第一章 通信网络体系架构

引言 网络的作用

1.1 网络体系结构

什么是网络体系结构？网络分层结构、服务与协议的集合
为什么要进行网络分层？相互通信非常复杂，将复杂问题简单化，网络功能分层
网络体系架构研究的主要内容：分层结构模型、层间接口以及各层协议

1.1.1 层次结构

5 层/7 层模型

三个部分：信息通信，网际通信，数据通信
OSI：7 层模型：应用层，表示层，会话层，传输层，网络层，数据链路层，物理层
5 层模型：应用层、传输层、网络层、数据链路层、物理层

网络层的重要性

核心层次：承上启下，穿越源端-目的端之间的所有网络

OSI 参考模型中的重要概念

服务：上层使用下层的服务，下层向上层提供服务
接口 SAP：层间交互的参数和内容，以及访问方式
协议：同层实体（对等实体）之间通信使用的一组规则和约定

TCP/IP 参考模型的特点

在传输层和网络层定义具体的主协议，并描述协议的具体细节
核心思想：在网络层用 IP 协议将不同的物理网互连起来，从而实现 IP 分组从源端到目的端的传递

OSI 模型与 TCP/IP 模型的比较

共性：分层，网络层以上都是端端通信，与具体网络无关
差异：OSI 是抽象的网络框架模型，TCP/IP 是具体的实现模型
OSI 模型是一个抽象的通用模型，不定义具体协议。但 ISO 为每层制定了独立的协议标准，不属于模型本身。

TCP/IP 层次通信模型

物理网取代 L1 和 L2
物理网：灵活多样

高层的合并

“云”化，模糊了“端”的概念，结合成一个整体

—通常使用的网络参考模型

修改的 OSI 参考模型（五层结构）

1.1.2 协议体系与协议栈

协议栈

协议栈与协议实体

实通信，虚通信
服务、协议、SAP（Service Access Point）

相邻上下层协议实体之间的通信

服务：上下层协议实体之间的通信
通过服务访问点 SAP 实现上下层协议实体之间的通信
“分组”：数据分割成小块传输

对等实体之间的通信

对等实体：通信双方的同层实体，且执行**相同协议**的实体
协议：对等实体间的对话规则
协议(对话规则)三要素：语法，语义时序

层间实体通信与对等实体通信的关系

层间实体：实通信；对等实体：虚通信
实通信是实现虚通信的基础

构建的基本问题

问题 1：每个协议实体上都可堆砌其它实体吗？
问题 2：实体间的连线可任意画吗？有什么规律可循？
问题 3：实体堆砌是否须按层次关系？
问题 4：要放置一个新协议进栈，有什么规则可循？

从协议类型看问题 1 和 2

通信型协议：提供通信服务，提供多条分组传递服务路径，具有上下的通道（接口）
功能型协议：实现某些功能、不提供通信服务，只有向下的通道（接口）

从对等关系看问题 3 和 4

对话交流：
TCP：端到端对话，用 TCP 报文
IP：逐跳对话，用 IP 报文
链路：分段对话，用数据帧
关键点：所有对话都发生在对等实体间

如何保证对等关系

技术方法：SAP Notation

协议栈构建准则

对等实体准则：需要知道一个协议实体在与谁进行对话
上下实体独立准则：实体不必关心 SAP 上面究竟放置了哪些实体
透明传输准则：对等实体间可传递任意数据、接收内容=发送内容

协议设计的重要思想

模块化思想

IP 协议栈详情

上层部分：简洁的传送协议和丰富的应用协议
网络层：本身很简单，需要很多辅助协议来帮助它
下层部分：丰富的数据通信系统

生态环境

IPv4：发育良好

1.1.3 网络拓扑结构

拓扑结构：物理拓扑，逻辑拓扑

概念

拓扑结构：网络系统组成的抽象表达，分为物理拓扑和逻辑拓扑
物理拓扑：网络节点互连形成的结构
逻辑拓扑：建立在物理拓扑之上

逻辑结构

按照属主划分

功能结构

接入网：为用户驻地网用户提供接入，网络结构决定于接入方式、功能、性能需求

汇聚网：有交叉、有冗余、不断流的汇聚网络

核心网：大流量、高速率，各向均衡分布

通信结构

无线局域网，无线广域网，按需布局网络拓扑，GPS、北斗，物联网、D2D、M2M

互联网思路：拓扑分层

应对日益扩大的网络

分层拓扑带来的核心问题

路由问题：网内路由、区域内路由、域间路由、核心网内路由、核心网间路由

Internet 网络结构

无尺度网络（不是随机网络）

对网络体系结构的理解

从宏观上理解网络：层次结构，协议体系，拓扑结构，路由结构

1.2 端到端原理（E---E 原理）

1.2.2 网络功能部署的相关问题讨论

端端通信的可靠性由端系统考虑？还是应该由网络来考虑？

多播功能由端系统实现？还是由网络来实现？

1.2.3 网络功能部署基本思想

足够就好，除非必要，如果有用

结论：网络功能简单化原则

1.3 网络三平面架构与典型的协议栈

1.3.1 网络的三平面架构

背景

三平面架构：传送平面、管理平面、控制平面

描述

传送平面：基本的用户业务承载数据传送功能，也称数据平面或用户平面

管理平面：执行系统理功能

控制平面：强化网络运营的控制特性

案例

ATM 网络，IP 接入网络，LTE/4G/5G 网络

1.3.2 LTE/4G 网络结构

3G 到 4G/5G 的变化

4G 的部分指标

与计算机网络的比较

网络结构，网络协议栈，网络控制方式

网络架构：核心网+接入网

网元主要功能描述

eNB，MME，S-GW，P-GW，HSS，PCRF

主要接口

Uu，X2，S1-MME，S6，S7，S11，S1-U，S5/S8，SGi

从 LTE/4G 的需求看网络结构的设计思想

用户数据的需求：方便传输，资源访问快捷，需要分布式架构

控制信息的需求：传输可靠，便于决策和控制，需要集中式架构

E-UTRAN 部分，E-UTRAN 与 EPC 之间，EPC 部分

将控制平面与用户平面分离

1.3.3 LTE/4G 协议栈

控制面协议栈

NAS 层，RRC 层，PDCP 层，RLC 层，SCTP，S1-AP，GTP-C，X2-AP，Diamteter，S10 接口

UE 接入的主要信令

1. UE 选择小区
2. 接入 E-Node B
3. 附着（Attach）

用户面协议栈

PDCP，RLC，MAC，GTP-U(GTP-User)

各网元协议栈(部分)

目前 LTE 的几种语音解决方案

三种语音解决方案的比较

SvLTE，CSFB，VoLTE/SRVCC

中外 4G 网络对比

中国：TD-LTE

美国：FDD-LTE

1.3.4 5G 网络结构与协议栈

5G 的目标

开启万物互联之门、海量用户，高密度接入、极低延迟、高数据传输速率、高速运动场景、各种新型业务

5G 的挑战

频谱资源、信道、功率、干扰、器件、无缝接入

发展历程

各国部署计划

应用场景

eMBB 大带宽，uRLLC 大规模连接，mMTC 低时延

核心网(5GC)，接入网(NG-RAN)

主要模块的功能

gNB/ng-eNB，AMF，UPF，SMF

SBA 架构

Service-based architecture，基于服务的网络架构

网络功能模块缩写

模块后面都是 Function

控制面协议栈

NAS-SM，NAS-MM，5G-AN Protocol layer

UE 通过 gNB 接入 GC

用户面协议栈

多了一层 SDAP（Service Data Adaptation Protocol）

与 4G 的不同

基站重构

gNB 重构为 CU+DU+RRU(或 RRH)

核心网部分功能下沉到 CU

BBU 部分功能移至 RRU(RRH)

好处

接入网的变化

5G 采用 C-RAN 接入网架构

核心网的变化

化整为零、由硬件变软件

控制面：4G 中 MME 的功能被分解到 AMF 和 SMF 中

用户面：原来 4G 中 SGW 和 PGW 被 UPF 替代

主要技术

网络虚拟化，云计算，移动边缘计算，网络切片

关键通信技术

覆盖增强，频效提升，频谱拓展，能效提升

网络融合-同构网络融合

从 3G、4G、5G 的演进看网络融合

异构网络的融合

5G 指标如何实现？解决方法：异构网络融合。

融合的难题

第二章 通信网络协议技术

2.1 协议的定义与协议模型

协议的定义：计算机网络中通信双方所遵守的规则、标准或约定的集合

协议的基本模型(协议三要素)：实体、服务、对话

协议模型

实体：协议实现和执行

服务：实体对上的接口

对话：信息的表示与交互的规则

2.1.1 协议实体

两种类型：主协议(通信型协议)，辅助协议(事务型协议)

例 1：IP 协议

IP 协议的完整性考虑：与下层协议的适配，出错的情况处理

性能问题：将 IP 实体看作一个线程

对性能影响较大的事务从 IP 中分离

IP、ARP、ICMP 并行处理

例 2：PPP 协议

两点间传输分组、如下子功能全部或部分组合

协议设计：多种类型信道帧，灵活实现子功能的各种裁剪与组合

设置几个事务型协议：LCP，PAP/CHAP，NCP

小结

协议功能设计(模块化思想)

2.1.2 协议的服务

服务如何提供？服务通过对上提供的接口来实现

提供服务接口模型：实体提供多个功能相同、地位相等的服务接口(SAP)

使用的服务接口：一般只使用一个实体提供的服务（网络、应用实体除外）

网络协议的服务

网络协议实体：使用多个实体提供的服务

应用协议的服务

应用协议实体：使用多个服务

小结

提供服务：一个 SAP 代表一个分组流，实体在分组首部用 SAP 来区分不同的分组流

使用服务：特定层次实体使用下面的多个服务

2.1.3 协议的对话

协议(对等协议)之间交互的规则（协议的三要素）

交互信息的格式—格式定义、信息交互控制

交互信息的内容—信息的表示

交互信息的顺序—时序如何控制

2.2 协议三要素

语法：各层实体 PDU 格式定义

语义(内容)：分组格式中如何表达对话内容

时限：需要等待吗、等多长时间

2.2.1 语法格式和语义的表示

协议用 PDU 首部（H）携带对话语义及参数，DU 携带的是其它（上层）实体的通信数据

应用层消息，传输层段（数据报），网络层包，数据链路层帧，物理层比特流

几种协议的语义

定位编码法

用固定字节数存放信息内容：每种信息在固定位置、占固定字节长度，用编码形式表示信息内容

固定字节位置存放固定 Key 的内容：优点：占用空间小，内容为机器所理解

协议 PDU 形式：一种协议，一种或多种报文格式

固定长度的首部：特定参数存放在特定位置。处理高效、但扩展不易

例：IP 报文的设计

IP 地址字段：固定的 4 字节

IP 报文分段(分片)：在不同 MTU 的物理网上传输

IP 头部应该如何表示分段的 IP 报文？

Identification：分组序号（16bits）

Frag Offset：段偏移值(13bits)，单位：8 字节

分段标志（3bits）

如果 IP 地址是可变长度的，问题是什么？

如果首部有一个或多个域长度可变，如何应对？

是为这些域预留足够长度？

如果某些域时有时无，如何应对？

另外两种技术

变长编码法：长度可变、可跟踪长度变化的编码

文字表述法：供机器阅读的语言

变长编码法

ISO　ASN.1 标准(Abstract Syntax Notation　#1)

TLV（Type+Length+Value）表示法的特点：变长表示，任意排列，效率不高，一错全错

用一个 TLV 把序列装起来(Sequence of)：检测 L 错误，快速定位，多次分组

SNMP(简单网络管理)协议是采用 TLV 表示分组首部的协议

文字表述法

用可显式字符串表示分组各个域

特点：技术支撑好，弹性大，扩展性强

XML：http 的强力扩展

其他文字表述法：如 JSON，键+值

2.2.2 协议的语法与时限

协议信息交互的逻辑顺序——语法

协议信息交互的时长与时限——时序

2.2.2 协议交互语法与类型

2.2.2.1 协议交互类型

三种类型：无应答交互，简单应答交互，序列可靠交互

分析基础

仅考虑物理信道的误码产生的 PDU 出错

所有 bit 正确，则 PDU 正确

2.2.2.2 无应答交互

基本特征：对等实体都可向对方发送 PDU，各个 PDU 互不相关，发送一个后就可发送下一个

分析

作为其它交互的比对基线

几个重要参数：

PDU 阻塞延迟$t_d = T_p$（PDU 发送时延）

最大 PDU 发送速率$r_p = R_p = \frac{1}{T_p}$

发送失败概率$p_f = P_f$（PDU 出错的概率）

出错了不管，不影响交互过程

2.2.2.3 简单应答交互

形式为单次的“请求—应答”交互

几个重要参数的表现：

PDU 阻塞延迟$t_d = T_{p1} + RTT + T_{p2}$（有时可令简单应答无时延，即$T_{p2} = 0$）

最大 PDU 发送速率$r_p = R_p \cdot \frac{T_{p1}}{T_{p1}+RTT+T_{p2}} = \frac{1}{t_d}$

发送失败概率$p_f = p_{f1} + p_{f2} - p_{f1}p_{f2}$（双方有一方出错，有时可令简单应答不出错，即$p_{f2} = 0$）

出错了必须管，否则影响下一次交互

分析

可能出错的情况：Req（Data）出错，Resp（ACK）出错

结果：请求方均收不到应答

解决方案：设置超时时限，设置超时重传机制

失败概率

如果交互不成功，再重传请求

重传 K 次不成功的概率：$P_K = \left(p_{f1} + p_{f2} - p_{f1}p_{f2}\right)^{K+1}$

阻塞延迟

一次交互成功的阻塞延迟

下一 PDU 发送最快将在：$t_d = T_{p1} + RTT + T_{p2}$后开始

若不成功，则产生重传：下一 PDU 发送将延迟$T_d = n\left(T_{p1} + RTT + T_{p2}\right)$(重传次数 n=1,2,3,…）

协议的最大交互能力(所有交互都一次成功)：$r_p = R_p \cdot \frac{t_{p1}}{t_{p1}+RTT+t_{p2}}$

抖动概率

在接连的多次交互中，出现了个别的长延迟

首次交互失败就会发生抖动，其概率为$p_{抖动} = p_{f1} + p_{f2} - p_{f1}p_{f2}$

重复数据

出现重复的原因是应答的丢失，重复概率为p_{f2}（重复 n 次概率为p_{f2}^n）

小结

简单交互+超时重传：有如下特性：

降低了交互失败概率

存在发送阻塞延迟

存在请求 PDU 重复的概率

在网络体系结构中看简单交互：层次越低的协议，影响面越大

2.2.2.4 序列可靠交互

连续的若干个分组，彼此存在前后顺序的逻辑关系，要求可靠地传递到对方。

三种典型的交互方式：停等协议，回退 N 协议，选择性重传

分析

可沿用简单应答交互的分析方法：Data-Ack 交互，Data PDU 出错为主线索

近似分析

特性

与简单应答交互相似：传输可靠性，存在发送阻塞，重复分组

不同点：存在终止通信的可能性，保证前后顺序的逻辑关系

关于停等协议的帧序号

需要序号来解决问题

前后顺序的问题

为什么会产生顺序问题

发送窗口大小的影响

出错的根本原因：接收方滑动窗口后，新的序号范围与老的序号范围重叠，无法根据序号判断是新帧还是旧帧

如何保证前后顺序的逻辑关系？

限制滑动窗口大小

序号循环使用：发送窗口大小 ≤ （序号最大值＋1）/ 2

协议交互事例

PPP 协议：启动链路的交互过程

WiFi 发送数据 PDU 的交互

2.3 无连接 vs 连接型协议

2.3.1 背景——两大阵营及技术路线

两大技术路线：面向连接的技术和无连接技术
典型的面向连接的技术：电路交换，虚电路交换
典型的无连接的技术：数据报交换
虚电路交换、数据报交换都是分组交换

2.3.2 交换技术——电路交换与分组交换

电路交换：在源和目的之间建立一条专用通路，通信期间信道独占。
信道独占，顺序到达，交换时延短，几乎无时延抖动，信道利用率低
分组交换：以分组为单位传输，在每个交换节点上存储转发，分为数据报交换和虚电路交换
信道不独占，信道利用率高，可能高时延，可能丢分组，可能乱序到达

交换技术

电路交换与分组交换在延时上的表现
电路交换：透明
分组交换：存储转发

电路交换与分组交换效率

电路交换：建立时延大，传输时延小。
分组交换：不需要建立，传输时延大。

面向连接的端到端延时计算

电路交换：没有交换时延。

无连接的端到端延时计算

分组交换：交换节点接收存储+寻路

信道利用情况

电路交换独占信道：用户没有数据时，信道仍被保留
分组交换共享信道：根据数据传输需要分配信道

小结

电路交换虽然有较短的过站延时，但是建立通路的时间比较长
分组交换的信道利用率高，减少分组交换过站延时可从两个方面入手（存储延时+寻径延时）

面向连接的协议与无连接协议

面向连接协议：建立一条从源到目的的电路/虚电路
无连接协议：对网络的多样化适应性强，在互联网层面上无法提供可靠性

2.4 协议设计的问题讨论

2.4.1 可靠性

可靠性涵盖哪些内容？差错控制，流量控制，顺序控制
如何设计可靠性协议？面向连接，重发机制，确认机制，PDU 编号，流控机制

2.4.2 健壮性

健壮性：对网络中出现意外或设备故障情况下的适应性或应变能力
无连接的协议具有好的健壮性、通用性

IP：不可靠无连接数据报协议

连接 vs. 可靠性
面向连接不一定保证可靠，无连接不一定不可靠
在连接上容易实现可靠性机制
IP 通信：可靠性问题交由高层协议解决

2.4.3 网络拥塞

网络拥塞：分组堆积
网络流量过于集中，超过节点处理能力、信道传输能力

拥塞是互联网的天然问题

互联网的流量具有突发特性
如果一个短时间内涌入大量的分组
发生拥塞时，网络性能下降

拥塞—死锁

一个点的拥塞会向全网蔓延
拥塞时丢弃分组或过长延时，使源节点重传，进一步加剧拥塞

拥塞控制的基本方法

开环策略：预防和避免拥塞（缓冲区预分配，分组丢弃，网络分组定额控制）
闭环策略：检测和采取措施（拥塞检测，反馈）

以 TCP 为例

TCP 应对网络拥塞的交互过程
多种因素制约下的交互（分组交换网环境、TCP 通信协同）

流控 vs 拥塞控制

流控：避免一个快速发送者用分组淹没慢速接收者
拥塞控制：避免一组发送者用分组淹没网络

TCP 拥塞控制的研究历程

TCP 建立连接的交互过程

建立连接—三次握手
目的：因双方使用起始序号随机，需要确保把自己的起始序号告诉对方

TCP 的数据交互过程

基于发送窗口尺寸的技术

Jacobsom 提出的 CC 方法

扩展基于窗口的协议，让窗口与拥塞联系起来
使用两个窗口 CWnd 和 RWnd
实际发送窗口： W= min{CWnd，RWnd}
假定：RWnd >> CWnd
方便起见，CWnd 的单位设为 MSS
CC 方法：动态调整 Cwnd 尺寸
三个方面的问题：如何检测拥塞？如何增加 CWnd？如何减少 CWnd？

检测拥塞—有什么方法？

让网络通知，分组延迟增加，分组丢失，超时

如何调整 CWnd？

缓慢增加窗口，快速减小窗口

Slow Start+AIMD

Slow Start(慢启动)，CWnd 呈指数上升趋势

解决措施

拥塞避免

AI: Additive increase (加性增)

MD：multiplicative decrease (乘性减)

进一步考虑

等待超时，是否时间太长了？

解决措施 (TCP Reno)

快速重传：当收到 3 个重复的 Ack 后，立即开始重传(不必等超时)

Threshold = CWnd/2

CWnd = Threshold

TCP—Reno (Jacobson 1990)

吞吐量公式

假定：分组丢失总发生在 CWnd 达到 Wmax 时

TCP 吞吐量模型

与 RTT 成反比、与报文丢失率的开平方成反比

如果两个如图的 TCP 一前一后启动，公平性如何

AIMD 启动很慢

TCP 更多的讨论

单个 TCP 流，可变带宽

多个 TCP 流，分享带宽

TCP 公平性分析

假设两个 TCP 流共享网络通信带宽

公平性如何分析？

AIAD，MIAD，MIMD，AIMD

AIMD 动态公平性

只有 AIMD 具有公平性

AIMD 有多公平？

取决于 RTT

TCP：适应高速网络吗？

超长的分组丢失间隔-不现实！TCP 不适应！

TCP 以外的方法-SACK

改变 TCP 的确认机制

使用 SACK 时，NewReno 算法可不使用

补充内容：网络协议簇

TCP/IP 协议簇

编址（Addressing）

物理地址、IP 地址、端口地址

TCP/IP 中的地址与层次关系

物理地址（网络接口层），逻辑地址（网络层），端口地址（传输层）

目的地址：单播、多播、广播

TCP/IP 的版本

第三章 中继与交换原理

3.0 典型中继设备

以太网交换机，IP 路由器，2/3/4 层交换机，WiFi 路由器

引言

中继：从一个端口接收 PDU，从另一个端口转发 PDU

程式问题

中继模型，中继处理，中继排队

图标

3.1 中继模型

引言

中继例子

IP 路由器

PDU、SDU 与 ICI

Protocol Data Unit 协议数据单元

Service Data Unit 服务数据单元

Interface Control Information 接口控制信息

两种中继类型

协议实体内部的 relay(协议内中继)

协议外部的中继(协议外中继)

——中继的差别

协议外中继

可以进行不同协议间的中继，非协议功能，DU 中继

协议内中继

协议掌握 PDU 的内容，用首部信息转发 PDU，PDU 中继

例：L2 中继和 L3 中继

L2：物理网内部的中继，中继实体不是协议实体

L3：中继实体是上层协议实体

3.2 中继协议栈

3.2.1 中继实现层次

哪些协议(主协议)定义了中继功能？

没有定义中继功能的层次上如何实现中继？

为什么要在这些层次实现中继？

为什么不在这些层次的协议中考虑中继？

链路层中继

链路层协议不定义中继功能：链路层中继都是协议外中继

作用：链路的延伸或扩展

例

以太网交换机：多口中继

PPP 中继：只能实现一对一中继

PPP—以太网中继：不可实现

网络层中继

协议内中继—网络协议的组网、路由、转发（路由器）

协议外中继—从一种网络进入另一种网络（网关）

例

IP 路由器

IP4/6 网关

传送层中继

传送层：通信端点，无中继功能

传送层中继：往端点外延伸的通信

不是 TCP—UDP，而是 TCP—TCP、UDP—UDP

TCP 和 UDP 如何实现外中继？在不同 SAP 之间中继。

例

将业务转移给合适的服务器

3.2.2 中继与对等实体

协议内中继：根据对等实体准则，该模型中各实体都是相同协议

协议外中继：中继两端的协议可以不是相同协议，可以实现协议转换

协议外中继不一定可行

协议内中继由协议功能做了很好的定义

协议外中继是否都可行，存在疑问

3.2.3 协议外中继的条件

实现两种地址空间的某种映射关系

实现两种协议的 SAP 间的某种映射关系

实现不同最大传输单元(MTU)的适配

地址映射

不同地址空间的映射是一个极大障碍

例

IPv4、IPv6 的地址映射（可以映射）

以太网（MAC）到 PPP 的中继（不能映射）

SAP 映射问题

MAC 和 PPP 中继情况

中继需了解所中继的 DU 是什么

MTU 问题

如果协议没有对 DU 分段重组功能：向下兼容

更多的例子

WiFi 无线网桥

802.3—802.11 中继(WiFi)

两者协议不同，但具有相同的地址格式和服务原语

无线路由器

将无线网桥中的交换实体替换为 IP 实体

中继由 IP 协议实体实现

MPLS

MPLS 首部只有一个域：流标签

根据（入口，标签值）选择中继出口

MPLS 交换效率非常高，超高速分组交换上应用广泛

用 PPP 协议将两端的以太网连接起来(远程网桥)

PPPoE

中继与互联

同层不同实体之间不能直接互连

网络互联技术

协议转换

802.3 与 Token Ring 协议的转换

隧道技术

LAN 通过 PSTN 远程互联（2 层）

LAN 通过 X.25 互联（3 层）

LAN 通过 Internet 互联（4 层）

IPV6 通过 IPV4 互联（4 层）

隧道技术的缺点

协议覆盖方式

需要互通但又无法实现协议转换

在上一层次寻求协议的统一

例：不同的链路层在网络层统一为 IP

通过网络层的路由功能实现网间互联

减轻了隧道需求压力

网络互联技术比较

3.3 中继性能分析

中继吞吐性能：选路分组数/秒

通信服务质量：通信的实时性、公平性等

选路算法

查表匹配算法，FIB (forwarding information base)

两种典型的分布式路由算法

基于网络距离的分布式路由算法--矢量距离法

基于信道状态的分布式路由算法--线(链)路状态法

距离矢量算法

初始化，各节点形成各自的本地信息－－即邻接路由器

扩散，各节点向邻居节点扩散已知的路由信息

计算，各节点根据邻居节点扩散来的信息计算新的路由

不断扩散，各节点定期不断向邻居扩散自己已知的路由信息

链路状态算法

以线路的延时作为链路度量

从每个节点探询相邻节点，得到延时(链路状态)初始值

每个节点定期和所有节点交换路由信息

根据收集到的路由信息，计算到其他节点的路径

出口选择算法

当 FIB 表项数很大时(如 ＞100 项)时，搜索时间变长，将严重影响分组中继性能

以太网交换机 FIB 算法

地址空间巨大、占用的 MAC 地址在空间中稀疏分布（利好哈希）

查表算法运算量分析

中继查表：表项数/2 次

Hash 算法：1 次

IP 路由器中的查表算法

不能采用 Hash 算法

手段 1：子网地址取代目标 IP 地址，压缩表项数

手段 2：缺省路由(default route)

手段 3：并行搜索

手段 4：优化 FIB 搜索算法（二分法，二叉树法，分级搜索法，高速缓存法……）

MPLS 交换机的 FIB 算法

每个标签对应一个出口，报文根据标签值从对应出口中继出去

小结

MPLS 处理效率最高

以太网交换机处理效率也很高

三者中，路由器处理效率最低，吞吐率最低

提高路由器中继效率-IP 交换技术

Cisco 提出 Tag-Switch，用标记的思想把寻址和转发统一起来，将路由过程和转发过程分离

通信业务的 QoS

QoS: 典型服务质量指标

传输延时、延时抖动、恒定流量、额定流量、最低流量

3.4 交换性能分析

Relay QoS

中继系统是一个多入多出的排队过程

QoS：各业务流在中继中的相互影响

中继排队模型

此处忽略 FIB 搜索过程

交换矩阵并行传递分组到多个输出口上

基本排队问题

输出排队(OQ, Output Queue)

分组直接进入交换矩阵，在输出端口排队

输入排队(IQ: Input Queue)

分组先进入队列，仅允许到不同输出端口的分组进入交换矩阵

IQ 和 OQ，该如何评价

IQ 和 OQ 性能分析

OQ：分组经交换矩阵直接到达输出口（性能好）

IQ：只能有一个分组进入交换矩阵。（矩阵简单）

Virtual Output Queues（VOQ）

既有 IQ 交换矩阵的简洁性，又有 OQ 的交换性能

VOQ 能趋于 OQ 的性能

连接如何共享？

什么是“正确”的速率分配？

很难一概而论

最大-最小公平原则

先满足最小的流，再将剩下的平分，直到分完

第四章 IP 组网原理

4.1 IP 编址技术

4.1.1 IPv4/v6 编址

（0）引言

网络地址的属性

身份属性（ID）：信息过滤、邻居管理

位置属性：寻址（路由）

IP 地址：通用标识符（唯一标识一台主机或一台路由器与因特网之间的一个网络连接）

一台主机可能分配多个 IP 地址

一台路由器通常分配多个 IP 地址

（1）IPv4 编址：地址结构与记法

Hierarchical address（分层地址）

包含位置属性、身份属性

更多的地址属性

功能属性、归属属性

网络编址 VS. 寻址

近处清楚，远处模糊

（2）IPv4 编址：分类/无分类编址

（2.1）IPv4 编址的演进

随着因特网应用爆炸式增长，IPv4 地址面临枯竭

（2.2）IPv4 地址分类

A 类：0+7 位 Net_id+24 位 Host_id

B 类：10+14 位 Net_id+16 位 Host_id

C 类：110+21 位 Net_id+8 位 Host_id

D 类：1110+28 位广播地址

E 类：1111+28 位（保留）

（2.3）分类编址：2 级编址

A、B、C 三类地址为固定数目地址块，造成极大的地址空间浪费

如何从 IP 地址中提取或计算相关信息

确定地址类别

提取 Net id 和 Host id（D、E 没有 Net id 和 Host id）

确定所在类别地址块包含的地址数量

确定所在类别地址块的首地址和末地址

首地址（网络地址）：Net id 不变，Host id 全 0

末地址（广播地址）：Net id 不变，Host id 全 1

网络地址、网络掩码、网络广播地址

网络地址：地址块的首地址

广播地址：地址块的末地址

网络掩码：Net id 全 1

（2.4）子网编址：3 级编址

划分子网（subnetting）理由：固定数目地址块存在浪费

基本思路：“化大为小”，将 A、B 地址块分为多个小地址块

Subnetting (子网划分)

给定的 IP 网分成一系列子网

Net id（定长）+Subnet id（变长）+Host id

Mask（掩码）

网络掩码：Net id 和 Subnet id 全 1

通过掩码可以确定：网络地址、网络规模

子网对 Internet 其他部分透明

外部仍将视一个组织划分子网后的网络为一个网络

采用分级路由：递交至网络——递交至子网——host

关键术语

网络号（Net id）：IP 地址的一个组成部分

网络地址（Net address）：一个 IP 地址（Host id 全 0）

子网号（Subnet id）：IP 地址的一个组成部分

子网地址（Subnet address）：一个 IP 地址（Host id 全 0）

主机号（Host id）：IP 地址的一个组成部分

主机地址（Host address）：一个 IP 地址

使用网络掩码计算网络地址

网络地址：掩码取 1 的位不变，取 0 的全 0

直接广播地址：掩码取 1 的位不变，取 0 的全 1

地址范围：网络地址~直接广播地址

超网

“聚小为大”：多个 C 类地址块聚合成一个大的地址块

Supernet id（变长）+Host id

子网 vs. 超网

子网：地址分配——一个物理网络

超网：地址申请（路由聚合）——一个组织（含多个物理网络）

（2.5）无分类编址：2 级编址

分类编址中的子网或超网并不能解决地址耗尽问题

可变长地址块

不属于任何类，每个地址块的地址数量必须是 2 的乘方

编址灵活，按需分配地址空间，有效利用地址

2 级编址：前缀（Net id）+后缀（Host id）

采用斜线记法，正式名称为 CIDR（classless interdomain routing，无分类域间路由选择）

a.b.c.d/n（n 为前缀长度）

网络前缀

具有相同网络前缀的地址可以用 1 个网络地址来表示——路由聚合

（2.6）IP 网络编址示例

全球管理机构 ICANN（因特网名字与号码指派公司）将大量地址块指派给 ISP（互联网服务提供商）

分配地址块时必须遵循的原则：

一个地址块的地址数 N 必须是 2 的乘方，$N = 2^{32-n}$，地址连续

地址块划分 IP 网络时必须遵循的原则:

每个 IP 网络的地址数 Ns 必须是 2 的乘方，地址连续

如何划分 IP 网络

定长划分：网络大小相同，较浪费

变长划分：网络大小不同，灵活、高效利用

变长子网掩码 VLSM（Variable-Length Subnet Mask）

IP 网络划分案例

每个物理网络所需的 IP 地址数量=本网络内主机数+本网络内路由器接口数+2（网络地址、网络广播地址）

每个子网的地址范围必须连续

IP 网络划分案例：分配 IP 地址

路由器的 IP 地址往往是网络内的前几个地址（不算网络地址）

分析：多接口设备

各接口可连接在同一个或多个不同的 IP 网络中（路由器各接口必须连接在不同的 IP 网络中）

每个接口需要一个 IP 地址

（3）IPv4 编址：特殊地址

网络地址：Net id 不全 0、不全 1，Host id 全 0，非源、非目的

直接广播地址：Net id 不全 0、不全 1，Host id 全 1，只能做目的

受限广播地址：255.255.255.255，只能做目的（本地广播，不走路由器）

全 0 地址：0.0.0.0，只能做源（DHCP 请求）

环回地址：127.0.0.1（127.0.0.1~127.255.255.254），只能做目的

私有地址：10.0.0./8, 172.16.0.0/12, 192.168.0.0/16，源、目的地址（不走路由器）

多播地址：224.0.0.0/4，只能做目的

直接广播地址

标识对特定网络的广播

受限广播地址

标识对本地网络的广播

全 0 地址

仅用于 DHCP 请求中

环回地址

测试本机上的软件

私有地址（Private address，专用地址）

不会在全球被识别：因特网中不可路由

用于隔绝其他网络，或通过 NAT 访问因特网

多播地址：D 类地址（224.0.0.0/4）

（4）IPv4 编址：移动 IP 编址

主机从一个网络移动到另一个网络时，面临两难

IP 保持不变，失去路由；更新 IP，失去连接

两个地址：归属地址、转交地址

归属地址 & 转交地址

归属地址（home address）：永久地址，关联归属网络

转交地址（care-of address）：临时地址，关联外地网络

（5）IPv6 编址：地址记法

十六进制冒号记法

也有 CIDR，表示多大的网

十六进制冒号记法的简写

忽略一个区（两个冒号间的 4 个数字）开头的零

零压缩：使用双冒号替代连续的几个零区（一个地址中只能使用一次双冒号）

（6）IPv6 编址：地址类型

单播地址：等同于 IPv4 单播地址

多播地址：等同于 IPv4 多播地址和 IPv4 广播地址（IPv6 认为广播是多播的一个特例）

任播地址（anycast address）：发送到任播地址的分组被交付给这个组中最容易到达的那个计算机

IPv6 地址空间分配

IPv4 兼容地址 0000::/80

未指明地址—— ::/128（等同于 0.0.0.0，不能做目的，只用于 DHCP 请求）

环回地址—— ::1/128（等同于 127.0.0.1）

嵌入的 IPv4 地址

兼容地址—— ::/96（不再使用）

映射地址—— ::FFFF/96

（如::FFFF:C000:201(或::FFFF:192.0.2.1)，表示 192.0.2.1）

全球单播地址 2000::/8

三级结构

n（48）位全球路由选择前缀+128-n-m（16）位子网标识+m（64）位接口标识
接口标识：可以嵌入长度小于等于 64 位的物理地址
以太网 MAC 地址（原 48 位）映射：反转第 7 位，在 24 位后插入 FFFE

用于本地编址的两个地址块

唯一场所单播地址
本地链路地址

本地链路地址 FE80::/10

功能：定义每个网络接口
在 IPv4 中，链路本地地址定义 169.254.0.0/16 地址块
应用：自动配置、邻居发现（相当于 ARP）

唯一场所单播地址 FC00::/7

等同于 IPv4 的私有地址

多播地址 FF00::/8

定义一组节点

（7）小结

4.1.2 NAT 技术

（1）Private Networks（私有网或专用网）

专门在一个组织内部使用
专用信道
不允许外人访问——保密性（Privacy）

Virtual Private Networks（VPN，虚拟专用网）

使用 Internet 进行机构的内部通信和外部通信
Tunneling（隧道）：一条建立在 Internet（公网）上的数据通道

隧道通信

原始报文加密，不被外部获知

举例：二层隧道

举例：三层隧道

专网地址

内部网专用地址：10.0.0.0/8，172.16.0.0/12，192.168.0.0/16
内部地址在公网上不会被路由

专网与公网的互联关系

专网地址无法在公网上使用，需要进行地址转换（NAT）

（2）NAT（Network Address Translation，网络地址转换）

私有地址（本地地址）←→全球地址（全局地址）
要求：网络仅有一条因特网的连接，且经由 NAT 路由器

NAT 类型

基于本地地址与全局地址的映射关系
基本 NAT（一个全局地址对应一个本地地址）、端口 NAT（NAPT）（一个全局地址对应多个本地地址）
基于 NAT 转换项的建立方式
静态 NAT/NAPT、动态 NAT/NAPT

（2.1）基本 NAT

所有经由 NAT 路由器的外出分组：去往因特网
所有经由 NAT 路由器的进入分组：进入内部网

基本 NAT 示例

（2.2）NAPT（端口 NAT）

问题：当同一个内网主机同时向不同外网发起通信时，NAT 如何处理？

Full Cone NAT
Address Restricted Cone NAT
Port Restricted Cone NAT
Symmetric NAT

NAPT 示例

NAT 应用－TCP 负载均衡

NAT 的限制

通信往往只能由内网主机（使用内网地址）主动发起
外网主机如何能够主动访问内网服务器？内网穿透（花生壳）

NAT 和 ISP

（3）小结

4.1.3 IP 地址与物理地址解析技术

（1）地址解析

已知目的 IP 地址，请求目的物理地址
互联网通信：网络层分组经由物理网络传输
网络层通信标识：IP 地址，全局唯一
物理网通信标识：物理地址，本地唯一

为什么需要地址映射

IP 分组交付到主机或路由器需要两级地址
互连网级：逻辑地址标识主机/路由器
物理网级：物理地址标识主机/路由器

地址解析实现方案

IP 对下接口规范化，增加 NI 软件模块处理具体的 AR

AR 方法

直接法，推导法，映射法

（2）ARP 协议

目标：动态掌握物理网内站点的(ip, ph)地址对应关系
原理：利用物理网具备的广播特性

设计 ARP 协议报文格式

主要功能：
通过询问的方式获取 IP 与 MAC 的绑定
支持多种类型的逻辑地址和物理地址的映射
两类操作：请求+响应

ARP 分组

需要预留请求解析的 IP 地址+MAC 地址，也要预留发送请求的 IP 地址+MAC 地址

用户输入命令“ftp 主机名”时的操作

ARP 实现时需要提高 IP 发送效率

ARP 为每个待解析地址设置待发 IP 分组队列

ARP 缓存表

维护最近请求的地址
提高 ARP 效率

ARP 请求分组示例

ARP 响应分组示例

ARP 软件结构

ARP 模块要实现的功能

Gratuitous ARP（免费 ARP）

IP 地址冲突检测（源和目的 IP 地址相同，如果收到了回应则说明 IP 地址冲突）

ARP 响应欺骗攻击

中间人攻击

ARP 请求欺骗攻击

利用免费 ARP

如何检测并防范 ARP 欺骗攻击？

设置静态 ARP 绑定，划分 VLAN，数据加密

（3）多播地址解析

IP 多播地址：D 类地址，每个多播地址定义一个多播组

IP 多播实现策略：利用物理网的多播（广播）机制

多播 MAC 地址

和单播 MAC 地址不一回事

映射 D 类 IP 地址的 MAC 多播地址

多播 IP 地址=1110+28 位多播 id

多播 MAC 地址=25 位前缀（01:00:5E:0）+多播 id 的后 23 位（也是 IP 的后 23 位）

问题：不明确的 MAC 地址

1 个 MAC 多播地址对应 32 个 IP 多播地址（前 4 位 1110，后 23 位确定，中间 5 位不确定）

（4）小结

4.1.4 主机地址配置技术

（1）引言

主机地址配置不仅仅只是配置 IP 地址

主机动态地址配置

至少配置三个参数：IP 地址、子网掩码、默认网关地址（访问外网的默认路径）

也会需要：DNS 服务器（解析域名）

服务器为主机动态配置 IP 地址等参数，使用 DHCP/DHCPv6 协议（UDP 通信）

设计 DHCP 协议流程

C/S 模型，允许存在多个应答者，Client 从中选择

分配的地址是有租期的， Client 可能需要续约

（2）DHCP 协议

协议模型

DHCP 通信：广播或单播

DHCP Client 用 UDP 广播 DHCP 请求

源 IP 全 0（还没有），目的 IP 全 1（广播），源 MAC 填客户端 MAC，目的 MAC 全 1（广播）

DHCP 服务器用 UDP 回答 DHCP 应答

源 IP 填服务器 IP，目的 IP 全 1（广播），源 MAC 填服务器 MAC，目的 MAC 填客户端 MAC（单播）

DHCP 通信：可靠性问题

可靠性由主机自己保证

DHCP 协议：分配 IP 地址

DHCP 协议：续租 IP 地址

DHCP 中继

如果客户端和服务器不在一个网里，路由器无法转发本地广播，解决办法：中继代理

中继如何工作？

前提：中继代理知道 DHCP 服务器的单播地址，在 67 端口监听广播信息

收到请求时：将请求封装在单播报文中，发送给 DHCP 服务器

属于应用层中继

优点：在多个子网上配置统一的 DHCP 服务器

安全问题示例：DHCP 服务欺骗攻击

服务器和客户端没有认证机制

DHCP 的安全问题

非法 DHCP 服务器

用户自设 IP 地址造成 IP 地址冲突

对 DHCP 服务器的拒绝服务（DoS）攻击

（3）IPv6 自动配置

主机为自己创建一个本地链路地址：前缀为 FE80::/64，后边接口 id 是 EUI-64 或以太网 MAC 地址映射

主机测试这个本地链路地址是否唯一：发送邻居询问报文并等待邻居通告报文

主机发送路由器询问报文并等待路由器通告报文

（4）小结

4.2 IP 分组传送技术

4.2.1 IP 分组递交与转发

（1）引言

因特网结构：内容（IP 分组），节点（路由器），信道（物理网络）

IP 分组投递方向

向上层递交，向下层交付

交付、转发、路由选择

路由选择（选路，Routing）：生成、维护路由表

转发（Forwarding）：查路由表，找到到达分组目的地的路径

交付（Delivery）：在网络层控制下，底层物理网络对分组的发送处理

逐跳交付和端到端交付

“跬步”与“千里”

路由表（RIB）和转发表（FIB）

RIB 由各种路由协议构建，包含达到同一目的地的所有路径

FIB 由 RIB 构建，只显示到达某个网络的最佳路由，如下一跳信息

物理地址（MAC）和逻辑地址（IP）

分组从源到目的节点，跨越多个物理网络

IP 地址始终不变，物理地址随物理网络发生变化

术语

路由表（Routing table，路由选择表）

Route：名词，路径

Routing：动词，选择路径的过程

Routed protocol：沿选定的路径交付分组，如：IP 协议

（2）IP 交付技术

直接交付：目的地址与发送接口在同一 IP 网络中（不走路由器）

间接交付：目的地址与发送接口在不同 IP 网络中（要走路由器）

交付操作

步骤：查路由表，找出下一跳物理地址，交给链路层

下一跳 IP 地址==目的 IP 地址——直接交付

下一跳 IP 地址≠目的 IP 地址——间接交付

下一跳（Next Hop）

IP 分组交付过程

一次交付过程==0 或多个间接交付+1 个直接交付（最后的交付）

分组：（源 IP 地址，目的 IP 地址）保持不变

帧：（源物理地址，目的物理地址）逐跳改变

主机间能进行 IP 通信的条件

同一 IP 网络中，或者不同 IP 网络有路由器连接

（3）IP 转发技术

无连接协议：基于 IP 分组的目的 IP 地址转发

面向连接协议：基于附加在 IP 分组上的标签转发

（3.1）基于目的地址的转发

使用尽可能少的信息实现转发

转发到目的网络（而非目的主机），记录下一跳（而非完整路径）

基于路由表的交付过程

逐条读取路由表项

如果目的网络（目的 IP&子网掩码）和本表项网络地址相等（匹配），则读取下一跳 IP 地址，ARP 为物理地址，再将其交付给数据链路层。如果不等则读取下一条表项。

如果遍历路由表也没匹配，则丢弃分组

例

路由表分析

特定网络路由：路由表的大小只与网络的个数有关

下一跳路由：路由器独立选路，只有最后一个路由器才知道目的主机是否存在

特定主机路由

掩码全 1

功能：检查路由，安全，管理

默认路由

不匹配特定网络路由的所有目的网络的转发路径

目的地址全 0，掩码全 0

功能：使路由表变得更小，隐藏大量的路由信息

默认路由示例

默认路由环路问题

Cisco 路由器的路由表实例

WinXP 主机的路由表实例

路由表与网络拓扑

路由表与网络拓扑的分析

无分类编址的转发

最长掩码匹配（longest mask matching）

直连路由——特定主机路由——特定网络路由——默认路由

前缀树

线性搜索效率低下

前缀树效率高，因为它是对数顺序

多级路由选择

路由聚合（Route aggregation）

聚合推论

聚合应用

合理的 IP 地址规划，隐藏网络结构

ISP 的分层路由选择

地理区域化路由选择（Geographical Routing）

（3.2）基于标签的转发

路由（Routing）涉及到搜索（searching）

交换（Switching）涉及到接入（accessing）

直接根据标签转发到相应接口，效率极高

MPLS（多协议标签交换，Multi-Protocol Label Switch）

MPLS 协议首部

标签堆栈

MPLS 网络架构

标签交换路由器 LSR（Label Switching Router）

边缘路由器 LER（Label Edge Router）：位于 MPLS 域边缘、连接其他网络的 LSR

核心 LSR（Core LSR）：区域内部的 LSR

IP 报文进入 MPLS 网络时，LER 给 IP 报文添加标签，所有 LSR 根据标签转发数据。当该 IP 报文离开 MPLS 网络时，标签由出口 LER 弹出

关键概念

FEC（Forwarding Equivalence Class，转发等价类）

LSP（Label Switched Path，标签交换路径）

如何建立 LSP

沿途 LSR 为特定 FEC 确定标签

MPLS 标签由下游分配，按照从下游到上游的方向进行分发

MPLS 转发过程

标签操作类型：Push，Swap，Pop

倒数第二跳弹出特性 PHP（Penultimate Hop Popping）

Label Stack 标签栈

类比：ATM（Asynchronous Transfer Mode 异步传输模式）中分级标签机制

引入标签：无连接→面向连接

标签表示一条流（flow），走相同的路径

标签的功能：加速路由过程

（4）IP 协议

（4.1）IP 协议特点

尽最大努力（Best effort）交付

无连接，不可靠，数据报

（4.2）IP 分组格式

Version field 版本号

4 bits，IP 版本检测

Header length field 头部长度

4 bits

固定长度 20 字节，可变长度最长 40 字节

以 4 字节（即 32bits）为单位，取值：5 ~ 15（首部长度 20~60 字节）

Service Type 服务类型

8 bits

以前定义为优先级（Precedence）位+服务类型（TOS）位

目前定义为 a set of differentiated services

ST 字段的演变

早期——该分组所期望的服务质量（作为路由器操作的提示，而非要求）

现在：Differentiated Services

6 bits Codepoint 定义了 64 种服务类型，2 bits 保留

Total length field 总长度

16 bits

包括首部，以 1 字节为单位，总长度＝首部长度（即 4×HLEN）+数据长度

为什么需要总长度字段？因为存在 Padding（填充），例如：以太帧数据部分要求最小 46 字节

Internet 中 IP 分组分片操作

关键问题：

如何区分多个分片来自同一个 IP 分组？

如何区分来自同一个 IP 分组的分片的顺序？

如何判断是最后一个分组？

Fragmentation Fields 分片字段

Identification（标识字段）：16 bits

源站每发送一个 IP 分组，标识+1。源 IP+标识可以唯一确定一个 IP 数据报

Flags（标志字段）：3 bits

1 bit 保留+1 bit D（Do not fragment） + 1 bit M（More fragments）

Fragmentation offset（片偏移字段）：13 bits

表示每个分片在整个数据报中相对位置，以 8 bytes 为单位

使用分片偏移来标识分段，不直接采用序号：有可能多次分段

Time To Live Field 生存时间

8 bits

分组寿命是受限的：防止路由成环时，IP 被无限次转发

分组允许经过的路由器最大数目（跳数）

Protocol Field 协议

8 bits

指明 IP 数据区的协议类型

指明 IP 分组应该交付到的高层协议

Checksum Field 校验和

16 bits

差错检测。只校验 IP 首部，数据部分由高层协议校验

每经过 1 个路由器，都需要重新计算 Checksum

Option Field 可选项

0 ~ 40 字节

很少使用

Options 选项

Types 类型

Record route 记录路由

Options Type=7

在 Option 区域中 Pointer 指向的字节，记录路由器的 IP

Pointer 原本为 4，每次记录后将 Pointer+4

Strict source route 严格源路由

Format 格式

作用

严格按给定的 IP 地址逐跳转发，中间不允许经过其它 IP 地址

IP 分组的 C 语言定义（Linux）

校验和算法

TCP/IP 协议使用最多的差错检测方法

IP 分组的以太帧封装

6 字节目的 MAC+6 字节源 MAC+2 字节 0x0800（表示封装的是 IP 分组）+4 字节 FCS（帧校验序列）

IP 分组示例

分片原因：物理网的 MTU（最大传输单元）限制

帧的数据长度不能超过 MTU

Fragmentation (分片)

在 MTU 较小的网络上，IP 将较长的数据报划分成更小的部分

分片的大小：

8 字节的整数倍（除最后 1 个分片）——保证偏移值是以 8 字节为单位

最接近网络的 MTU

Fragmentation Operation 分片操作

路由器和源主机可以进行分片

可以不止一次分片

每个分片都有 IP 头，具有相同 id，校验和重新计算

只在目的主机重组，因为每个分片都是独立的，可能经过不同路由器

Fragmentation Example 分片例

Reassembly 重组操作

只在目的主机重组

无连接，各分片的传输路径可不同

Reassembly Timer 重组时限，丢失分片则无法重组

目的主机能区分和重组不同的源 IP 报文（源 IP 地址，标识）

如何重组——重组表

功能：

找出一个分片属于哪一个原始的数据报

将属于同一个数据报的分片进行排序

（4.3）IP 协议操作

IP 协议软件模块

添加首部模块，处理模块，转发模块，分片模块，重组模块

Linux IP 协议处理流程

（5）IPv6 协议

（5.1）IPv6 协议概况

IPv6 的产生原因：IPv4 地址紧缺

IPv6 的目标：取代 IPv4

IPv6 的改进：地址空间，分组格式，资源分配，安全性……

IPv6 的使用情况

采用进度延缓

（5.2）IPv6 分组格式

基本首部和扩展首部

IPv6 报文头部

Next header 下一个首部：定义了基本首部后的扩展首部

Flow Label 流标签：可以面向连接

扩展首部

Authentication 认证：验证发送方，确保数据完整性

Source Routing 源路由

IPv4 和 IPv6 的分组格式对比

（5.3）IPv4 到 IPv6 的过渡技术

Dual stack 双协议栈，Tunneling 隧道技术，Header translation 首部转换（NAT-PT）

双协议栈

隧道技术

用于穿越 IPv4 的网络，通信两端是 IPv6

（IPv6 嵌套一层 IPv4）

自动隧道（Automatic tunneling）（使用兼容 IPv4 的 IPv6 地址）

配置隧道（Configured tunneling）

首部转换

在 IPv6 网络和 IPv4 网络之间设置 NAT-PT 网关

（6）小结：交付&转发

小结：IP & IPv6 协议

4.2.2 ICMP 与差错协调

（1）ICMP 协议

IP 协议：尽最大努力交付（缺少差错控制、缺少管理查询机制）

ICMP 协议：IP 的辅助协议

ICMP 报文封装

ICMP 逻辑上与 IP 同在网络层，但实现在 IP 之上

报文封装：Protocol = 1

ICMP 协议报文

差错报告报文，查询报文

ICMP 报文类型

差错报告报文

只能送给 IP 分组的源站，只提供差错报告

差错报告报文的数据字段

出错分组的 IP 首部和数据部分的前 8 字节（提供高层协议信息）

路由器转发产生的 ICMP 差错报文

IP 分组转发中出现了异常

重定向

A 连接 R1、R2，默认路由是 R1。

A 给 B 发送分组，B 在 R2 连接的网里。

R1 收到分组后，转发给 R2，并向 A 发送 ICMP 重定向报文

表示给 B 发送分组直接走 R2 更快。

目的主机产生的 ICMP 差错报文

主机的各协议实体处理出现异常时

排错工具

Ping（Packet InterNet Groper）测试网络上任意站点是否可达

Traceroute 跟踪一个分组从源到目的的路径

（2）ICMPv6 协议

ICMP、ARP、IGMP 合并到 ICMPv6

ICMPv6 差错报文

ICMP 重定向报文被 ICMPv6 邻居发现协议（NDP）取代

ICMPv6 信息报文

ICMPv6 邻居发现报文

Neighbor-Discovery (ND) protocol 邻居发现协议

Neighbor-Solicitation Message 邻居请求报文（类似 ARP 请求）

Neighbor-Advertisement Message 邻居通告报文（类似 ARP 响应）

Router-Solicitation Message 路由器请求报文

Router-advertisement Message 路由器通告报文

（3）小结

4.2.3 多播/广播 IP 分组转发

（1）单播、广播、多播通信

unicast 单播：目的地址=单播地址

broadcast 广播：

本地广播：目的地址=全 1 地址，网络广播：目的地址=网络广播地址

multicast 多播：目的地址=多播地址

Unicasting 单播 VS. Multicasting 多播

单播：只从一个接口转发出去

多播：可能从多个接口转发出去

多播通信 vs. 多个单播通信

多播通信：带宽占用少，负载小，延迟低，更高效

多播路由器如何路由

每个多播路由器都需要知道至少有一个多播用户的组所对应的接口

多播应用

Access to distributed database 访问分布式数据库

Information dissemination 信息传播

Dissemination of News 传播新闻

Teleconferencing 电视会议

Distance learning 远程学习

IP 协议收发多播分组的条件

加入指定的多播组

每一个多播地址代表一个多播组，IP 实体可同时加入多个多播组

（2）IGMP 协议

Internet Group Management Protocol 网际组管理协议

功能：帮助多播路由器创建和更新每个接口子网中的多播组成员列表

IGMP 协议设计

目标：路由器能够实时监控所在网络中存在哪些组播业务

基本功能：路由器如何知道有哪些组播业务，路由器如何实时监控

IGMP 基本工作原理

IGMP Operation 协议操作

Joining a group 加入组

Leaving a group 离开组

Monitoring membership 监控组成员

3 个版本

IGMP 协议操作

加入多播组操作：本地子网内

退出多播组操作：主机

（3）多播分组转发技术

跨子网的多播通信：路由器转发多播分组

特征：动态（成员动态变化），无方向性（存在转发环路）

多播分组转发的要求

每个组成员仅应收到多播分组的一个副本

非多播组成员不应收到多播分组的副本

解决方案：反向路径转发，多播树

反向路径转发技术

Reverse path forwarding（RPF）

使用源 IP 地址查找单播路由表

MBONE

Internet 中的多播骨干网（multicast backbone）

把多播封装在单播中

新型组播技术：BIER

Bit Index Explicit Replication 位索引显式复制

（4）小结

4.2.4 移动 IP 代理技术

（1）引言

移动 IP：主机在多个网络间自由移动且通信

问题：IP 地址专为固定主机设计

当主机从“家乡”子网移动到“外地”子网时：

如果改变主机的 IP 地址，则移动切换网络时通信中断，用户身份改变

如果 IP 地址不变，全网所有路由器都要特定主机路由，路由规模过大

移动 IP 的要求

除“家乡网络”和主机驻留的“外地网络”外，网络的其它部分与移动无关

“双地址”，目标对外不改变移动主机的 IP 地址

归属地址（home address）+转交地址（care-of address）

不在无关路由器上设特定主机路由

（2）移动 IP 代理技术

移动主机的接收：

归属代理截获所有到移动主机的 IP 报文，并转送给外地代理

外地代理将归属代理送来的 IP 报文转送到移动主机

移动主机发送不受影响：路由器按 IP 报文目的地址寻址

移动通信的三个阶段：发现、登记、传输

移动 IP 通信模型：FA 转交

移动 IP 通信模型：同址转交

（3）移动 IP 通信分析

Proxy ARP（代理 ARP）

功能：产生子网效应

路由器代表另一个物理网络中一组主机回答 ARP Request，在 ARP 响应中通告自己的 MAC 地址

让 ARP 请求的发送者误以为路由器是目的主机，但目的主机实际上在路由器“另一边”

方案：路由器上运行 Proxy ARP 软件

代理 ARP、免费 ARP

移动主机在外地网络中并注册成功：

归属代理：为该移动主机代理 ARP

归属代理：执行免费 ARP（更新 ARP 缓存，防止向移动主机的 MAC 发送信息）

移动主机回到归属网络后：

归属代理：停止为该移动主机代理 ARP

移动节点、归属代理：执行免费 ARP（防止向代理的 MAC 发送信息）

二次穿越问题

移动主机与远程主机在同一个 IP 网络中

三角路由问题

移动主机和远程主机在不同的 IP 网络中

解决低效率问题

让远程主机绑定移动主机的归属地址和转交地址

移动主机再次移动时，归属代理向远程主机发送告警分组以通知绑定的改变

（4）小结

4.3 端到端传送与网络拥塞控制技术

4.3.1 传输层技术概述

（1）传输层功能

传输层：应用层与网络层之间的桥梁

两种服务：Message Delivery 数据报 VS Stream Delivery 字节流

Message Delivery service (UDP)独立处理每一个数据报

Stream Delivery service (TCP)在“管道”中传递字节流

（2）进程到进程的通信

IP 协议是主机间通信，传输层协议是进程间（端口间）通信

IP 地址 vs. 端口号

在 C/S（客户端/服务器）模型中，与端口号相关的问题

问题 1：如何为服务器分配端口号

Well-known ports 熟知端口 VS. Ephemeral ports 临时端口

Well-known ports 熟知端口

Registered ports 注册端口

Dynamic ports 动态端口

问题 2：如何实现并行服务

方案 1：Iterative server 迭代服务器（用户排队，等待服务器轮流服务）

方案 2：concurrent server 并发服务器（服务器子进程在临时端口为不同用户提供服务）

socket pair（套接字对）源 IP、源端口、目的 IP、目的端口四个元素唯一确定一个连接

复用和分用

（3）控制机制

Flow control 流量控制，Error control 差错控制，Congestion control 拥塞控制

IP 不可靠，需要依靠 TCP 确保可靠

Window based control Protocol 基本协议技术（基于窗口的控制）

序号：TCP 的解决方案 Numbering Bytes 给字节编号

给每个字节编号，而不是给报文段编号

每个连接的编号是独立的

初始编号是随机的，不是从 0 开始

报文段的序号=报文段中第一个字节的字节号

确认：Acknowledgment，ACK

对已经收到的字节表示确认

Positive ACK 肯定确认+Cumulative ACK 累计确认

收到一个 ACK，表示此编号前的字节全部正确收到（希望收到该序号的字节）

超时重传机制

每发送一个报文段就启动定时器，如果超时未收到 ACK 则重传

发送方必须缓存已经发送但未收到确认的报文段

报文段损坏或丢失：超时重传

ACK 丢失：收到 ACK 时可以确认该 ACK 以上的报文段（例如 ACK1601 丢失，收到 ACK1801 时也能对 SEQ1601 进行确认）

滑动窗口

循环滑动

停等协议（Stop-and-Wait Protocol）

接收方向发送方反馈每一帧的接收信息

接收方将收到的数据交付给网络层后，向发送方发送 ACK，表明允许发送方发送下一帧

发送方每发完 1 帧，就停止发送，等待 ACK 到达；当接收到 ACK，再继续发送

停等协议的性能

信道的利用率低

协议改进——提高信道利用率

允许发送方在等待应答之前发送多个数据帧

滑动窗口：发送窗口与接收窗口

sending window 发送窗口：已发送但还未收到确认，或可以立即发送

receiving window 接收窗口：收方希望接收的字节号

发送窗口——滑动且大小变化

min（rwnd 接收窗口，cwnd 拥塞窗口）

滑动（sliding）——随时改变滑动窗口的大小

（3.1）流量控制

平衡生产者产生数据和消费者消耗数据的速度

生产者（发送端的应用层）推送 Pushing 到传输层

消费者（接收端的应用层）拉取 Pulling 传输层的报文

推送过快或过慢时进行流量控制

Expanding the sender window（发送窗口的扩展）

消费速度大于生产速度，推送过慢

增大接收窗口，扩展发送窗口

Shrinking the sender window（发送窗口的缩小）

消费速度小于生产速度，推送过快

减小接收窗口，缩小发送窗口

Closing the sender window（发送窗口的关闭）

接收方缓存满了，接收窗口置 0，关闭发送窗口

（3.2）差错控制

Reliability 可靠性

Sequential 按序, without error 无错, and without any part lost 不丢 or duplicated 不重

如何实现可靠传输？——ARQ（Automatic Repeat reQuest，自动重传请求）

解决方案：序号，确认号，超时重传

滑动窗口：差错控制 + 流量控制

收方：反馈确认号

发方：发送窗口滑动（左端前移）

（3.3）Congestion Control（拥塞控制）

Congestion 拥塞：网络中，路由器接收过多的分组，超过其处理能力时，发生拥塞

拥塞引起的重传，会使情况更糟

网络拥塞的主要原因：

到达速率>处理速率输入→队列长度

离开速率<处理速率输出→队列长度

拥塞影响：延时大，吞吐量小

拥塞控制策略

开环拥塞控制：拥塞发生前的预防措施

闭环拥塞控制：拥塞发生后的缓解措施

（4）连接机制

无连接服务：无法有效实施流量控制、差错控制或拥塞控制

面向连接服务：能实施流量控制、差错控制或拥塞控制

三个阶段：连接建立、数据传输、连接拆除

（5）小结

4.3.2 UDP 协议

（1）UDP 协议概述

“非连续性”通信：DU 间基本没有前后关联性、DU 丢失对应用影响不大

对 IP 协议的增强：提供进程到进程的通信

UDP

User Datagram Protocol 用户数据报协议

UDP 的应用

简单、简短的应用

一般使用方式——客户/服务器模式（C/S 模型）

服务端固定守候在特定 port x，提供服务

（2）UDP 协议报文

Protocol = 17

UDP 首部：8 字节（64 位）

2 字节源端口+2 字节目的端口+2 字节总长度+2 字节校验和

UDP 校验和 checksum

可以选择不校验：减少传输开销

校验三个部分：Pseudo header + UDP header + Data

Pseudo header 伪首部：发送端不传输，接收端根据 IP 首部的部分信息形成

伪首部格式

12 字节（96 位）

4 字节源 IP+4 字节目的 IP+1 字节填充 0+1 字节协议号（17=00010001）+2 字节 UDP 总长度

（3）UDP 协议操作

关于 UDP 输出模块的讨论

UDP 输出模块：封装 UDP 报头

计算校验和的困难：起初不知道源 IP。

1.先把 UDP 传到 IP

2.IP 层得到源 IP，送回 UDP

3.UDP 计算出校验和，封装完成后再送到 IP

UDP 的服务特性

无目标，无效率，无结果，盲接收，新块接收

（4）小结

4.3.3 TCP 协议

（1）TCP 协议概述

高层应用的需求：reliability 可靠性

底层网络和 IP 网络是不可靠、无连接投递

TCP：进程到进程的通信，面向连接、可靠的字节流通信

TCP

Transmission Control Protocol 传输控制协议

TCP 面临的问题

丢失，延迟，乱序

超时重传机制，自适应加速机制，接收窗口机制

字节流传送

TCP 为所有数据字节编号

缓冲区和报文段

应用程序：使用自己认为适宜的任何大小的数据片进行发送或接收（最少 1 字节）

TCP：根据网络情况选择适当的发送缓冲区（分割）或接收缓冲区（合并）

（2）TCP 协议报文

TCP 首部：基本首部 20 字节+可选首部 0~40 字节

2 字节源端口+2 字节目的端口

+4 字节 SEQ 号

+4 字节 ACK 号

+4 bits 首部长度+6 bits 保留+6 bits 控制字段+2 字节窗口大小

+2 字节校验和+2 字节 Urgent pointer 紧急指针

+可选 0~40 字节

初始序号（ISN）

在建立连接时，通信双方的 TCP 各自随机产生初始序号

Control Field 控制字段

用于：流控，连接建立拆除，TCP 传输模式

URG，RST，ACK，SYN，PSH，FIN

校验和

和 UDP 计算方式相同，但不需要回送（面向连接，知道 IP 地址，直接算校验和）

紧急指针与紧急（带外）数据

紧急指针：只有当控制域中的 URG 置位时，才有效。指向带外数据的最后一个字节

（也就是说带外数据是从数据开头到紧急指针指向的位置）

紧急数据：不需要接收进程按序读取，不在数据流中排队，直接递交上层

序号、确认号

序号：报文段中第一个数据字节的序号。初始序号是随机值

确认号：期望接收的下一个数据字节的编号（之前都收到了）。

确认号是累积值

TCP 选项

最大报文段长度（MSS）选项

发送该选项的 TCP 端能接收的报文段的最大数据长度

（3）TCP 协议操作

（3.1）TCP 连接

虚连接：(源 IP,源端口)——(目的 IP,目的端口)

Discussion

Finite State Machine（有限状态机）

TCP 的状态转换图

TCP 连接建立：3 次握手

1:—SYN→，2:←SYN+ACK—，3:—ACK→

同时打开

TCP 中的 SYN Flooding （SYN DoS）Attack（SYN 洪泛攻击）

攻击者伪造源 IP，向服务器发送大量 SYN 请求，但不回复 ACK，导致服务器缓存耗尽

TCP 数据传输

推送数据 & 紧急数据

推送数据（pushing data）：强调信息的实时性（如视频通话）
发送方 TCP：设置 PSH 标志位，不需要等待窗口填满，立即发送报文段
接收端 TCP：尽快交付给接收进程，不必等待更多的数据到来
紧急数据（urgent data）：强调信息的突发性（如警报）
发送方 TCP：设置 URG 标志位，将紧急数据插入报文段数据起始处
接收方 TCP：收到时将紧急数据抽出来，立刻通知应用程序

Urgent Data（紧急数据，带外数据）

目的：使数据在接收方 TCP 无需缓存，立即交给接收进程处理

连接拆除（Connection Termination）

通信是双向的，但连接终止是单向进行的
连接拆除的发起方只能关闭自己的发送方向，而 TCP 的另一端可以继续发送数据——半关闭
不能发送数据，可接收数据，可发送确认
接收方向由对方拆除

TCP 连接终止：4 次握手

1:—FIN→，2:←ACK—，3:←FIN—，4:—ACK→
1+2:A→B 的半拆除。3+4:B→A 的半拆除。

TCP 连接复位

发送 RST 报文段，无需确认

（3.2）TCP 流量控制

TCP 中的数据流和流量控制反馈
传输层：接收端向发送端反馈
接收端：传输层向应用层反馈

发送缓冲区 & 发送窗口

发送缓冲区：等待进程写入的空闲部分
发送窗口：已经发出未被确认的部分，以及可以立刻发出的部分

接收缓冲区 & 接收窗口

接收缓冲区：按序存放 TCP 报文段数据
接收窗口：接收缓冲区的空闲部分

发送窗口大小

=接收窗口
接收快，扩展发送窗口，加快发送
接收慢，收缩发送窗口，减慢发送
接收缓冲区慢，接收窗口置 0，关闭发送窗口，停止发送

窗口管理

流量控制效果

接收方掌握流量控制的话语权
流控机制不会立即见效

糊涂窗口综合症

首部长度占比过大，数据很少，效率降低
解决方案：发送方延迟发送，接收方 0 窗口确认，或推迟确认

（3.3）TCP 差错控制

TCP 的差错：受损，丢失，失序，重复
差错控制工具：校验和，确认，动态超时重传

确认

肯定累积确认：期望接收的下一个字节序号
选择确认：报告失序、重复报文段（TCP 选项）
何时发送：
有数据发送时：捎带确认
无数据发送时：立即或推迟确认

动态超时重传

重传的超时时间：应大于并接近往返传输延迟 RTT
TCP 应对策略：动态重传超时定时

（3.4）TCP 拥塞控制

拥塞问题：传输路径中的“瓶颈”
拥塞对于端点而言，表现为：TCP 传输延迟增加，导致超时重传
TCP assumes（关键假设）：丢包就是因为发生了拥塞
发现拥塞就控制发送速率

Congestion Window（拥塞窗口）

发送窗口=min{接收窗口，拥塞窗口}
接收窗口 rwnd：流量控制
拥塞窗口 cwnd：拥塞控制

Congestion Control Algorithms 拥塞控制算法

Slow start（慢启动）
Fast retransmit/Fast recovery（快速重传和快速恢复）

Slow start（慢启动）

建立一条新的 TCP 连接以后，执行慢启动
在 TCP 连接建立初期试探网络的拥塞状况

Congestion avoidance（拥塞避免）

当发送窗口大小＝ssthresh 时，发送方进入拥塞避免阶段
只有当窗口中所有报文都确认后，拥塞窗口＋1

拥塞检测

推测拥塞：超时重传
1.ssthresh（慢启动阈值）设置为当前 cwnd（拥塞窗口）的一半
2.cwnd（拥塞窗口）= 1
3.重新执行慢启动

快速重传和快速恢复

如果不是超时，而是连续收到 3 个 ACK：
说明 1 个报文段可能丢失了，但后面的几个报文段又安全到达（有点拥塞但不多）
1.ssthresh（慢启动阈值）设置为当前 cwnd（拥塞窗口）的一半
2.cwnd＝当前阈值
3.开始拥塞避免阶段（线性增加）

TCP 拥塞策略小结

拥塞举例

TCP 协议性能

能适应各种性能的网络
能适应时延长短不同及变化范围大的传输路径

应用适应性

TCP 协议的缺陷

拥塞控制，连接管理，传输时延

（4）SCTP 协议

（4.1）SCTP 协议概述

Stream Control Transmission Protocol 流控制传输协议

面向连接（关联）、可靠、报文流

简要对比：UDP、TCP、SCTP

UDP：面向 datagram 报文，无连接，不可靠

TCP：面向 segment 字节，面向 connection 连接（一条流），可靠

SCTP：面向报文（数据块、控制块，packet），面向 association 关联（多重流、多重归属），可靠

多重流 Multiple Stream

Partially ordered 部分有序

多重归属 Multihoming

冗余备份，而非负载均衡

SCTP 分组

首部+Control chunk 控制块+Data chunk 数据块（可以有多个，可以属于不同流）

报文流

TCP：字节流，字节编号，序号

SCTP：报文流，数据块编号

SCTP 分组、数据块、流

进程 A 与进程 B 间并行传输多条流

（2）SCTP 协议报文

通用格式：首部+控制块+数据块（控制块在数据块之前）

通用首部：8 字节

2 字节源端口+2 字节目的端口+4 字节验证标签（安全）+4 字节校验和（校验整个分组）

块（Chuck）

通用格式：8 bits 类型+8 bits 标识+16 bits 总长度+块信息（4 字节的倍数）

数据块：8 bits 标识=5 bits 保留+U（数据交付方式）+B（是开头）+E（是结尾）

控制块（Control Chunk）

建立关联、终止关联、探测关联的状态、流量控制、差错控制、拥塞控制

INIT 块、INIT ACK 块

参数的方向性

变长参数

SACK 块

流量控制、差错控制

（3）SCTP 操作

（3.1）SCTP 关联

状态转换图

SCTP 建立关联：四次握手——对抗 SYN DoS attacks

1.—INIT→

2.←INIT ACK—（携带 Cookie）

3.—COOKIE ECHO→

4.←COOKIE ACK—

Verification Tag 认证标签（使用 VT 标识一条连接，而不是 IP+端口号）

Cookie 缓存（收到 COOKIE ECHO 之前不分配资源，防范 DoS 攻击）

“Cookie”

技术类比：CHAP (Challenge Authentication Protocol，质询认证协议)

口令认证面对的问题：明文传输

一般情况

SCTP 关联操作 vs. TCP 连接操作

建立：四次握手 vs. 三次握手

终止：三次握手 vs. 四次握手

异常中止：ABORT vs. RST

数据传输

SCTP：只有 DATA 块才消耗 TSN

TCP：DATA、SYN 比特、FIN 比特都要消耗序号；ACK、RST、窗口更新不消耗序号

（3.2）SCTP 流量控制

SACK 控制块：累积 TSN 确认，接收窗口通告

接收方：1 buffer 接收方缓冲，3 variables（cumTSN/winSize/lastACK）

发送方：1 buffer 发送方缓冲，3 variables（curTSN/rwnd/inTransit）

SCTP 流控示例

（3.3）SCTP 差错控制

受损数据块：ERROR 控制块

丢失数据块：重传

失序、重复数据块：SACK 控制块

协议技术：“指定起始”

（3.4）SCTP 拥塞控制

与 TCP 相同的拥塞控制策略：慢启动、拥塞避免、拥塞检测、快重传/快恢复

与 TCP 的区别：多归属，显式拥塞通知

（4）小结

第五章 IP 路由技术

5.1 IP 路由问题

5.1.1 Forwarding vs. Routing

Forwarding 选路：data plane 数据面，本地

Routing 计算路由表：control plane 控制面，全局

时间尺度上：Forwarding：纳秒级，Routing：（希望做到）数十毫秒级

分类

静态：非自适应

动态：自适应

先应式 vs. 反应式

分布式 vs. 集中式

Routing

Destination-Based Routing 基于目的地址

Flow-Based Routing 基于流

Policy-Based Routing 基于策略

Destination-Based 基于目的地址 vs. Flow-Based 基于流

不同源到相同目的地址

基于目的地址：路径如果出现交叠，那么后面的路径一定相同

基于流：路径可能差异很大

Convergence（收敛）

网络中所有路由器都达成一致的拓扑结构

Convergence time（收敛时间）：从不一致到一致所经历的时间

影响因素：路由协议、距离、路由器数量、带宽负载、路由器负载、流量模式

Metric（度量）

用来衡量通过某一网络所需的代价

Hop count 跳数, bandwidth 带宽, delay 延迟, MTU, load 负载, reliability 可靠性, …

Routing protocol 使用 Metric 来选择一条 best path 最优路 for routing

5.1.2 Internet 路由体系

由“自治系统（Autonomous System，AS）”构成的多级网络，也称作 domain（域，或自治域）

Autonomous system

一个管理员管理的一组网络和路由器

引入 AS 之后，路由分为

域内路由 intra-domain routing：内部网关协议 IGP，主要考虑度量 metric

域外路由 inter-domain routing：外部网关协议 EGP，主要考虑策略 policy

IGP VS. EGP

IGP: matric-based 基于度量

EGP: policy-based 基于策略

三级路由体系

单个网络内部：到达每台主机（L2）

Intra-domain：AS 内部（L3，域内路由）

Inter-domain：AS 之间（L3，域间路由）

网络的多样性

ISP（服务提供商），企业网，Data Center（数据中心）

Data Center（数据中心）

5.1.3 Internet 路由协议

IGP：AS 内路由器交换拓扑信息，形成 AS 内的最短路径

EGP：AS 边界路由器交换内部网络可达性信息，形成合理的 AS 路径

分布式路由协议的基本要素

彼此共享互连网络信息，独立计算路径，建立/更新路由表

（1）IGP

AS 内部的路由协议

距离：度量（Metric，整数值）

最短路径定理

若从 S 到 D 的一条最短路径经过了中间节点 K，则从 K 沿这条路径到 D，是 K 到 D 的最短路径之一

正确的路由

没有“环路”：难

正确的路由状态：形成以目的节点为根的“生成树”

例

典型的 IGP 协议

RIP：距离矢量 DV 路由算法

OSPF：链路状态 LS 路由算法

（2）EGP

AS 间的路由协议

行政管理对域间路由的修正

AS 可以自由选择自己的路由策略

AS 间商业关系对拓扑和策略的修正

AS 间的连接仅当存在商业关系时出现

三种基本关系：

AS-A 为 AS-B 的用户：向 B 付费

AS-A 为 AS-B 的服务提供者：收取 B 的付费

AS-A 和 AS-B 对等：差额结算

为什么需要对等

域间路由上流动的是钱

AS 间拓扑关系对应了 AS 间的商业关系

AS 间的商业关系决定了路由是如何通告的

关键：路由宣告

典型的路由宣告策略

Gao-Rexford 规则

从下级来的消息，告诉上级、平级、别的下级

从同级或上级来的消息，只告诉下级

Gao-Rexford 规则形成的关系图是一个有向无环图（DAG ，directed acyclic graph)

AS 类型

残桩 AS（stub AS）（只能到达另一个 AS）

多归属 AS（multi-homed AS）（有多条连接到达其他多个 AS，但不转发）

转接 AS（transit AS）（允许数据流量转接的多归属 AS）

5.1.4 小结

5.2 基本路由算法与协议

5.2.1 距离向量路由算法和 RIP

Distance vector routing（DV）

使用图论中的 Bellman-Ford 算法

RIP

Routing Information Protocol 路由信息协议

实现在应用层，使用 UDP 封装，控制网络层的路由表

internet 网络→Graph 图

图：一组节点和边

在网络中，路由器用点标准，网络用连接两点的边表示

距离向量（Distance Vector，DV）

DV：一个有序对(v,d)。v：目的网络，d：到该目的网络的距离

从哪获得 DV？邻居节点周期性的路由通告报文中

Bellman-Ford Algorithm

找到任意两点间的最短路

Distance Vector（距离向量）Routing

基本原理：每一个路由器周期性地与其邻居路由器分享关于整个网络的路由信息

分享：整个路由表，周期间隔，只和邻居

计算路由表：Bellman-Ford 或 Ford-Fulkerson

RIP V1 Message 报文

Format(<=512 bytes)

封装在 UDP（端口号 520）

采用广播的方式通告路由

RIPv2 报文

格式

Types of Messages 报文类型

请求（command = 1）广播，希望得到路由表

响应（command = 2）询问响应：响应 Request 分组（单播），非询问响应：周期更新（广播）

Example of RIP Message 例

RIP Operation 操作

Discovery 发现（直到收敛）

Topology change 拓扑改变（类似发现：直到收敛）

Calculating 计算路由表

Example：Initial routing tables in a small autonomous system 例：小自治系统的初始化路由表

Sending RIP Responses 发送响应

三种情况：收到请求；定时到了；路由变了

Updating the Routing Table 更新路由表

收到 RIP 响应时

Network Discovery 网络发现

路由表：目的网络、下一跳、发送接口、Metric

Topology Change 拓扑改变

Timers in RIP 定时器

Periodic 30s 定期发送路由表

Expiration 180s 超时则将跳数设置为 16（先不删除，把不可达信息扩散出去）

Garbage collection 120s 删除该路由

Problems with RIP

Count to Infinity——Slow convergence 缓慢收敛

Instability 不稳定性

Slow Convergence 缓慢收敛

每次扩散平均都要 15s

解决方案：把跳数限制为 15，16 表示不可达

Routing Loop 路由环路

可达信息不及时消失

Some Remedies for Instability 不稳定性的解决方案

Triggered update 触发更新

Split horizons 分割范围

Poison reverse 毒性逆转，属于分割范围的变体

Hold-down timer 抑制定时器

都不能保证 100%有效

Triggered Update 触发更新

试图加速收敛。如果网络改变，立刻发送消息

特殊处理：限制触发频率（等待 1~5s），限制触发的路由器（只有直接影响的路由器）

Split horizons 分割范围

通过某个接口获得的路由信息，不再从这个接口回送过去

Poison Reverse 毒性逆转

对于那些从某个接口获得的路由表项在通过同样的接口更新扩散时，要将其 Metric 值设为 16

多节点环路

解决措施：抑制定时器（hold-down timer）

当某条路由被检测为不可达，该路由会进入抑制状态。

该路由将不会被更新或接受新的路由信息，直到抑制定时器超时。

在抑制期间，路由器都不会轻信任何更新，从而避免因为旧的或不正确的信息引发环路。

RIP 的应用

适用于中小规模的网络

好消息传的快，坏消息传的慢

5.2.2 链路状态路由算法和 OSPF

Link State routing（LS）

使用 Dijkstra 算法

Dijkstra Algorithm

把当前点当做根，建立一棵最短路树

把节点分为两组：临时性节点+永久性节点

Dijkstra 算法例

基于 SPF 树

Link State Routing

每个路由器都有网络的完整拓扑

OSPF 协议

Open Shortest Path First 开放最短路径优先

实现在应用层，不经过传输层，直接封装在 IP（协议号 89），控制路由表

获取网络拓扑

分享链路状态，洪泛给每个路由器，当发生变化时（事件驱动）

什么是链路状态路由协议

链路状态：直接连接了哪些网络+通过链路连接了哪些路由器

路由器需要建立每台路由器的所有链路状态

执行 SPF，得到以该路由器为根，由它出发到其他各个网络的路径为树枝的一棵树

路由器间的邻居关系

由连接它们的网络类型决定

点到点网络：只连接两台路由器

转接网络：多台路由器连在一个以太网上

OSPF 网络类型

点到点网络、转接网络 Transit network

转接网络分为广播多路访问网络、非广播型多路访问（NBMA）网络

OSPF Link Types 连接类型

1.点到点链路：连接两个路由器，邻居就是邻居路由器

2.转接链路：每个路由器可能有多个邻居节点（邻居——指定路由器）

连接：Transit network 存在的问题

N 个路由器组成的转接网络，每个路由器都有（N－1）个邻居路由器

将转接网络本身也抽象成一个路由器

具体方案：选择一个指定路由器，来充当转接网络抽象成的路由器

在 Transit network 中存在：DR(指定路由器)和 BDR(备份指定路由器)

在转接网络中，所有一般路由器只与该网络中的 DR/BDR 建立邻居关系

目的：减少 Transit network 的路由开销

OSPF Link Types 连接类型

3.残桩链路（Link to stub network）邻居——网络（将网络抽象成 1 个节点）

4.虚链路（Virtual link）（并未直接相连，而是经过多个路由器）

关键：能够根据链路状态信息形成拓扑图

举例

OSPF 报文

格式

多播发送：224.0.0.5，224.0.0.6

封装在 IP 分组中：协议号 = 89

OSPF 报文类型

Hello 周期发送，建立、维护双向邻居关系

DBD（Database Description），数据库描述

LSR（Link-State Request），链路状态请求

LSU（Link-State Update），链路状态更新

LSAck（Link-State Acknowledgement），链路状态确认

OSPF 操作

确定自己的邻居关系（双向）

向全体路由器通告自己与邻居的链路状态（可靠的洪泛）

用收到的链路状态独立计算路由（Dijkstra、SPF 树）

OSPF：邻居关系维护

Hello 机制：间隔 10s，失效间隔 40s

双向邻居：邻居发来的 Hello 报文的邻居列表中有自己

对比：RIP 的周期性路由信息报文

OSPF：链路状态通告

初始时，每个路由器仅知道邻居

将自己与邻居的链路状态信息向全体路由器通告，洪泛出去

获得全网所有路由器与邻居的链路状态

链路状态数据库（LSDB）初始化：本地网络拓扑结构

链路状态请求：特定链路的状态信息

链路状态全网通告：可靠洪泛

OSPF 的扩散与 RIP 不同，效果不同

RIP：更新后再扩散，产生新错误

OSPF：原样扩散，没有新错误

OSPF：可靠洪泛

必须对收到的新 LSA（链路状态确认）进行确认

重复发送 LSA 直到被确认

广播型转接网络上的 LSA 洪泛

1.DROther（非 DR）只向 DR 和 BDR 发送 LSA

2.DR 将该 LSA 洪泛给所有 DROther

3.所有路由器在它们的其它所有接口上洪泛该 LSA

多区域 OSPF

OSPF 的不利影响：链路状态数据库消耗路由器内存，SPF 算法复杂消耗路由器 CPU，洪泛扩散消耗网络带宽

划分区域（area）：构成一个互联网的路由器的一个子集

减少网络开销、节省路由器资源

OSPF 区域类型

区域的类型：决定了该区域内路由器所能接收的路由信息的类型

标准区域：区域内路由器，能够接收链路状态更新和路由归纳（区间路由）

主干区域（Backbone Area，Area 0）：负责标准区域，同时它还需要负责互连其它所有区域。其他区域不能直接互联

残桩区域（Stub Area）：不接收那些自治系统以外的路由信息（≠残桩网络）

分段区域（Partitioned Area）：因链路失效而将一个区域的一部分与其它部分隔离开来

孤立区域（Isolated Area）：没有链路与互联网相连

OSPF 路由器类型

Internal router, IR 区域内部路由器

Backbone router, BR 骨干路由器（所有属于 Area 0 (backbone) 的路由器）

Area Border Router, ABR 区域边界路由器

AS Border Router, ASBR 自治系统边界路由器

LSA 类型

区域内

Type 1：router-LSA，路由器链路 LSA

Type 2：network-LSA，网络链路 LSA

区域间

Type 3：summary-LSA，汇总链路到网络 LSA

Type 4：ASBR-summary-LSA，汇总链路到 ASBR LSA

外部

Type 5：AS-external-LSA，外部链路 LSA

LSA general header

Link state type，Link state ID，Advertising router

区域内的 LSA

Type 1：所有路由器向所在区域内宣告它的链路信息

Type 2：指定路由器 DR 向其所在转接网络宣告这个转接网络上的所有路由器

区域间的 LSA

Type 3：区域边界路由器 ABR 向一个区域宣告它连着的其他区域的网络

Type 4：区域边界路由器 ABR 向非主干区域（Area 0 以外非残桩区域）宣告自治系统边界路由器 ASBR

外部 LSA

Type 5：自治系统边界路由器 ASBR 向 AS 内所有非残桩区域宣告 AS 外的网络

举例

OSPF 的应用

单个区域可支持到 50 台路由器，适用于大中规模的网络

5.2.3 路径向量路由算法和 BGP

路径矢量路由算法 Path vector routing：受 DV 的启发

Reachability 可达性

初始路由表：AS 内每个网络的可达性信息

交换 PV 之后，路由器的路由表

路由聚合之后

与 DV 的重要差异

非“最短路径”选择、需要某种手段来避免环路、选择性的路由宣告

基于策略的路由宣告和选择

“我”是否愿意为其提供转发那些发往该路径分组

典型的路由选择策略

优先挣钱，性能最大化，最小化自己的带宽使用

BGP 协议

Border Gateway Protocol 边界网关协议

实现在应用层，封装在 TCP（端口号 179），控制 IP 的路由表

BGP 报文

BGP 报文首部

封装在 TCP（端口号 179）：单播、可靠

BGP 报文类型

Type 1: OPEN，打开报文（建立 peer 邻站关系）

Type 2: UPDATE，更新报文（宣告可达性或撤回路由）

Type 3: NOTIFICATION，通知报文（通知错误或关闭连接）

Type 4: KEEPALIVE，保活报文（互相保持邻站关系）

BGP 路由交互操作

建立 BGP 邻站 peer 关系（BGP 会话）

交换 BGP 路由信息（可达、不可达）

BGP 会话

EBGP 会话：External BGP session（AS 外）

IBGP 会话：Internal BGP session（AS 内交换路由信息，不一定完全等于物理连接）

BGP 路由处理操作

BGP 路径属性

用来帮助 BGP 协议选路

重要的属性：AS_PATH，NEXT_HOP

NEXT_HOP 属性示例

到达某个网的下一跳路由器

LOCAL_PREFERENCE 属性示例

到达某个网有多个路由器，选择最喜欢（值最大）的那个

MED（Multi-Exit Discriminators）属性示例

当一个 BGP 通过不同的 eBGP 得到目的地址相同，但下一跳不同的多条路由时

在其它条件相同的情况下，将优先选择 MED 值较小者作为最优路由

BGP 路径属性

Well-known mandatory 熟知强制属性：路由器必须实现，且必须出现在 BGP 更新报文中

Well-known discretionary 熟知自选属性：路由器必须实现，但不一定出现 BGP 更新报文中

Optional transitive 可选传递属性：没有实现该属性的路由器必须传递给下一个路由器

Optional non-transitive 可选非传递属性：没有实现该属性的路由器必须丢弃该属性

BGP 路由选择决策

最简单：烫手山芋路由：选择能最小化自己的带宽使用的下一跳

BGP 边界网关协议和 IGP 内部网关协议

BGP 路由选择决策流程

BGP 的更多话题

IBGP，可达性，安全性，性能，应用

（1）IBGP 内部边界网关协议

在同一个 AS 内部的路由器之间传播 BGP 路由信息

IBGP 必须全连接 full mesh：避免路由环路

IGP 内部网关协议和 BGP 边界网关协议

IGP 与 BGP 路由表是独立的

同一个 AS 内不同 BGP 路由器间建立的 IBGP 会话需要 IGP 协助

如果支持转接流量，那么不同 BGP 路由器需要交换外部路由信息

IBGP 与 EBGP 外部边界网关协议在路由中的不同

EBGP 和 IBGP 在扩散 BGP 路由时，对 next-hop 属性的处理不同

BGP 路由黑洞

有个 AS 声称能到，其实到不了

解决方案

IBGP 可扩展性解决方案

路由反射器 Route Reflector，RR 方案

RR 方案

BGP 联盟（BGP Confederation）方案

（2）可达性

IGP：拓扑连通则可达

BGP（策略路由）：拓扑连通不一定可达

（3）安全性

让 AS“证明”它有这条路由

AS 转发分组的路径可能与它宣告的不一样

（4）性能

收敛性（遵从“Gao-Rexford 规则”的策略可收敛）

基于策略的路由不一定是最短路由

（5）BGP 的应用

转接 AS 需要 BGP

5.2.4 小结

5.3 策略路由技术

Policy routing/Forwarding：根据管理员制定的规则转发分组

例：Cisco 路由器（route map）

5.4 多播路由技术

4.2.3 有简介

Unicasting 单播 VS. Multicasting 多播

Emulation of Multicasting with Unicasting 用单播实现多播

How does the multicast router route 多播路由

IGMP 基本工作原理

5.4.1 多播树

Optimal Routing 优化路由：Shortest Path Trees 最短路树

Shortest path tree in unicast routing 单播最短路树

Shortest path tree in multicast routing 多播最短路树

每个路由器给每个多播组构造一棵多播树

两种：有源树（源点基准树），分享树（组共享树）

Shortest Path Tree in Unicast Routing 单播树

Shortest Path Tree in Multicast Routing 多播树

Source-Based Tree 有源树：每个路由器给每个组+源构造一棵最短路树

Group-Shared Tree 共享树：只有核心路由器参与多播，它给所有源构造相同的最短路树（汇集点树，RPT）

RPT 对于某些多播组可能不是最优的

有源树

每一对（多播源，多播组）有一个树

典型的有源树多播路由协议：DVMRP，MOSPF，PIM-DM

有源树示例

N 个组，M 个源——N×M 个有源树

组共享树

每个多播组有一个树

典型的有源树多播路由协议：CBT，PIM-SM

组共享树示例

N 个组，M 个源——N 个共享树

双向共享树示例

单向共享树示例

多播源如何将多播信息流送至根?——用有源树或单播隧道传给核心路由器

单向共享树+有源树

单向共享树+单播隧道

有源树 vs. 组共享树

有源树的树根就是多播源，共享树的树根不是多播源

有源树个数随（多播源，多播组）的增加而增加，共享树个数随多播组的增加而增加

有源树是从多播源到所有组成员的最短路径树，共享树只是从汇集点到所有组成员的最短路径树

（4）剪枝，pruning

剪掉树中没有多播接收成员分支的过程，有时效性

（5）嫁接，grafting

剪掉的分支上如果又出现了多播接收成员，则将其快速接回树上

5.4.2 多播路由协议

DM and SM Protocols

DM（Dense Mode，密集模式）协议：假设都加入（洪泛），有不加入的再剪枝

SM（Sparse Mode，稀疏模式）协议：假设都不加入，有加入的再加进来

DM 模式

高效洪泛

Reverse path forwarding（RPF）反向路径转发

看源 IP 在哪，决定向哪转发

Outgoing interface list 输出接口列表：由多播树构造，进行转发

RPF Check Example

分组到达错误接口——丢弃

分组到达正确接口——按输出接口表转发

SM 模式

首先需要确定 RP，组播源需要显式注册到 RP，组播目的节点显式加入共享树

PIM-DM

Protocol Independent Multicast—Dense Mode 协议无关多播——密集模式

特点：协议无关，有源树，DM 协议

PIM-SM

Protocol Independent Multicast—Sparse Mode 协议无关多播——稀疏模式

特点：协议无关，单向共享树+有源树，SM 协议

MOSPF

Multicast Open Shortest Path First 多播开放最短路径优先

特点：链路状态路由选择 IGP，有源树，LS 协议

DVMRP

Distance Vector Multicast Routing Protocol 距离向量多播路由选择协议

特点：距离向量路由选择 IGP，有源树，DM 协议

CBT

Core-Based Tree 有核树

特点：协议无关，双向共享树+单播隧道，SM 协议

5.4.3 小结

第 6 章 未来网络

6.1 点对点通信 Peer-to-Peer

Communications

Why does P2P get attention?
P2P 的流量猛增
Idea of P2P 思想
分布式的极端
Classic Client/Server System 传统 C/S 系统
每个实体都有不同的身份
Pure P2P architecture 纯 P2P 结构
没有长期的服务器
完全是端系统直接通信
点间歇地连接起来或改变 IP 地址
以视频直播为例
主播的直播软件把视频数据分块，分别发给一部分离主播近的用户
用户将数据扩散出去，集齐所有数据块就能恢复直播内容
有效消除网络传输性能瓶颈，降低对分组交换能力的要求
P2P Applications 应用
BitTorrent
Tracker（追踪器）、种子文件
如何工作：
1.BT 客户端首先解析种子文件得到 Tracker 地址，然后连接 Tracker 服务器
2.Tracker 服务器回应下载者的请求，提供下载者其他下载者（包括发布者）的 IP
3.下载者再连接其他下载者，根据种子文件，两者分别告知对方自己已经有的块，然后交换对方所没有的数据
P2P Case study：应用：Skype
首个 P2P 的 IP 电话网络
三种实体：超级节点 SN，普通节点，登录服务器
Peers as relays 中继节点
问题：对话的两方都使用 NAT
解决方案：使用 SN 进行中继
File Distribution 文件分发：Server-Client vs P2P
C/S：从一个服务器向 N 个用户发文件，发送了 N 个备份
P2P：服务器只用发出一个文件
Server-client vs. P2P: example
Promising properties of P2P 预期优势
自组织，大容量，自治，抗 DoS 攻击，负载均衡，抗审查
P2P 的历史（工业界）
Napster 运行原理
Gnutella
Gnutella 工作原理
为 P2P 正名
P2P 核心问题：怎么定位节点
Unstructured 去结构化 vs Structured 结构化
去结构化允许资源放在任何节点。拓扑是任意的，增长是自发的。
结构化通过定义拓扑结构和资源放置规则，简化了资源放置和负载均衡。
Distributed Hash Table 分布式哈希表 (DHT)
键值对存储在 DHT 中
任何参与节点可以高效地获取键对应的值
Structure of a DHT 结构
An abstract key-space（抽象的键空间）
A key-space partitioning scheme（键空间分割机制）
An overlay network（覆盖网络）
How dose DHT work？工作方式
在 DHT 上存储文件名为 A，内容为 B 的数据
任意节点通过 DHT 上读取文件名为 A 的文件内容
Keyspace partitioning 键空间分割
最重要的好处：移除或增加一个节点只改变拥有相邻 ID 的一组键，不影响其他节点
结构化 P2P 信息系统工作原理（以 Chord 算法为例）
Chord 资源定位
基于 DHT 的 P2P 应用架构
DHT Layered Architecture 层次结构
四大结构化模型
结构化 P2P 的特点
Chord
CAN(Content-Addressable Networks)
Kademlia : BitTorrent DHT
Kademlia : XOR based closeness
All in the application layer 都实现在应用层
Summary : Unstructured vs Structured 总结：去结构化和结构化
去结构化：网络简单，负载大
结构化：网络复杂，效率高

Information Centric Networking

Popular Conception:Content Distribution Over the Internet Does Not Scale
互联网的内容分发是无法衡量的
Why Content Networking Is Proposed?
内容网络好在哪里
Problems with Today's Networks 今天网络的问题
URL 和 IP 负载过重
无法追踪一个备份
信息传播低效
无法信任一份不信任的节点发送的备份
应用和内容独立
信息中心网络（Information Centric Networking）
信息互连
Focus on information objects 关注信息本身

IP VS. CCN (Content-centric networking 内容中心网络)（一种 ICN）

支持内容存取

内容块（Content chunk）代替了 IP

网络中内建存储功能

基本思路

Name Resolution and Name-based routing 基于文件名路由

6.2 软件定义 SDN

传统网络：Per-router control plane

每个路由器独立计算

除了转发，路由器还能做什么？

SDN：Logically centralized control plane 中心化控制平面

远程控制器交互本地路由表

Generalized forwarding in data plane: “match-plus-action” 泛化转发

数据平面泛化转发：匹配+动作

Generalized Forwarding and SDN 泛化转发和 SDN

中心化的路由控制器计算并给每个路由器分发流表

OpenFlow data plane abstraction 数据面抽象

首部域定义了流

泛化转发：简单的数据包处理规则

流表定义了路由器的匹配+动作规则

OpenFlow: Flow Table Entries 流表条目

Examples 例

OpenFlow example

Components of the SDN architecture 结构组成

数据面交换机，SDN 控制器，网络控制应用

SDN 控制器架构

Interface, abstractions for network control apps 接口、管理控制 APP 的抽象

Network-wide distributed, robust state management 网络内的分布式、鲁棒状态管理

Communication to/from controlled devices 与受控设备的通信

SDN: control/data plane interaction example 控制、数据平面交互例

OpenFlow protocol

控制器给交换机的信息

交换机给控制器的信息

SDN: control/data plane interaction example

传统网络架构 V.S. SDN 架构

将网络设备的控制和转发功能解耦，使网络设备的控制面可直接编程

将网络服务从底层硬件设备中抽象出来

控制面的抽象考虑

普适转发模式，网络状态，规范化配置

1、转发模式抽象

关于 OpenFlow

标准化交换机

OpenFlow 控制消息

OpenFlow 带来的：

OpenFlow 交换机简单（以太网交换机、或 IP 路由器设备、防火墙、NAT）

2、网络状态的抽象

网络的全局视图

常规的交换机/路由器

邻居之间运行分布式算法

Software Defined Network (SDN)

控制平面与数据平面物理上分离，单个控制系统多台转发设备

SDN：接受流、选择流的路由

3、规范化配置

不负责实际物理网行为的具体实现，只配置交换机的转发表

例：访问控制

SDN 结构模型

应用控制：接入控制、转发系统、防火墙、二层交换

↑北向接口：应用编程接口

控制平面(SDN 平台)：控制软件（Open DayLight）→东西接口

↓南向接口：数据面接口(OpenFlow)

数据平面

SDN 实际应用

SDN 路由应用

路由应用软件将路由结果下发给 SDN 平台，SDN 平台再将流表下发到相关交换机

访问控制应用

应用控制软件决定“谁可以和谁对话”

灵活组网

将主机组成两个独立的以太网

关于 SDN 的思考

SDN 有多重要

SDN：完全改变了网络的生态系统

软件定义网络行为

自治系统间的应用——分布式控制平面

SDN 处理的是“Flow”——流表优化

网络边界上的 SDN

SDN 的未来

Segment Routing 段路由

什么是 Segment Routing（SR）

一种（可编程）源路由协议

为什么需要 SR？

SR 支持两种转发平面

SR-MPLS、SRv6

如何构成“路由段”

Prefix Segment（前缀段）、Adjacency Segment（邻接段）、混合段

SR 与 SDN

SRv6 是一种 Hybrid 的 SDN 架构

SR 同时支持传统网络和 SDN 网络

保障现有网络平滑演进到 SDN 网络

网络编程

Hybrid SDN

可扩展、可靠性、高性能

SRv6 是原生 IPv6（Native IPv6）

SRv6 报文格式

SRv6 SID (Segment Identifier) 结构

Locator、Function、Args

SID 举例

常见 SID

SRv6 如何工作的？

6.3 云计算与虚拟化

Cloud computing 云计算

云计算定义

分布式处理、并行计算、网格计算

云计算技术

云计算优点

数据在云端、软件在云端、无所不在的计算、无限强大的计算

云计算类别

IaaS（Infrastructure as a service——基础设施即服务）

PaaS（Platform as a service——平台即服务）

SaaS（Software as a Service——软件即服务）

云计算关键技术

虚拟化，分布式，并行计算，海量存储，桌面应用，资源调度，安全

虚拟化技术

对象是各种各样的资源

逻辑资源对用户隐藏了不必要的细节

在虚拟环境中实现其在真实环境中的功能

典型的虚拟化

网络虚拟化，存储虚拟化，桌面虚拟化，服务器虚拟化，应用虚拟化

例如：系统虚拟化

Virtual Machine Concept 虚拟机

用软件模拟物理服务的特征

虚拟机监视器：用软件提供物理硬件到其支持的虚拟机的抽象

两种类型的 Hypervisor 虚拟机监视器

在物理主机上部署，能直接控制主机资源。（裸机）

部署在操作系统自身和资源之间，依靠操作系统，在监视器上处理硬件交互。

发展：Container Virtualization 容器虚拟化

Network Virtualization 网络虚拟化

网络虚拟化概念

Network Functions Virtualization (网络功能虚拟化 NFV)

用软件和虚拟机实现网络功能

部署服务时也不要跟具体网络系统挂钩

计算机虚拟化

网络虚拟化

虚拟链路

虚拟网络接口卡(NIC)

虚拟交换机

vSwitch，OvS

网络虚拟化—虚拟与现实

用虚拟拓扑结构，覆盖在实际网络之上

两个维度

分布式——抗毁

集中式——可信

软件——灵活

硬件——性能

6.4 无线 Ad hoc 网络

6.4.1 基本概述

Ad hoc 术语的含义

为某个特殊的目的、临时的、事先未准备的

Ad hoc 网络的历史

移动 Ad hoc 网络

移动 Ad hoc 网络的特点

无固定基础设施

Ad hoc 网络与常用无线网络比较

其它差异

典型的无线 ad hoc 网络包括

无线传感器网络（WSN）、无线网格网络（WMN）、车联网（IoV）

无线 Ad hoc 网络的应用

6.4.2 无线 Ad hoc 网络的体系结构

平面结构、分层结构

分级结构

单频分级、多频分级

基于虚拟骨干网的单频分级

使用多频的两级结构

平面结构 vs 分层结构

6.4.3.1 无线 Ad hoc 网络的路由协议

为何要针对无线 Ad hoc 网络设计新的路由协议

无线 Ad hoc 网络对路由选择的要求

建立路由时间越短越好

路由控制报文数量越少越好

路由长度越短越好

无线 Ad hoc 网络路由协议分类

先验式/表驱动路由协议

反应式/按需路由协议

混合式路由协议

路由协议举例

AODV、ZRP

6.4.3.1 无线 Ad hoc 网络的功率控制

功率消耗源

与通信有关、与计算有关

功率控制方法

6.4.3.1 无线 Ad hoc 网络的 MAC 协议

现有无线 MAC 协议的缺陷

挑战：隐藏站点问题

暴露站点问题

无线 Ad hoc 网络中 MAC 层需要解决的主要问题

多跳共享问题、隐藏终端问题、暴露终端问题、节点移动的影响

无线 Ad hoc 网络 MAC 协议分类

单信道协议、双信道协议、多信道协议

CSMA/CA 协议

载波侦听，随机后退，避免冲突

载波侦听

物理层载波侦听、MAC 层虚拟载波侦听

载波侦听——MAC 层虚拟载波侦听

RTS 与 CTS

避免冲突，采用 RTS/CTS 握手机制

能较好的解决隐藏站点的问题

随机后退机制

IEEE 802.11 中的三地址结构