高等计算机网络

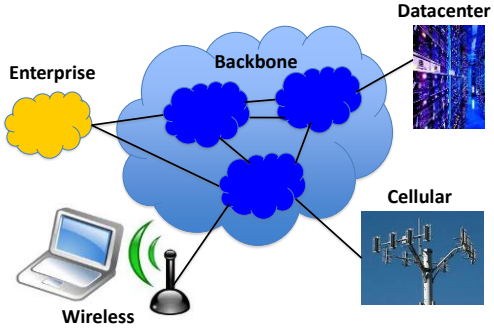
李巍（liw@buaa.edu.cn）

北航计算机学院 2020 秋季

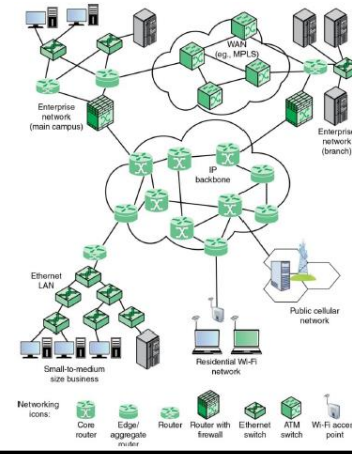
目标

- 基于计算机网络分层体系结构的框架，掌握Internet网络互连的基本原理；理解Internet核心协议和算法的设计思想和方法。
- 将层次结构的方法和系统分析方法相结合，理解新型网络系统与技术的基本原理，为未来从事网络技术和应用研究奠定理论基础。
- 理解计算机网络前沿应用研究的重要协议和关键算法，跟踪当前网络研究的一些热门领域、研究方法和研究方向，引导研究工作。

计算机网络的互连



网络生态系统



网络生态系统

- 端用户
- 服务提供商
  - 网络提供商
  - 应用提供商
  - 内容提供商
- 网络体系结构
  - Internet
    - 路由器：核心路由器，边缘路由器
    - 核心/主干网；接入网
  - 数据中心网络
  - 物联网IoT/Fog network
  - 云计算网络
  - 边缘计算
  - 5G
  - 。。。。。。

新型应用对网络的需求

- 网络技术演变的驱动力：应用的发展
  - 传统Internet应用
    - 文件传输，电子邮件，远程登录。。。
  - WWW
  - 音视频、流媒体
  - 移动计算
  - 云计算
  - 大数据
  - 人工智能
  - 沉浸式体验、区块链、量子计算等。。。

如何降低网络对于应用性能的影响?

应用实例

- 高清和超高清互联网视频将占全球互联网流量的64%
  - 激增的视频流量和工业机器应用带来了大量的拥塞崩溃和数据包延迟。
- 工业互联网中的数据上传和控制指令下发、远程机器人手术、无人驾驶、VR游戏等
  - 需要将端到端时延控制在微秒到几毫秒量级，将时延抖动控制在微秒级，但传统的网络只能将端到端的时延减少到几十毫秒。
- 除此之外，网络的时延成为影响集群计算性能的首要指标，深度学习、分布式计算、分布式存储、计算存储分离等技术对数据中心网络低时延特性提出迫切需求。

未来网络的发展

- 大量的应用需求是生产型服务
  - 需要确定性、差异性、强调 QoS 的能力
  - “尽力而为”的传统网络架构难以支撑未来应用对差异性服务质量保障、确定性带宽和时延的需求
- 《未来网络发展白皮书（2019版）》
  - 预计到2030年，未来网络将具备支撑万亿级连接服务等7种能力

未来网络应具备的能力

1. 支持超低时延、超高通量带宽、超大规模连接的能力；
2. 满足与实体经济融合的需求，具备支持差异化服务的能力；
3. 实现网络、计算、存储多维资源一体化，并具备多维资源统一调度的能力；
4. 实现海陆空天一体化融合的网络架构；
5. 做到简化硬件设备功能的同时保证其处理性能，并通过软件定义的方式增强网络弹性；
6. 具备“智慧大脑”，实现网络运维智能化；
7. 成为一个内生安全、主动安全的网络，进而更好地维护全球网络安全。

《未来网络发展白皮书（2019版）》

网络技术的演化

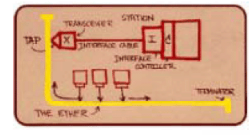
- 以太网→ VLAN →企业网络和数据中心网络→云计算的网络（2013：SDN；虚拟化NFV；容器网络；云原生）
- 无线网络技术→移动网络→物联网IoT→边缘计算
- 传统Internet →下一代Internet （中国未来网络(FutureNetworks)，Planetlab、美国GENI、欧盟Onelab、Corelab）
- 目标

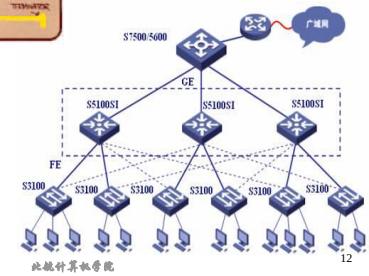
highly robust　highly efficient　highly flexible　highly secure

演化路线-1

- 以太网→ VLAN →企业网络和数据中心网络→云计算的网络
  - 组建网络
  - 管理网络
  - 优化网络

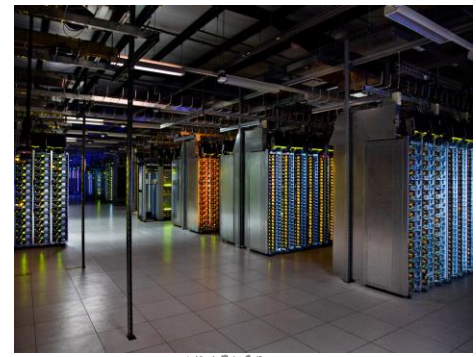
以太网





数据中心

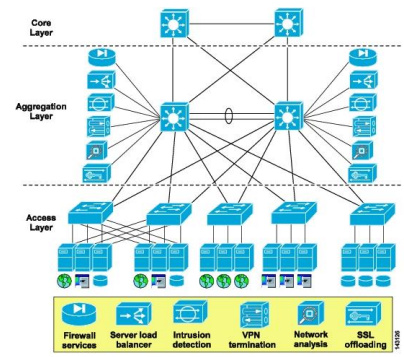
Google 数据中心网络



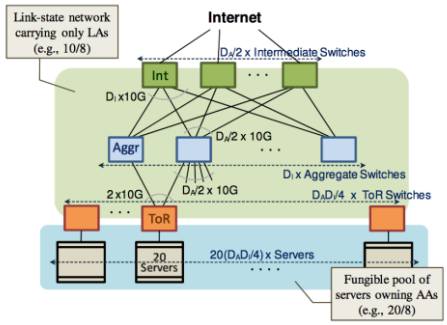
Google worldwide network

GDG DevFest 2013 Season

数据中心网络



微软的VL2数据中心架构



- 微软在2009年提出VL2数据中心架构（SIGOMM'09）

VL2

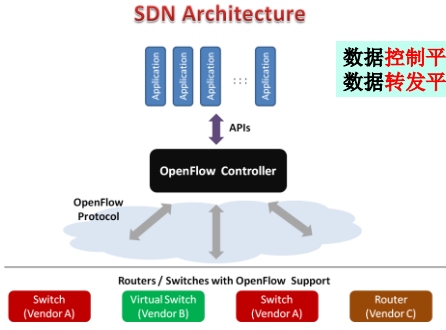
- VL2架构：底层服务器和上层交换机两层架构
- 机架交换机（top of rack，简称ToR）连接
  - 交换机层包括汇聚交换机（Aggregate Switches）和中继交换机（Intermediate Switches）
  - 汇聚交换机和中继交换机之间的链路连接形成完全二分图，网络采用CLOS架构，扩展链路带宽。
  - 每个汇聚交换机都可以通过中继交换机与其他汇聚交换机相连。
  - 这种设计增加了路径数量和网络的健壮性

数据链路层(Layer 2) vs. 网络层(Layer 3)

- 以太网交换机 (layer 2)
  - ✓ Auto-configuration (plug & play) 自动配置
  - ✓ Seamless mobility, migration, and failover 无缝迁移
  - x Broadcast limits scale (ARP) 广播域制约
  - x Spanning Tree Protocol 生成树协议
- IP 路由器 (layer 3)
  - ✓ Scalability through hierarchical addressing 分层架构
  - ✓ Multipath routing through equal-cost multipath 多路径路由
  - x More complex configuration 配置复杂
  - x Can't migrate w/o changing IP address 迁移困难

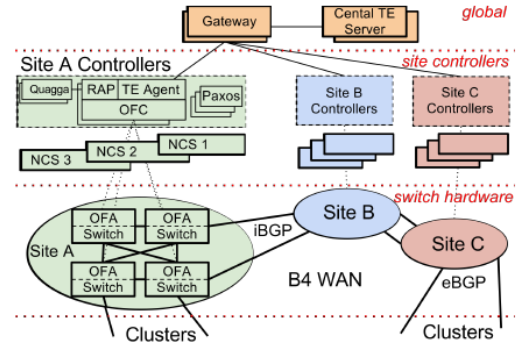
如何管理网络?

软件定义网络 SDN



数据控制平面与
数据转发平面分离

Google B4 SDN网络体系结构



P4

- 目前软件定义网络的可编程性局限于网络控制平面，其转发平面在很大程度上受制于功能固定的包处理硬件
- "P4"（www.P4.org）高级语言
  - 具有协议无关性、目标无关性以及现场可重配置能力
  - 它能够解决 OpenFlow 编程能力不足以及其设计本身所带来的可拓展性差的难题
- 可编程数据平面
- P4 定义数据包的处理流程，然后利用编译器在不受限于具体协议的交换机或网卡上生成具体的配置，从而实现用 P4 表达的数据包处理逻辑
- P4 联盟提出了带内网络遥测（In-band Network Telemetry，INT）

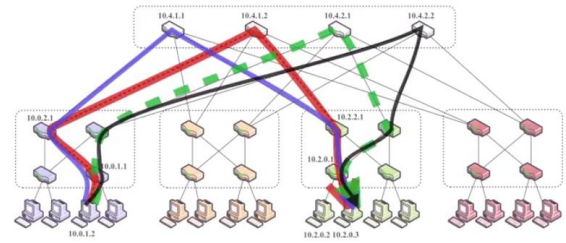
网络功能虚拟化NFV

- NFV: Network Function Virtualization
  - 将路由、防火墙、入侵检测、NAT等网络功能（NF）从专用硬件平台分离出来，并用软件实现
  - 需要高性能硬件平台支持
  - 可用于数据平面功能和控制平面功能
- 与SDN的关系
  - 相互独立
  - 虚拟化平台：SDN+NFV

如何优化网络?

路由和流量工程

Multipath routing



Segment Routing

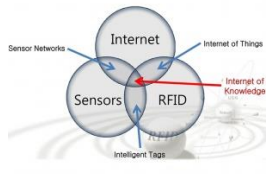
- IETF 支持 SDN 架构的新型路由转发协议（SR）
  - 是一种源路由机制，用于优化 IP、MPLS 的网络能力
  - 以更加简单的方式提供 TE、FRR、MPLS VPN 等功能。
- 在未来的 SDN 网络架构中，Segment Routing 将为网络提供和上层应用快速交互的能力。
  - SR 通过 SDN 控制器，可以根据网络状态，进行源路由路径控制，无需修改路径上网络设备的路由信息，从而使得大规模部署流量工程变得简单可行。
  - 未来支持城域网、广域网、核心网等 SDN 的规模应用

Intent-Based Networking

- 思科提出的一种新的网络控制和管理理念
  - 用户只需要提供目的，由网络设施自动翻译为网络配置指令执行，并不断收集和监控网络运行状态进行反馈，从而实现持续优化网络的目的。
- 适用于数据中心， 园区网和广域网
- 基于 SDN 与 AI 的 IBN 系统能够灵活、快速地执行用户的策略和意图，实现自动化运维，使业务目标与网络结果保持一致。

演化路线-2

- 无线网络技术→移动网络→物联网→边缘计算
  - 物联网IoT（Internet of things）



移动计算：Sensor network

物联网（IoT）的发展

- 物联网由连接到互联网并彼此共享数据的设备组成。
  - 物联网设备不仅包括计算机，笔记本电脑和智能手机，还包括配备有芯片以通过网络收集和通信数据的对象，如联网汽车，自动售货机，智能可穿戴设备，手术医疗机器人等
  - 2020年IoT设备达到百亿数量级
- 负载迁移
  - 从本地数据中心迁移到云
  - 从云数据中心迁移到更靠近所处理的数据源的“边缘”位置
- 目标
  - 缩短数据的传输距离，从而消除带宽和延迟问题，最终提升应用和服务的性能和可靠性，并降低运行成本。

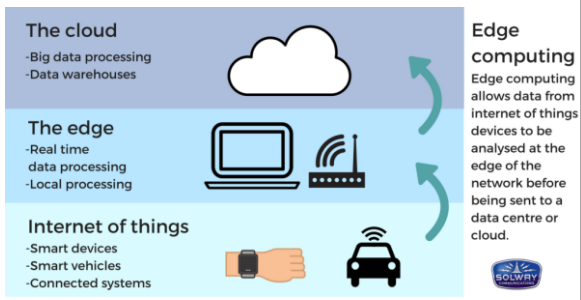
边缘计算Edge Computing

- Gartner defines edge computing as “a part of a distributed computing topology in which information processing is located close to the edge – where things and people produce or consume that information.”
  - computation and data storage

边缘计算 Edge Computing

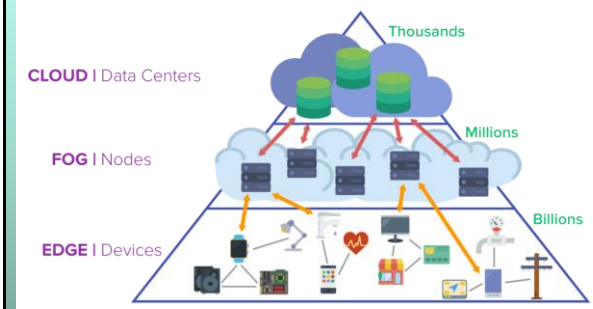
- 边缘计算指的是靠近物或数据源头的网络边缘侧，融合网络、计算、存储、应用核心能力的开放平台
- 该平台就近提供边缘智能服务，满足行业数字在敏捷联接、实时业务、数据优化、应用智能、安全域隐私保护等方面的关键需求。
- 2014 年 12 月，ETSI 成立了 MEC ISG 工业标准组，提出了 MEC 标准草案，并于 2016 年将此概念扩展为多接入边缘计算（Multi-Access Edge Computing，MEC）。
- 2017 年 3 月，IEEE 推动边缘计算成为P2413（Standard for an Architectural Framework for the Internet of Things）重要内容之一。

- 在产业界方面，2018 年 12 月，工业互联网联盟(IIC)和开放雾联盟(Open Fog Consortium)宣布合并，共同推进工业物联网、雾和边缘计算领域协调发展。
- 2018 年 2 月，AT&T 宣布开源其基金会项目 Akraino，该项目是为在虚拟机和容器中运行电信运营商级边缘计算应用而设计，以支持商用级边缘计算应用的可靠性和性能要求。
- 2019 年 1 月，Linux 基金会宣布推出 LF Edge 开源国际组织，旨在建立独立于硬件、芯片、云或操作系统的一个开放的、可互操作的边缘计算框架。



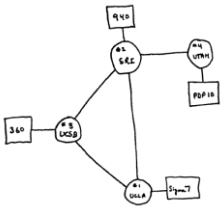
- 云cloud
- 边缘 edge
- 物联网 IoT

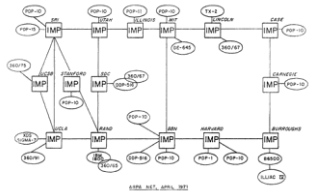
边缘计算Edge Computing



演化路线-3

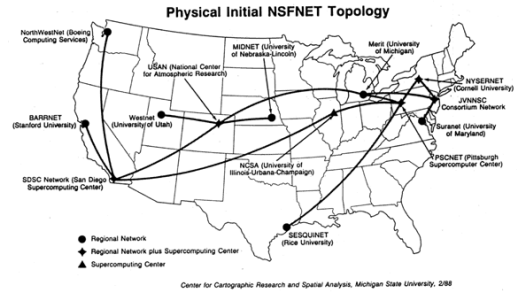
◆传统Internet →下一代Internet



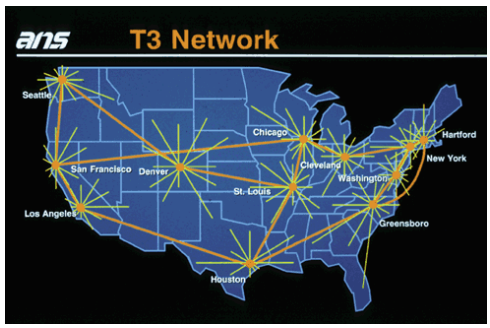


1969年 4个节点的ARPANET　　1971年 ARPANET

NSFNet T-1 Backbone Map, 1988

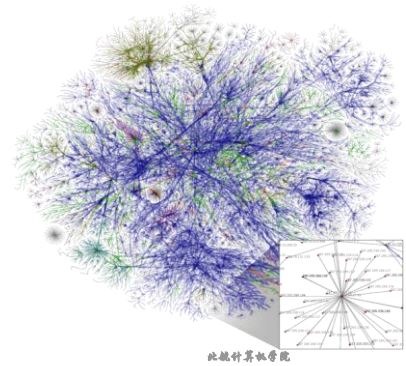


T-3 Network Map, 1991



Internet map

January 15, 2005



现有网络面临的挑战

- Internet 从最初的单一军用网络不断在向包括军用、民用、商用等的各个方面普及
  - 传统 TCP/IP 网络设计目的是进行高效的数据传输： IP 协议提供“尽力而为”的服务，TCP 使用的重传和滑动窗口机制实现可靠传输
- 挑战
  - 给实时数据的传输带来难以预料的时延及抖动
  - 无法保证吞吐量和传送时延等服务质量（Quality of Service，QoS）要求。
  - 可扩展性、安全性、管控性、移动性、服务分发能力、绿色节能等一系列问题

低时延-确定性时延网络

- 工业控制、远程医疗、机器人、VR 游戏、导弹控制等场景需要端到端时延的精准控制，要求毫秒级的时延和微秒级的抖动；
- 数据中心、5G 物联网等场景下，高性能计算、大数据分析和浪涌型 IO 高并发等技术要求网络满足超低时延的要求。
- 由此可见，低时延、确定性时延成为时延敏感型业务的迫切需求。

未来网络架构

- 2005年开始，国外学术界提出了“从头再来(Clean Slate)”的设计思想，
  - “未来互联网”(Future Internet)或“未来网络”(Future Network)。
- 与此同时，美国、欧盟、日本等发达国家都各自提出了未来网络计划
  - Plantlab、GENI、Onelab等一系列基础实验网络建设项目
  - 谷歌等率先在数据中心部署应用了Openflow/SDN技术，
  - 2012年美国SDN相关并购金额达到50多亿美元2013年，思科、爱立信、IBM等成立了开源SDN组织

下一代 vs. 未来互联网

- 中国
  - 2003年正式启动下一代互联网示范工程（CNGI），建成CERNET
  - 2013年2月，以SDN、云服务等为内容的未来网络CENI被正式立项为“国家重大科技基础设施建设中长期规划”
  - 2017年11月26日印发《推进互联网协议第六版（IPv6）规模部署行动计划》
- 美国：未来互联网
  - 2005年，美国NSF资助FIND（FutureInternetNetworkDesign，未来互联网网络设计）、GENI（Global Environment for Network Innovations，全球网络创新环境）
  - 2010年美国NSF资助未来网络架构研究项目，其中包括命名数据网络（Named Data Networking，NDN）
  - 2014年，组建“命名数据网络(NDN)联盟”

信息中心网络

- **信息中心网络（Information-Centric Networking，ICN）**
  - **Xerox PARC 研究中心和 ULCA 大学的科学家提出**
  - **以信息命名方式取代传统的以地址为中心的网络通信模型实现用户对信息搜索和信息获取**
  - **增强互联网安全性、支持移动性、提高数据分发和数据收集的能力、支持新应用与新需求**
- **集中式架构和分布式架构**
- **典型的分布式架构：命名数据网络（Named Data Networking，NDN）**

NDN

- 命名数据网络（Named Data Network，NDN）
  - 美国国家科学基金会（NSF）于2010 年在未来网络架构（FIA）项目中重点资助的四个项目之一
  - 替代现有的以 IP 为核心的网络体系架构
  - “以数据为中心”将通信范式的重点从关注于“where”（地址、服务器、主机）转变到“what”（通信的内容）。
  - 以对数据命名的方式代替位置（IP地址），将数据转变成网络的第一要素。

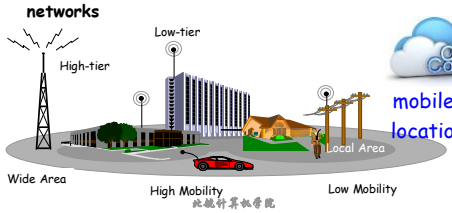
我国IPv6的发展现状

- 截至2019年6月，我国IPv6地址数量为50286块/32，较2018年底增长14.3%
- IPv6活跃用户数达1.3亿，基础电信企业已分配IPv6地址用户数12.07亿；域名总数为4800万个，其中“.CN”域名总数为2185万个，较2018年底增长2.9%，占我国域名总数的45.5%。

网络的复杂性……

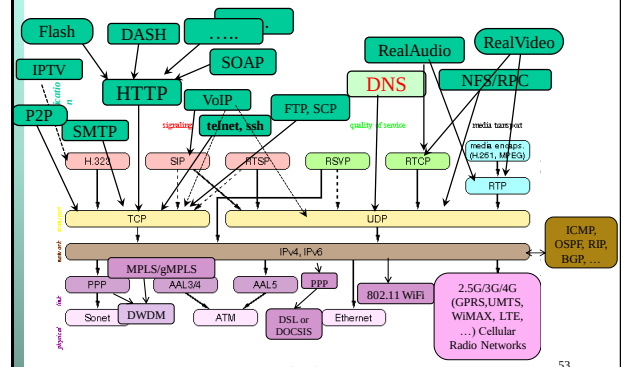
接入Internet：更多接入方式

- **服务器，桌面设备，便携设备**
  - 智能电话，平板，电子阅读器
  - 电视，智能测量装置等
- **无线通信技术：Internet的变革**
  - WiFi, bluetooth, 3G、4G、5G cellular networks

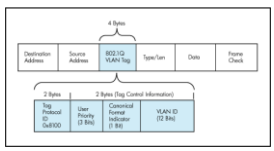


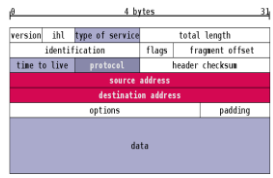
mobile computing
location services

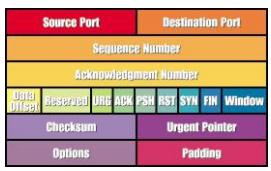
Internet 的协议

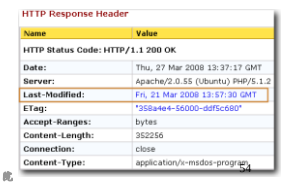


协议字段（首部）？



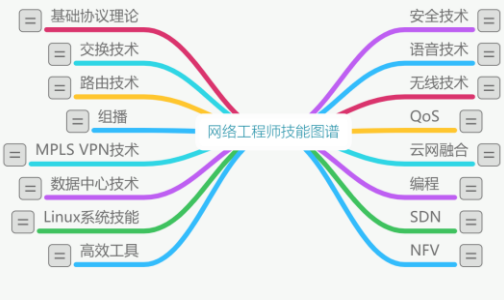






学习视角

- **应用程序员**
  - 设计应用程序， 利用网络进行有效的信息传输
- **网络运营商**
  - 网络部署、配置、故障处理、计费
- **网络设计者**
  - 网络资源分配的各种策略和折中
  - 协议设计



https://www.sdnlab.com/23519.html

基本原理 vs. 前沿研究

基本原理

分层体系结构

The Classical Past (textbooks)

- 问题提出
- 研究思路和方法

◆ 分层Layering

◆ 传统的Internet体系结构

- 路由Routing (IP)
- 网络资源管理和服务质量
- 传输Transport (TCP)
- 命名机制Naming (DNS)

前沿研究

the state of the art

- 新型网络系统结构
- 新的协议和算法

◆ 热点：

- SDN, NFV
- 云计算和数据中心中的网络技术
- 移动和无线网络Mobility/wireless
- P2P 应用
- 内容分发网络 CDN
- 下一代互联网
- 网络安全
- ....

课程大纲

◆概述

◆网络互联基础

- Internet设计原理
- 网络层基础
- 路由技术
  - BGP，组播路由、选播路由等
- 路由器体系结构

◆软件定义的网络SDN

- 数据平面：Openflow
- 控制平面
- 网络功能虚拟化NFV

◆数据中心网络

- 大二层技术
- 拓扑，性能，虚拟化

课程大纲

◆网络资源管理

- 网络层的拥塞控制
- 排队规则
- 传输层的拥塞控制
- 深入理解TCP协议

◆应用层网络

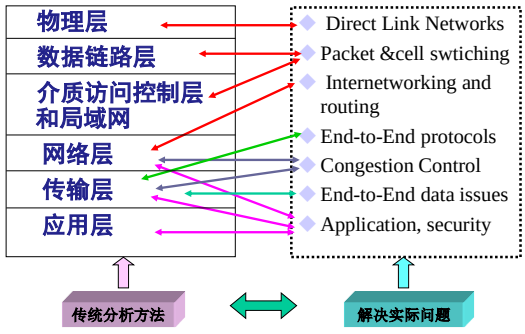
- P2P和DHT
- 内容分发CDN
- 网络测量

◆移动计算和无线网络概述

◆网络安全概述

◆课程设计及专题讨论

分层结构和实际网络的关系



相关理论?

- 算法与数据结构 Algorithms and data structures
  - 分组转发和分类
- 控制论Control theory
  - TCP
- 排队论Queuing theory
  - 包交换，统计时分复用
- 优化理论Optimization theory
  - TCP拥塞控制，流量工程
- 博弈论 Game theory and mechanism design
  - 域间路由，无线网络
- 形式化方法 Formal methods
  - 协议设计和形式化
- 信息论 Information theory
  - 网络特征测量和推断
- 密码学 Cryptography
  - 安全协议

相关理论?

- 程序设计 Programming languages
  - 配置，SDN
- 图论 Graph theory
  - 网络规划，流量矩阵分析
- 分布式系统 Distributed systems
  - 应用
- 操作系统 Operating systems
  - 主机协议栈
- 计算机体系结构 Computer architecture
  - 网卡，路由器硬件
- 软件工程 Software engineering
  - 协议设计，需求分析
- ...

主要国际会议

- Core networking conferences and journals
  - SIGCOMM, NSDI, HotNets, IMC, CoNEXT, CCR, INFOCOM, ACM/IEEE ToN
- Wireless
  - MobiCom, MobiSys, HotMobile, SenSys, IPSN, Percom
- System and Networking
  - SOSP, OSDI, USENIX ATC, HotOS, ICDCS
- Security and Networking
  - CCS, USENIX Security, NDSS, IEEE Symposium on Security and Privacy
- Theory and Networking
  - SIGMETRICS, PODC, SPAA, MobiHoc
- Multimedia Systems and Networking
  - MMSys, NOSSDAV, ACM Multimedia, ACM TOMCCAP, Springer Multimedia Systems Journal, IEEE TMM

参考书目

- 参考教材（中文版，影印版）
  - Andrew S.Tanenbaum, *Computer Networks*, 清华大学出版社，（第五版，《计算机网络》）
  - Larry L.Peterson,Bruce S.Davie, *Computer Network: A System Approach*, 机械工业出版社，（《计算机网络：系统方法》，第五版）
  - James F.Kurose，Keith W. Ross, *Computer Networking: A Top-Down Approach*, 高等教育出版社，（《计算机网络：自顶向下方法》，第六版，第七版，机械

- 学术论文
  - 经典论文 vs. 前沿研究
  - 课程中心网站（course.buaa.edu.cn）本课程站[illegible]中的链接

课程安排

课程要求

- 先修课程
  - 计算机网络基础，计算机组成原理，操作系统，程序设计基础
- 授课方法
  - 课堂讲授（Lecture）
  - 论文阅读与交流（Reading& Discussion）
  - 课程设计和讨论（Project & presentations）
- 考核方式
  - 平时成绩：50%（包括考勤）
    - 考勤：5%
    - 小作业（论文阅读与交流）：15%
    - 大作业（课程设计）：30%
      - 小组讨论：15%
      - 课程论文：15%
  - 期末考试：50%(开卷)
- 授课时间和地点
  - 3—18周
  - 周五：11节-13节
  - 地点：1-201
- 课程中心网站 course.buaa.edu.cn
  - 讲义及参考论文下载
  - 课程安排与要求
  - 作业提交（注意截止期）
    - 按时提交
    - 独立完成
  - 通知
  - 站内消息

小作业说明：论文阅读与交流（1）

- 在课程进行中，按专题组织论文阅读与讨论
  - 论文在课程中心下载
- 主要专题
  - 网络体系结构
  - SDN与NFV
  - 拥塞控制
  - 数据中心网络
  - 应用层网络
  - 网络安全
- 每个专题五篇论文
  - 经典论文 vs 前沿研究
  - 综述 vs. 研究

小作业说明：论文阅读与交流（2）

要求：每个学生独立完成，在课程中心网站上提交

◆ **步骤**

1. **选择阅读的论文**
   - 课程网站给出需要阅读的论文和所属专题（5个专题，约25篇）
   - 整个学期，每个同学至少选择3个专题（1篇论文/专题）完成小作业。
2. **完成论文评论（paper review），要求：**
   - 作者主要观点和要解决的问题
   - 论文中关键技术分析，包括优点和局限性
   - 未来发展方向
   - 其他研究方法等
3. **提交：**
   - 作业提交格式：.docx和.pptx
   - 按时提交：教学网站相关说明（注意截止期）

◆ **课堂交流与讨论： 抽查（也可以自荐），每人介绍5分钟左右**

如何阅读论文（参考）

◆略读
- Skim abstract/intro + section headings + references (5-10min)

◆精读 （忽略细节）
- Read but ignore details (e.g. proofs) (1hr)
- Good general understanding of techniques
- Identify related work you need to look at

◆再现 "Virtual re-implementation" (1-3hrs)
- Identify hidden assumptions
- Identify issues with techniques used

大作业说明：课程设计

分组：
- 自由组合进行分组，每组2~3人（不超过3人）
- 确定选题
  - 需要解决什么问题？
  - 拟采用什么方法？

选题：根据专题建议范围选择类型，也可以自行设计。

三种类型：

（1）网络协议设计与系统实现 design/implementation

（2）基于模拟平台的研究 simulation

（3）网络测量和分析 measurement & analysis

大作业说明：课程设计

提交作业：三个环节

1.计划：小组提交课设计划 （project proposal）
- 包括研究目标，实施方法。成员分工和主要参考资料
- 课堂讨论

2. 进展：提交中期报告

3. 完成
- 每个小组提交一份大作业ppt

  课堂讨论：20页左右，约讲8分钟
- 每人期末提交技术报告

  按要求时间提交（期末考试前）

计算机网络
基本概念回顾

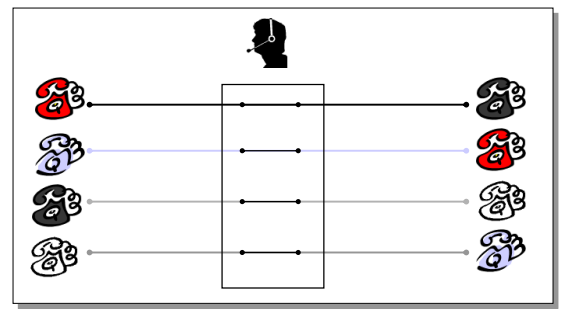
目标：满足应用需求

- 不同计算机上应用系统之间的通信
- 理解应用系统对网络的需求
  - 流量数据率 (Traffic data rate)
  - 流量模式 (Traffic pattern)
    - bursty or constant bit rate
  - 流量目标 (Traffic target)
    - multipoint or single destination, mobile or fixed
  - 延迟敏感性 (Delay sensitivity)
  - 丢包敏感性 (Loss sensitivity)

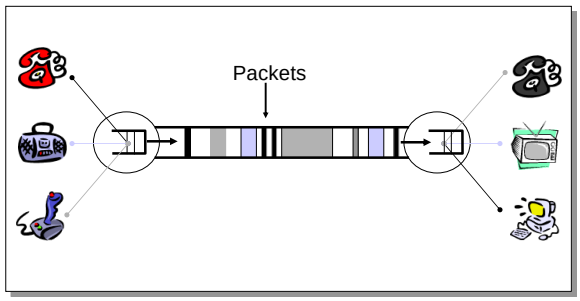
网络需要满足的目标

- 效率(Efficiency) – resource use; cost
- 能力 The “ilities”:
  - 可演化 Evolvability
  - 可管理 Managability
  - 安全 Security (securability, if you must)
  - 易于:
    - 创建 Creation
    - 部署 Deployment
    - 应用 Creating useful applications
  - 可扩展 Scalability

传统的通信方式：电话网络



Internet：分组交换Packet Switching



分组交换的特点

优点：

- 分组的发送方式
  - Interleave packets from different sources
- 高效性： 按需使用资源
  - Statistical multiplexing
- 通用性
  - Multiple types of applications
- 支持突发流量：
  - Addition of queues

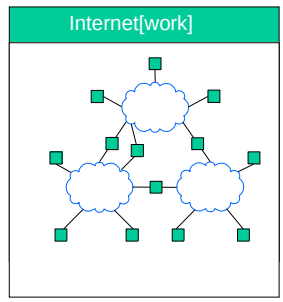
存在挑战：

- 存储-转发：Store and forward
  - 分组的数据结构
  - 独立选择路径：重新排序
- 竞争：Contention
  - 拥塞
  - 延迟
- 互联网络的差异
  - 地址格式，带宽；分组大小；丢包模式
  - 路由等
- 问题提出：
  - 如何在不同网络之间转换？

网络互连

相互连接的网络的集合

- 主机Host: Inetwork endpoints (computer, PDA, light switch, ...)
- 路由器Router: node that connects networks
- 如何进行转换？



需要解决的问题?

- 异构性 Heterogeneity
  - 地址：Address formats
  - 性能：Performance – bandwidth/latency
  - 包大小：Packet size
  - 丢包率：Loss rate/pattern/handling
  - 路由：Routing
  - 不同网络技术：Diverse network technologies
    - satellite links, cellular links, carrier pigeons

基本方法

- 如何发现节点？
- 路由和转发？
- 差错处理？
- 链路过载？
- 传输差错？

如何发现节点?

Computer 1　　Computer 2

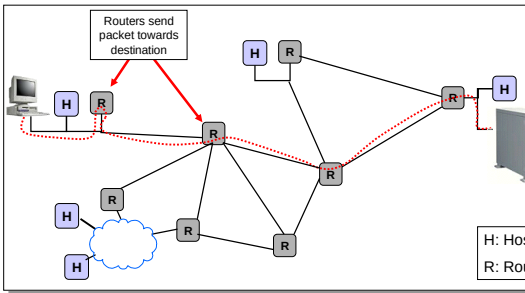
Need naming（**命名**）and routing（**路由**）

命名Naming

Computer 1　　Local DNS Server

主机名字（可读的）→ IP地址（logical endpoints）

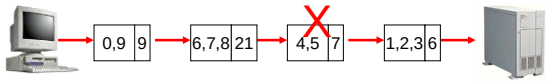
路由 Routing



如何处理数据传输中的错误?

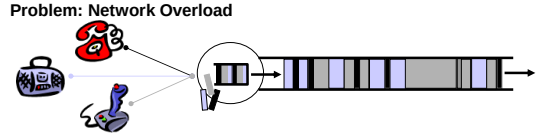
Problem: Data Corruption

Solution: Add a *checksum*



如何处理网络过载?

Problem: Network Overload



Solution: Buffering and Congestion Control

- Short bursts: buffer
- What if buffer overflows?
  - Packets dropped
  - Sender adjusts rate until load = resources → "congestion control"

如何处理数据丢失?

Problem: Lost Data

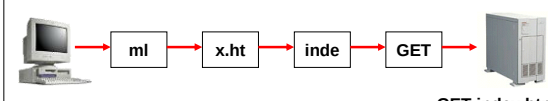
Solution: Timeout and Retransmit

如何处理数据包大小不匹配?

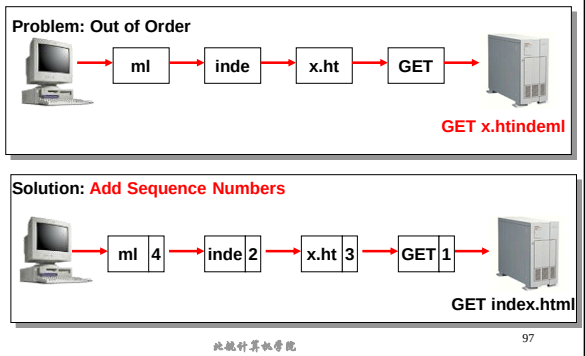
Problem: Packet size

- On Ethernet, max IP packet is 1.5kbytes
- Typical web page is 10kbytes

Solution: Fragment data across packets



如何处理数据乱序?



网络提供的服务

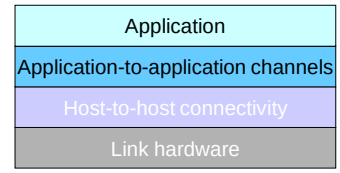
- 可靠性reliability
  - Corruption
  - Lost packets
- 流量控制和拥塞控制
  - Flow and congestion control
- 分段
  - Fragmentation
- 按序投递
  - In-order delivery
- etc...

网络实现的其他功能

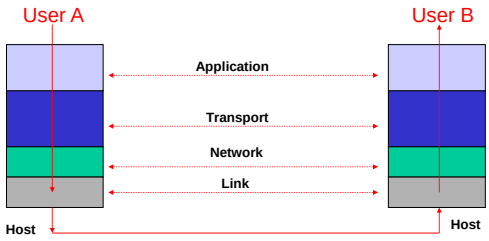
- Link
- Multiplexing
- Routing
- Addressing/naming (locating peers)
- Reliability
- Flow control
- Fragmentation
- Etc....

分层Layering: 模块化方法

- 问题分解
  - Each layer relies on services from layer below：服务service
  - Each layer exports services to layer above: 接口interface
- 层间接口：interaction
  - Hides implementation details
  - Layers can change without disturbing other layers （黑盒子）



分层 Layering

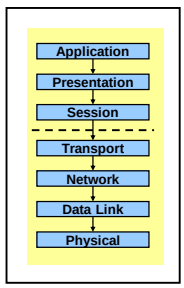


分层：简化复杂系统的方法

例子：OSI 体系结构

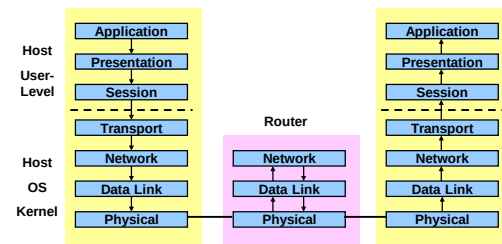
- Open Systems Interconnect (OSI) Architecture
  - 标准化组织：International Standards Organization (ISO)
  - International Telecommunications Union (ITU, formerly CCITT)
  - "X dot" series: X.25, X.400, X.500
  - Primarily a reference model

OSI Protocol Stack



- Application: Application specific protocols
- Presentation: Format of exchanged data
- Session: Name space for connection mgmt
- Transport: Process-to-process channel
- Network: Host-to-host packet delivery
- Data Link: Framing of data bits
- Physical: Transmission of raw bits

OSI Protocol Stack



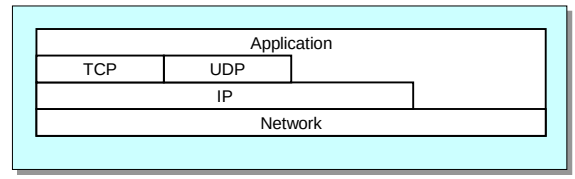
例子：Internet 体系结构

Internet Architecture (TCP/IP)

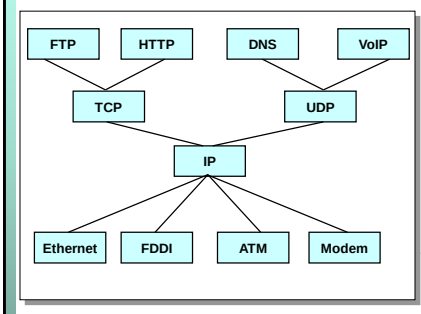
- 起源：Developed with ARPANET and NSFNET
- 标准化组织：Internet Engineering Task Force (IETF)
  - Internet Culture: implement, then standardize
  - OSI culture: standardize, then implement
- 实现：Popular with release of Berkeley Software Distribution (BSD) Unix; i.e., frees software
- RFC文本：Standard suggestions debated publicly through "requests for comments" (RFC's)

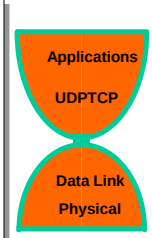
Internet 体系结构：特点

- No strict layering
- Hourglass shape – IP is the focal point



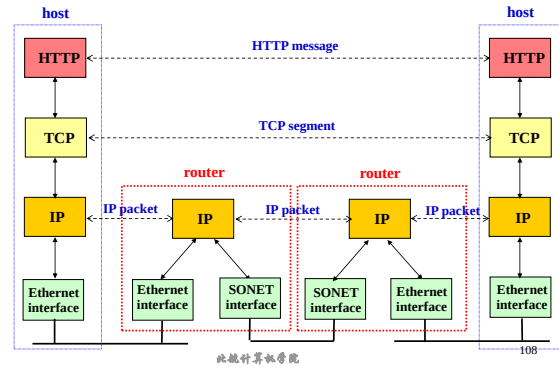
Internet 体系结构：沙漏设计



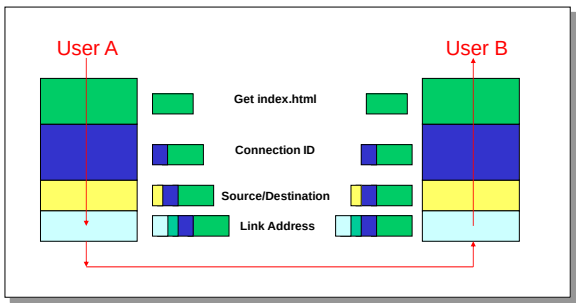


The Hourglass Model

Internet: End Hosts vs. Routers



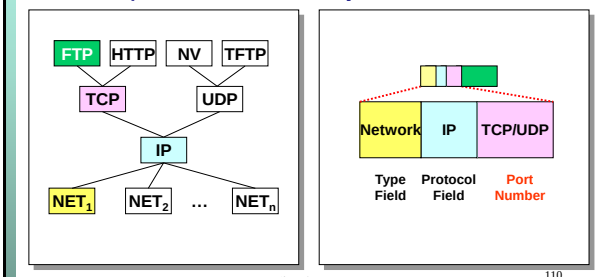
封装：Layer Encapsulation



Protocol Demultiplexing

协议多路解复用

- Multiple choices at each layer



分层设计需要考虑的问题

- 寻址 *addressing*
- 数据传输方式
  - simplex, half-duplex, full-duplex
- 差错控制 *Error control*
  - detection , correction
- 顺序投递 *Ordered delivery — sequencing*
- 分段和重组 *Fragmentation and reassembly*
- 流量控制 *Flow control*
- 多路复用和分解 *Multiplexing and demultiplexing*
- 路由 *Routing*

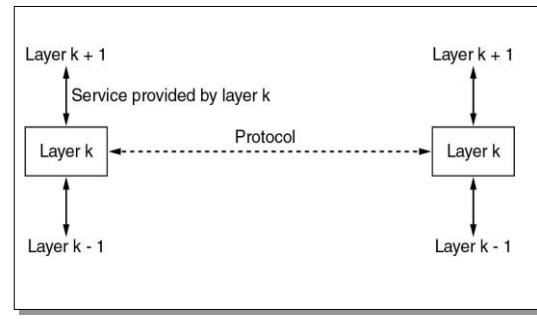
分层的副作用

- 冗余操作
  - 高层复制底层的功能
    - E.g., error recovery to retransmit lost data
- 冗余信息
  - 需要同样的信息
    - E.g., timestamps, maximum transmission unit size
- 降低性能
  - E.g., hiding details about what is really going on
- 庞大头部
  - Sometimes more header bytes than actual content

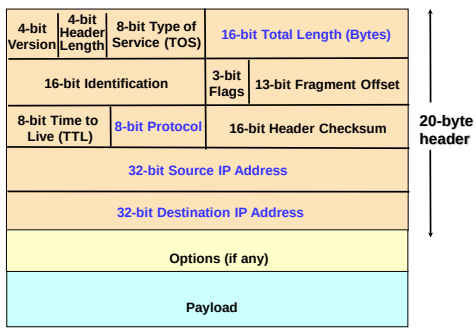
协议Protocols

- 协议 Protocol – how is the service implemented
  - 语法、语义、时序：
    - a set of rules and formats that govern the communication between two peers
  - 实现：
    - Module in layered structure
- 协议定义了:
  - Interface to higher layers (API)
  - Interface to peer
    - Format and order of messages
    - Actions taken on receipt of a message

协议的层间交互



协议实例: IPv4 Packet

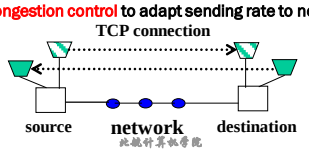


协议实例：IP协议

- 分组交换 Packet switching
  - 将数据封装成分组
  - 分组头部：source & destination address
- 尽力传输 Best-effort delivery
  - Packets may be lost
  - Packets may be corrupted
  - Packets may be delivered out of order

协议实例: TCP

- 通信服务类型 (socket)
  - Ordered, reliable byte stream
  - Simultaneous transmission in both directions
- 主机端的处理 （Key mechanisms at end hosts）
  - Retransmit lost and corrupted packets
  - Discard duplicate packets and put packets in order
  - Flow control to avoid overloading the receiver buffer
  - Congestion control to adapt sending rate to network load

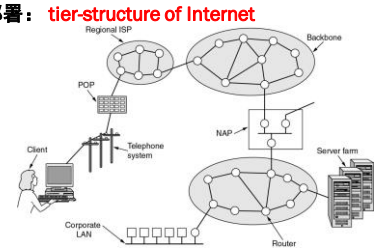


有关标准化组织

- IETF- The Internet Engineering Task Force
- ITU-TS - Telecommunications Sector of the International Telecommunications Union.
  - government representatives (PTTs/State Department)
  - responsible for international "recommendations"
- IEEE - Institute of Electrical and Electronics Engineers.
  - responsible for many physical layer and datalink layer standards
- ISO - International Standards Organization.
  - covers a broad area

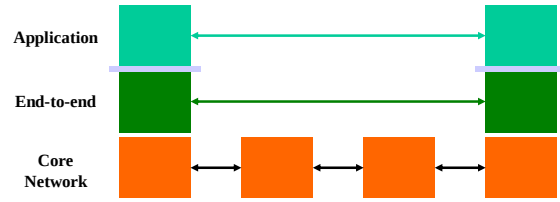
计算机网络的体系结构

- Topological view
  - 组织：organizational structure
  - 部署： tier-structure of Internet



计算机网络的体系结构

- Service model view
  - 分层：layered structure of network
  - 抽象：abstraction of implementation complexities
  - 模块化：modularization of technologies



网络性能参数

带宽

带宽 Bandwidth

Amount of data transmitted per unit of time; per link, or end-to-end

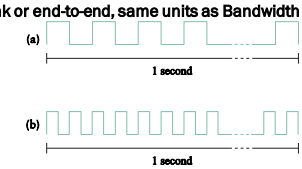
Units 1KB = 2^{10}（或10^3）bytes, 1Mbps = 10^6 bits per sec

How many KB/sec is a 1Mbps line? How many MB/sec?

吞吐量 Throughput

Data rate delivered by the a link, connection or network

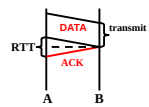
Per link or end-to-end, same units as Bandwidth



时延

◆**时延 Latency/delay**

Per link or end-to-end



- Time from A to B
  - **Example: 30 msec (millliseconds)**
- round-trip time (RTT) 往返时延
  - from one host to another and back again
- Components
  - **传输时间 Transmission time**
  - **传播时延 Propagation delay over links**
  - **排队时延 Queueing delays**
  - **软件处理开销 Software processing overheads**

计算时延

T_t: **传输时延Transmission Delay:**

file size/bandwidth

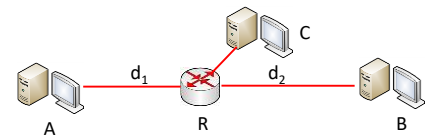
T_p: **传播时延Propagation Delay:**

time needed for signal to travel the medium,

Distance / speed of medium

T_q: **排队时延Queuing Delay:**

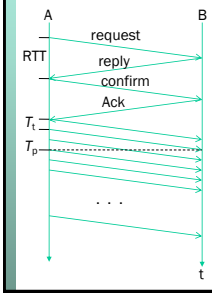
time waiting in router's buffer



例1

传输1.5 MB 文件，若 RTT为80 ms, 分组大小为1KB，初始“握手”延迟为2xRTT，带宽为10 Mbps.

数据可以被连续发送。计算传输文件所需的时间（最后1个bit 到达目的地）。



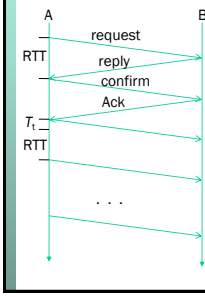
RTT = 80 ms
$T_t = 10^3 x 8 \text{ bits}/10^7 \text{ bits/s} = 0.8$ ms
$T_p = 40$ ms
# of packets = 1500 (1.5×10^3)

$D = 2xRTT + 1500x T_t + T_p$
$= 160 + 1200 + 40$ ms
$= 1.4$ s

例2

传输1.5 MB 文件，若 RTT为80 ms, 分组大小为1KB，初始“握手”延迟为 2xRTT，带宽为10 Mbps.

每发送一个分组需等待一个RTT。计算传输文件所需的时间。



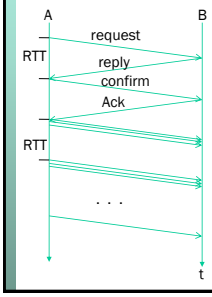
RTT = 80 ms
$T_t = 10^3 x 8 \text{ bits}/10^7 \text{ bits/s} = 0.8$ ms
$T_p = 40$ ms
# of packets = 1500 (1.5×10^3)

$D = 2xRTT + 1499x(T_t + RTT) + T_t + T_p$
$= 160 + 121119.2 + 0.8 + 40$ ms
$= 121.32$ s

例3

传输1.5 MB 文件，若 RTT为80 ms, 分组大小为1KB，初始“握手”延迟为 2xRTT。链路允许无限快速发送。

但每个RTT仅发送20个分组。计算传输文件所需的时间。



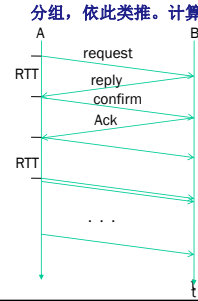
RTT = 80 ms
$T_t = 0$ ms
$T_p = 40$ ms
# of packets = 1500 (1.5×10^3)

$D = 2xRTT + （75-1）RTT + T_p$
$= 160 + 74x80 + 40$
$= 6.12$ s

例4

传输1.5 MB 文件，若 RTT为80 ms, 分组大小为1KB，初始“握手”延迟为 2xRTT。链路允许无限快速发送。

第1个RTT发送1个分组，第2个RTT发送2个分组，第3个RTT发送4个分组，依此类推。计算传输文件所需的时间。



RTT = 80 ms
$T_t = 0$ ms
$T_p = 40$ ms
# of packets = 1500 (1.5×10^3)

# of waits $(1+2+...2^{n-1} = 2^n - 1)$
$2^{11} - 1 = 2047$ packets, n = 11,
第11个RTT中，最后一个分组在T_p后到达
$D = 2xRTT + 10xRTT + T_p$
$= 160 + 800 + 40$ ms
$= 1$ s

延迟与带宽

与应用相关：

不同网络：表示为 RTT/Bandwidth

（1）传输小文件

1 byte file, 1ms/1Mbps vs. 100ms/100Mbps

需要时间：1 ms + 8μs = 1.008ms,

100ms + 0.08μs =100 ms. （传输时延可忽略不计）

（2）传输大文件

1GB file, 1ms/1Mbps vs. 100ms/100Mbps

1ms + 10^9 x 8 /10^6 = 1ms+ 2.2h ,

100ms + 80 s （传输时延为主）

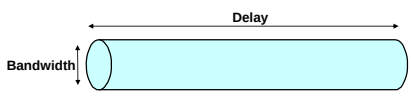
参考数据

- 光速 Speed of Light
  - 3.0 x 10^8 meters/second in a vacuum
  - 2.3 x 10^8 meters/second in a cable
  - 2.0 x 10^8 meters/second in a fiber
- 说明
  - 在直接连接的链路上没有排队时延
  - 传输小数据时，受带宽影响小
  - 短距离传输时，软件开销的影响显著
- 问题：影响性能主要因素（传播时延 vs. 带宽）？
  - Latency dominates small transmissions
  - Bandwidth dominates large

带宽时延乘积

Delay x Bandwidth Product

- channel = pipe
- delay = length
- bandwidth = area of a cross section
- bandwidth x delay product = volume



Delay x Bandwidth Product

- 例子: 长距离传输Transcontinental Channel
  - 带宽 BW = 45 Mbps
  - 时延 delay = 50ms
  - bandwidth x delay product

    = (45 x 10^6 bits/sec) x (50 x 10^{-3} sec)

    = 2.25 x 10^6 bits
- 意义
  - 若发送方保持管道满负载，在第一个bit到达接收方之前，发送方能够传输的bit数。
  - 接收方应答：单向时延

高速网络

Link Type	Bandwidth	Distance	RTT	Delay x BW
Dial-up	56 kbps	10 km	87 μs	5 bits
Wireless LAN	54 Mbps	50 m	0.33 μs	18 bits
Satellite link	45 Mbps	35,000 km	230 ms	10 Mb
Cross-country fiber	10 Gbps	4,000 km	40 ms	400 Mb

无限带宽Infinite bandwidth

Propagation delay dominates

Throughput = Transfer size/Transfer time

Transfer time = RTT + Transfer size/Bandwidth

1MB file across 1Gbps line with 100ms RTT, Throughput is 74.1 Mbps

网络设计考虑

- 如何考虑功能分离
  - Across protocol layers
  - Across network nodes
- 两篇经典论文（课外阅读）
  - [SRC84] End-to-end Arguments in System Design
  - [Cla88] Design Philosophy of the DARPA Internet Protocols

如何阅读论文

S. Keshav., How to read a paper. *SIGCOMM Comput. Commun. Rev.* 37, 3 (July 2007), 83-84

Keshav′ s Three-Pass Approach: Step 1

- A ten-minute scan to get the general idea
  - Title, abstract, and introduction
  - Section and subsection titles
  - Conclusion
  - Bibliography
- What to learn: the five C′ s
  - Category: What type of paper is it?
  - Context: What body of work does it relate to?
  - Correctness: Do the assumptions seem valid?
  - Contributions: What are the main research contributions?
  - Clarity: Is the paper well-written?
- Decide whether to read further...

Keshav's Three-Pass Approach: Step 2

- A more careful, one-hour reading
  - Read with greater care, but ignore details like proofs
  - Figures, diagrams, and illustrations
  - Mark relevant references for later reading
- Grasp the content of the paper
  - Be able to summarize the main idea
  - Identify whether you can (or should) fully understand
- Decide whether to
  - Abandon reading in greater depth
  - Read background material before proceeding further
  - Persevere and continue for a third pass

Keshav's Three-Pass Approach: Step 3

- Several-hour virtual re-implementation of the work
  - Making the same assumptions, recreate the work
  - Identify the paper's innovations and its failings
  - Identify and challenge every assumption
  - Think how you would present the ideas yourself
  - Jot down ideas for future work
- When should you read this carefully?
  - Reviewing for a conference or journal
  - Giving colleagues feedback on a paper
  - Understanding a paper closely related to your research
  - Deeply understanding a classic paper in the field

Other Tips for Reading Papers

- Read at the right level for what you need
  - "Work smarter, not harder"
- Read at the right time of day
  - When you are fresh, not sleepy
- Read in the right place
  - Where you are not distracted, and have enough time
- Read actively
  - With a purpose (what is your goal?)
  - With a pen or computer to take notes
- Read critically
  - Think, question, challenge, critique, …