通用转发和SDN
- 中间盒子(Middleboxes)
  - 中间盒子功能
  - 互联网的演化、架构原理

中间盒子

中间盒子(Middlebox) (RFC 3234)

“any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host”

中间盒子无处不在

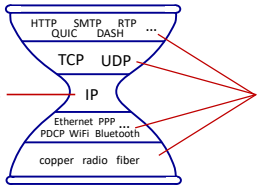


中间盒子

- 最初: 专有（封闭）硬件解决方案
- 转向实现开放API的“白盒”硬件
  - 摆脱专有硬件解决方案
  - 通过匹配+动作实现的可编程本地操作
  - 迈向软件创新/差异化
- SDN:（逻辑上）通常在私有/公共云中进行集中控制和配置管理
- 网络功能虚拟化（NFV）: 白盒网络、计算、存储上的可编程服务

IP 沙漏

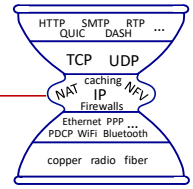
互联网的“细腰”:
- 一个网络层协议：IP
- 必须由每（数十亿）个互联网连接设备实现



物理层、数据链路层、传输层和应用层中的许多协议

目前的IP沙漏

- 在网络内部运行的中间盒子



互联网的结构原理-RFC1958

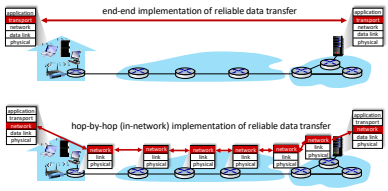
RFC 1958

"Many members of the Internet community would argue that there is no architecture, but only a tradition, which was not written down for the first 25 years (or at least not by the IAB-Internet Architecture Board). However, in very general terms, the community believes that the goal is connectivity, the tool is the Internet Protocol, and the intelligence is end to end rather than hidden in the network."

三个基本信念：
- 简单的连接
- IP协议：窄腰
- 网络边缘的智能、复杂性

端到端原则

- 一些网络功能（例如可靠的数据传输、拥塞）可以在网络中或网络边缘实现



端到端原则

- 一些网络功能（例如可靠的数据传输、拥塞）可以在网络中或网络边缘实现

"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)

We call this line of reasoning against low-level function implementation the "end-to-end argument."

Saltzer, Reed, Clark 1981

智能在哪里?

20th century phone net:
- intelligence/computing at network switches

Internet (pre-2005)
- intelligence, computing at edge

Internet (post-2005)
- programmable network devices
- intelligence, computing, massive application-level infrastructure at edge

第4章：完成！

- 网络层：概述
- 路由器内部工作原理
- 网际协议IP(Internet Protocol)
- 通用转发和SDN
- 中间盒子(Middleboxes)

Question: 如何计算转发表（基于目的地的转发）或流表（通用转发）？
Answer: 通过控制平面（下一章）

作业

1. 在IP首部中，哪个字段能用来确保一个分组的转发不超过N台路由器？
2. IP地址223. 1. 3. 27的32比特二进制等价形式是什么？
3. 考虑使用8比特主机地址的数据报网络。假定一台路由器使用最长前缀匹配并具有下列转发表：

前缀匹配	接口
1	0
10	1
111	2
其他	3

对这4个接口中的每个，给出相应的目的主机地址的范围和在该范围中的地址数量。

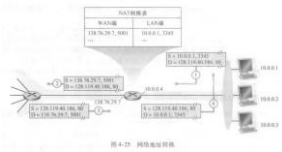
4. 考虑一个具有前缀128. 119. 40. 128/26的子网。给出能被分配给该网络的一个IP地址（形式为 xxx. xxx. xxx. xxx）的例子。假定一个ISP拥有形式为128. 119. 40. 64/26的地址块。假定它要从该地址块生成4个子网，每块具有相同数量的IP地址。这4个子网（形式为a. *b. c. d/x*）的前缀是什么?

作业

5. 考虑在图4-25中建立的网络。假定ISP现在为路由器分配地址24.34.112.235，家庭网络的网络地址是 192.168.1/24。

a. 在家庭网络中为所有接口分配地址。

b. 假定每台主机具有两个进行中的TCP连接，所有都是针对主机128.119.40.86的80端口的。在NAT转换表中提供6个对应表项。



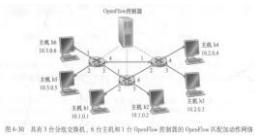
作业

6. 考虑显示在图4-30中的SDN OpenFlow网络。

假定对于到达s2的数据报的期望转发行为如下：

- 来自主机h5或h6并且发往主机h1或h2的任何数据报应当通过输出端口2转发到输入端口1。
- 来自主机h1或h2并且发往主机h5或h6的任何数据报应当通过输出端口1转发到输入端口2。
- 任何在端口1或2到达并且发往主机h3或h4的数据报应当传递到特定的主机。
- 主机h3和h4应当能够向彼此发送数据报。

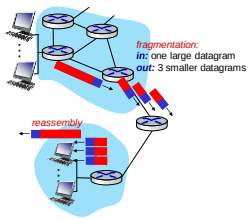
详述实现这种转发行为的s2中的流表项。



第4章附加的幻灯片

IP 分片/重组

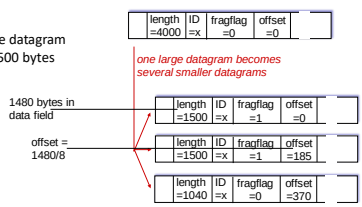
- 网络链接具有最大传输大小MTU (max. transfer size) -最大可能的链接级别帧
  - 不同的链路类型，不同的MTU
- 大型IP数据报在网络中被分割（“分片”）
  - 一个数据报变成几个数据报
  - 仅在目的地“重组”
  - IP头部位用于识别，排序相关片段



IP 分片/重组

例子:

- 4000 byte datagram
- MTU = 1500 bytes



DHCP: Wireshark output (home LAN)

request

```
Message type: Boot Request (1)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Request
Option: (61) Client identifier
    Length: 7; Value: 010016D323688A;
    Hardware type: Ethernet
    Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
    Length: 11; Value: 010F03062C2E2F1F21F92B
    1 = Subnet Mask; 15 = Domain Name
    3 = Router; 6 = Domain Name Server
    44 = NetBIOS over TCP/IP Name Server
    .....
```

reply

```
Message type: Boot Reply (2)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
    Length: 12; Value: 445747E2445749F244574092;
    IP Address: 68.87.71.226;
    IP Address: 68.87.73.242;
    IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."
```