服务质量（QoS）研究工作案例

孙毅
sunyi@ict.ac.cn

提纲

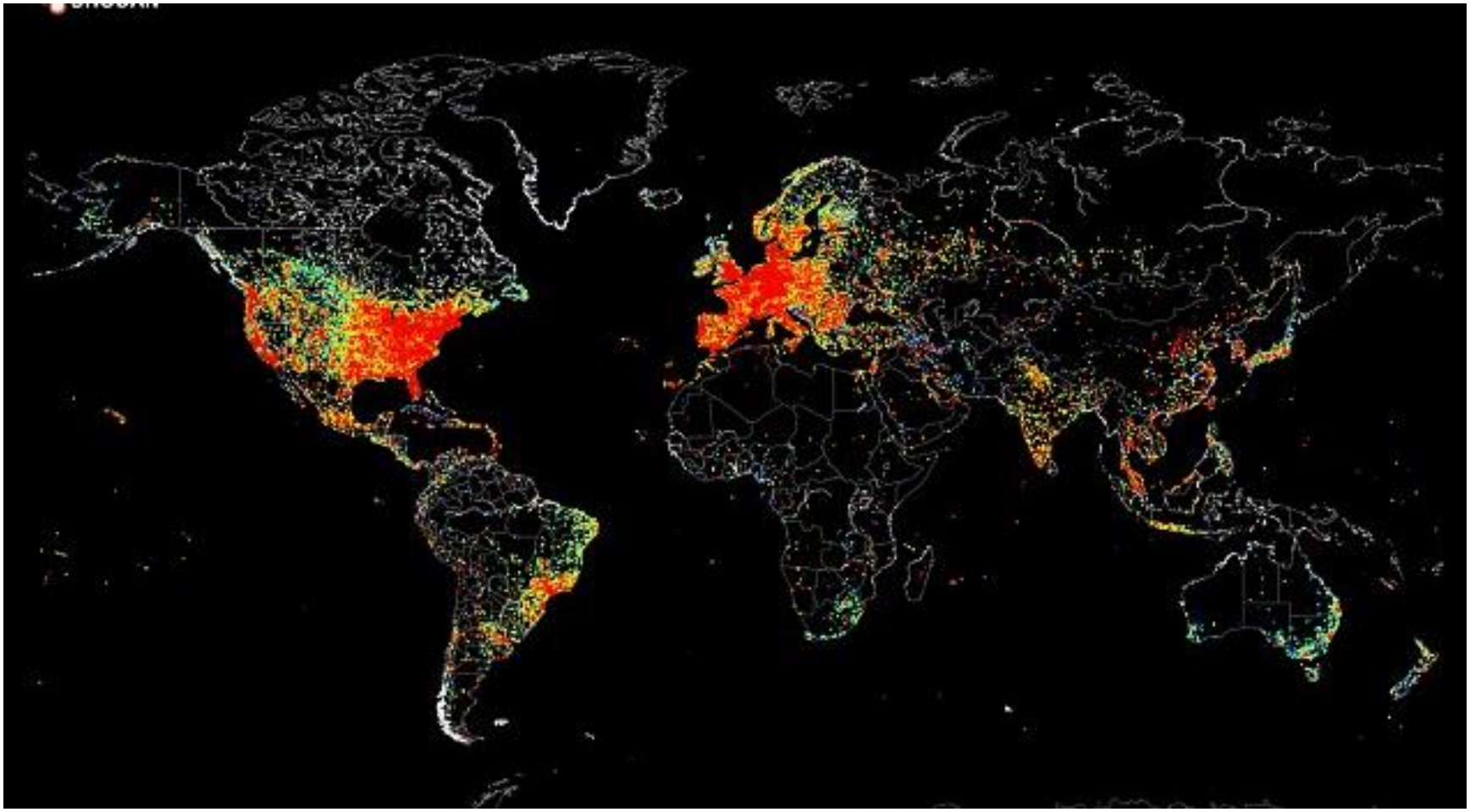
- 网络化缓存与视频服务质量相互影响分析
- 基于网络感知的应用层流量优化
- 数据中心网络服务质量保障方法
  - 路径划分
  - 多路径资源预留

网络化缓存与视频服务质量相互影响分析

视频已成为互联网中最重要的业务

美国：高峰时段带宽占用64%
中国：2/3的用户经常使用互联网看视频

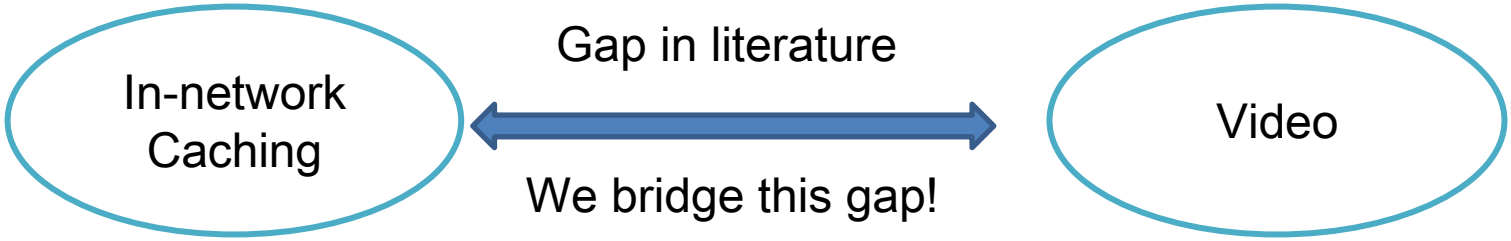
网络缓存是未来互联网的重要特征



网络化缓存与视频传输的相互作用

两个重要问题：

- 视频内容如何影响网络化缓存效率？
- 网络化缓存如何提升视频传输用户体验?



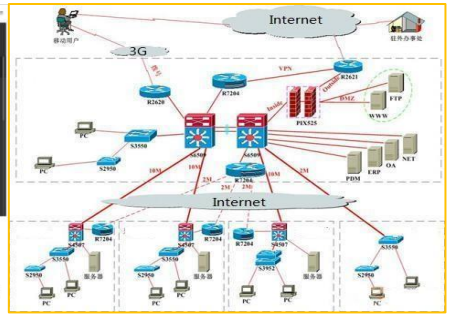
Generic Zipf-like workloads and small-scale topologies

CDN-based delivery architecture

数据集

拓扑数据集
89 ISPs, 80K 路由器, 31省

视频观看数据集
196M 视频观看请求, 16M 用户，500K 影片 137TB



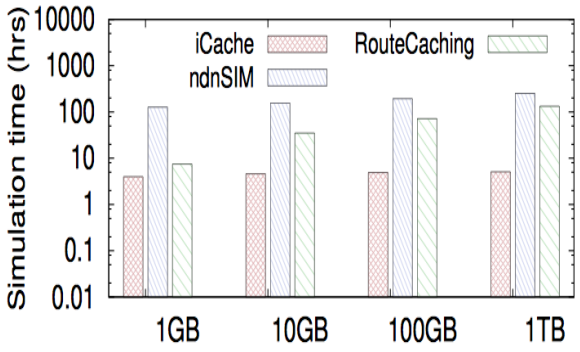
- 35种不同的缓存策略组合（7种放置算法、5种替换算法）
- 4种典型的缓存大小（100MB，1GB，10GB，100GB）
- 总计：35 * 4 = 140 场景

iCache系统

以我们数据集的规模，目前最快的仿真器(NDNSim)需要接近700天！！！

本领域已有工作	拓扑	访问量	仿真器
CMU	三千	百万	流仿真器
巴黎电信	几十	百万	NDN-Sim
ICT	八万	亿	iCache

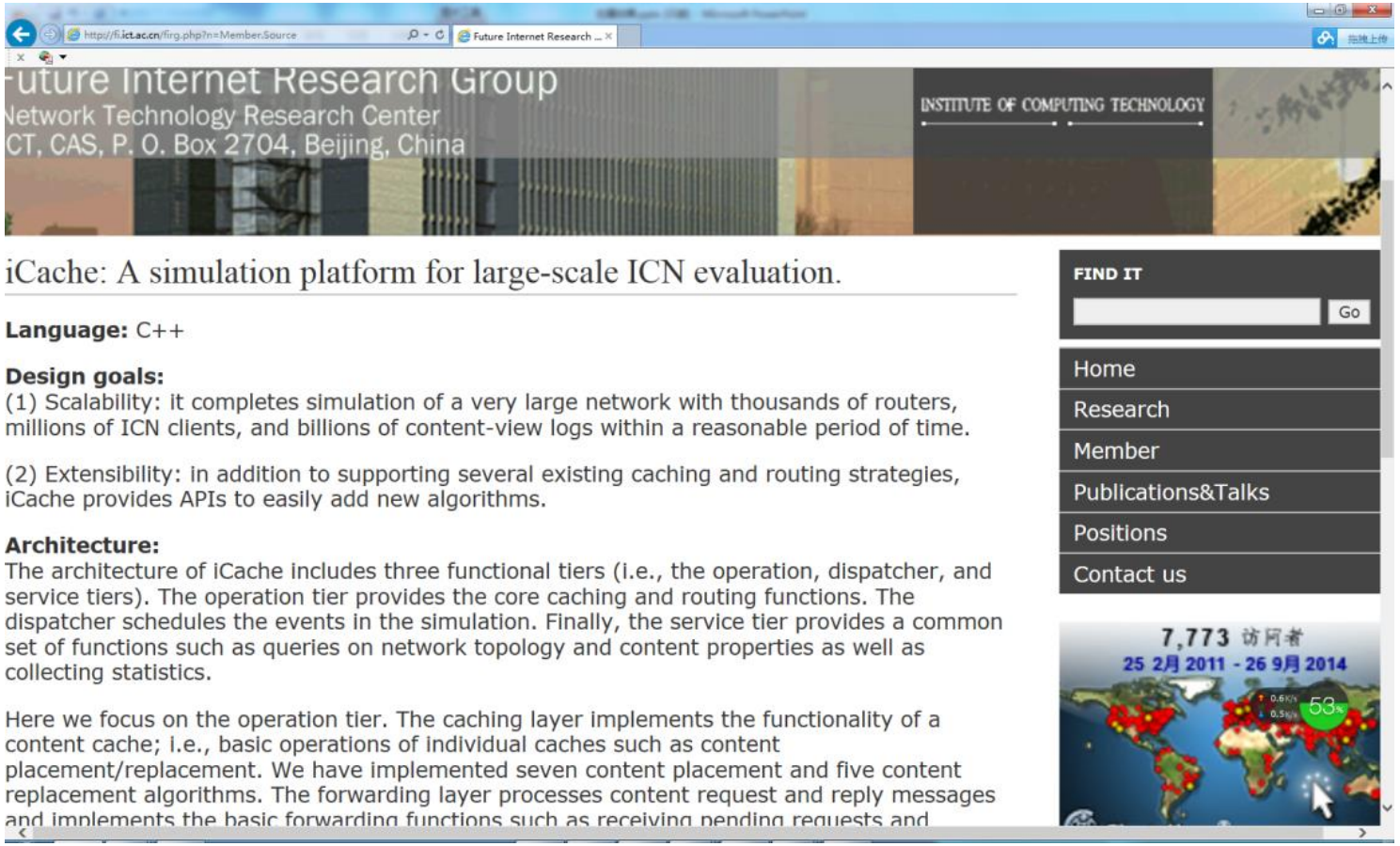
仿真时间小于半个月，50倍以上的加速



工欲善其事，必先利其器！

iCache：高性能ICN缓存服务仿真系统

http://fi.ict.ac.cn/firg.php?n=Member.Source



视频如何影响网络化缓存

- 三个关键问题

1. 视频对于缓存容量的需求?

2. 有无总是最优的缓存策略?

3. 有无近似最优的缓存策略?

缓存命中率

1 GB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	0.003	0.004	0.005	0.004	0.004	0.002	0.004
TTL	0.008	0.007	0.008	0.008	0.008	0.003	0.008
LRU	0.008	0.010	0.006	0.006	0.004	0.004	0.008
LFU	0.007	0.003	0.007	0.007	0.007	0.002	0.007
FIFO	0.007	0.010	0.006	0.006	0.004	0.004	0.007

10 GB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	0.010	0.011	0.014	0.012	0.012	0.007	0.010
TTL	0.020	0.025	0.022	0.020	0.019	0.011	0.020
LRU	0.013	0.023	0.017	0.012	0.014	0.011	0.013
LFU	0.025	0.014	0.025	0.026	0.025	0.011	0.025
FIFO	0.013	0.023	0.016	0.012	0.013	0.008	0.013

100 GB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	0.026	0.029	0.035	0.032	0.034	0.016	0.028
TTL	0.036	0.048	0.046	0.041	0.041	0.023	0.038
LRU	0.034	0.047	0.048	0.043	0.046	0.023	0.038
LFU	0.059	0.039	0.058	0.059	0.058	0.030	0.059
FIFO	0.032	0.046	0.045	0.039	0.040	0.020	0.035

1 TB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	0.065	0.070	0.079	0.076	0.082	0.030	0.072
TTL	0.079	0.090	0.091	0.088	0.088	0.043	0.085
LRU	0.084	0.091	0.096	0.095	0.097	0.043	0.091
LFU	0.111	0.085	0.104	0.108	0.103	0.049	0.111
FIFO	0.078	0.089	0.091	0.088	0.088	0.040	0.085

Hit Rates are Low, Very Low!

A sub-linear improvement as function of the cache size.

网络流量减少 %

1 GB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	1.8	2.1	2.5	2.2	2.1	1.3	1.9
TTL	3.7	3.5	3.4	3.3	3.2	1.4	3.6
LRU	2.6	3.6	2.3	2.3	1.9	2.0	2.5
LFU	3.0	1.7	3.3	3.3	3.2	1.3	3.0
FIFO	2.5	3.6	2.3	2.3	1.8	1.8	2.5

10 GB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	5.5	4.9	6.7	6.1	5.8	3.7	5.7
TTL	8.6	10.1	9.0	8.4	7.7	5.2	8.3
LRU	5.7	9.6	7.6	5.9	6.4	4.5	5.8
LFU	10.5	6.4	10.7	10.7	10.2	5.0	10.5
FIFO	5.6	9.3	7.1	5.7	5.6	3.9	5.6

100 GB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	12.8	11.9	14.1	15.5	15.4	7.1	13.3
TTL	15.4	17.4	17.3	16.2	15.5	9.8	15.8
LRU	15.0	17.5	18.1	17.1	17.2	9.6	16.3
LFU	21.9	15.0	20.4	21.3	20.3	11.9	21.8
FIFO	14.1	16.8	17.0	15.7	15.3	8.6	15.1

1 TB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	23.7	22.1	25.0	24.5	25.0	12.2	25.4
TTL	27.2	26.3	27.6	27.8	26.9	15.4	28.5
LRU	28.6	26.7	28.7	29.3	28.8	16.4	30.0
LFU	33.7	25.3	29.9	31.6	29.9	18.5	33.5
FIFO	27.1	26.3	27.6	27.8	26.9	15.4	28.4

Reduction is >20% only with cache > 100 GB

.

服务器负载降低 %

1 GB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	2.3	3.1	3.5	2.9	3.0	1.7	2.5
TTL	5.6	5.1	5.6	5.5	5.4	1.9	5.4
LRU	5.3	7.1	4.4	4.3	3.1	3.1	5.4
LFU	4.8	2.5	5.1	5.1	5.2	1.7	4.8
FIFO	5.1	7.1	4.4	4.3	3.1	2.6	5.3

10 GB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	6.6	7.3	9.3	8.1	8.1	4.9	7.0
TTL	13.2	16.1	14.4	13.2	12.5	7.4	12.9
LRU	8.8	14.9	11.1	8.4	9.8	7.3	9.1
LFU	16.1	9.4	16.3	16.5	16.5	7.2	16.2
FIFO	8.8	14.7	10.6	8.2	8.6	5.4	8.9

100 GB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	16.6	18.3	21.9	19.8	21.2	10.9	17.6
TTL	21.9	28.8	27.7	24.7	25.0	15.2	22.8
LRU	20.7	28.4	28.5	25.9	27.7	15.2	22.7
LFU	33.7	24.1	33.5	34.1	33.9	18.8	33.8
FIFO	19.5	27.5	26.9	23.9	24.6	13.3	21.3

1 TB

	LCE	LCD	Rand	Prob	PProb	Cent.	Cross
Size	36.3	40.2	43.8	42.2	45.3	19.2	39.3
TTL	42.0	49.1	48.8	47.2	47.5	24.4	44.7
LRU	44.3	49.8	51.1	49.9	51.3	25.9	46.9
LFU	54.5	47.3	54.3	55.2	53.8	29.2	55.0
FIFO	41.9	49.0	48.8	47.1	47.5	24.3	44.6

The reduction is substantial only when cache size is > 100GB

有无近似最优的方案?

For each cache size and metric

$$Score_{pr,s} = \frac{Perf_{pr,s}}{Perf_s^*}$$

Replacement

Placement

	FIFO	*LFU*	*LRU*	*TTL*	*Size*	RowAvg
LCE	0.64	0.92	0.66	0.76	0.48	0.69
LCD	0.86	0.60	0.88	0.85	0.52	0.74
Rand	0.71	0.91	0.74	0.82	0.62	0.76
Prob	0.65	0.93	0.69	0.78	0.57	0.72
PProb	0.62	0.91	0.67	0.76	0.58	0.71
Centrality	0.38	0.43	0.43	0.40	0.30	0.39
Cross	0.66	0.92	0.70	0.77	0.52	0.71
ColAvg	0.65	0.80	0.68	0.73	0.51	

Prob +LFU is quite close to optimal almost always !

网络化缓存如何影响视频传输

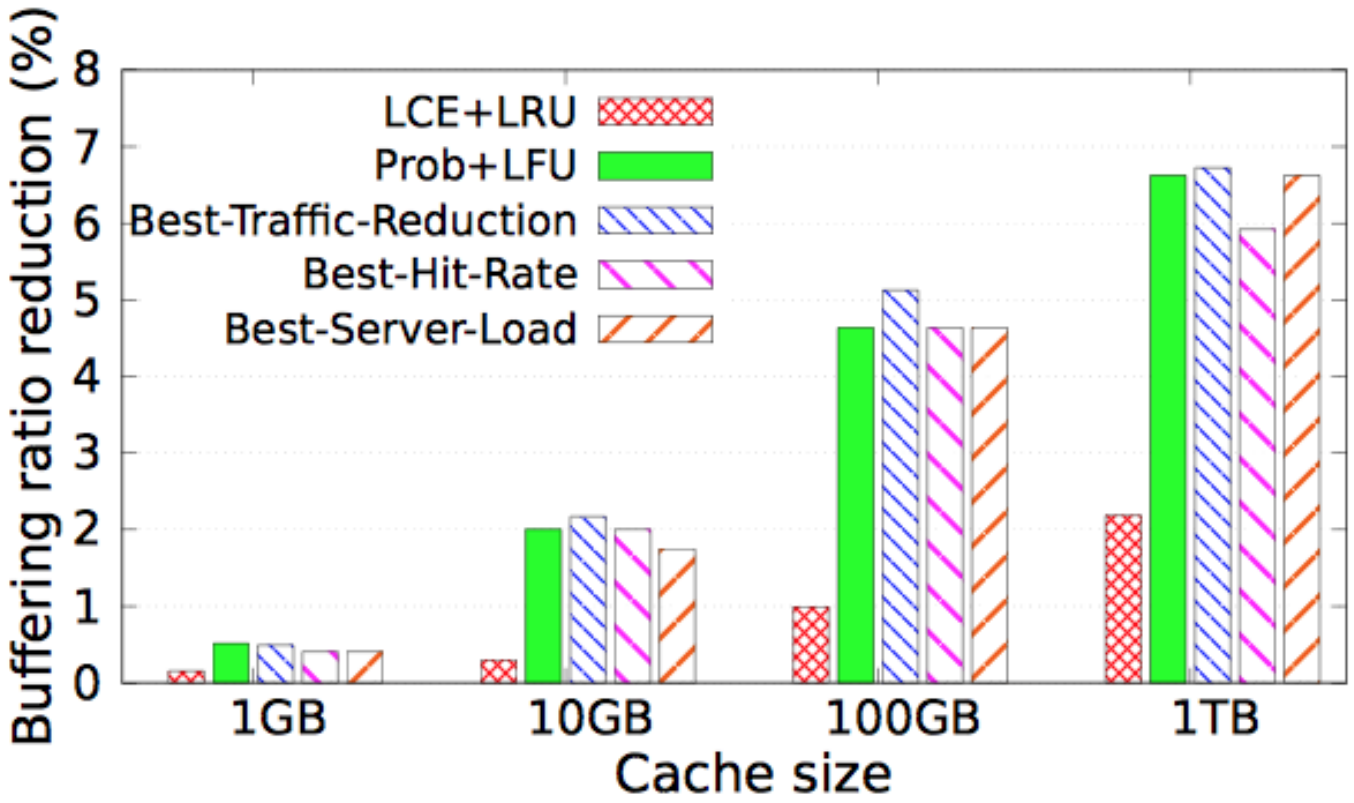
- 三个关键问题

1. 网络化缓存对于提升视频用户体验的作用？

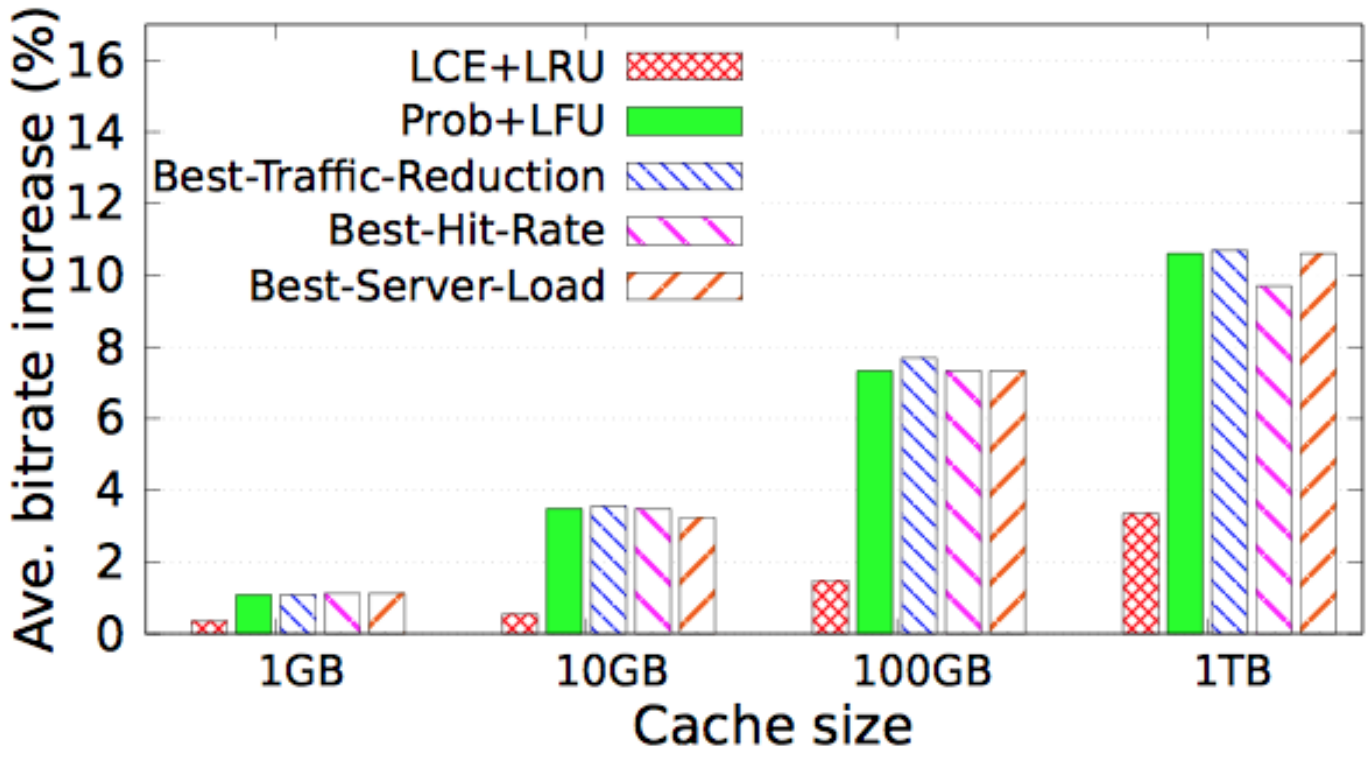
2. 最优的网络化缓存策略对于提升视频用户体验的作用？

3. 什么是最优的缓存策略？

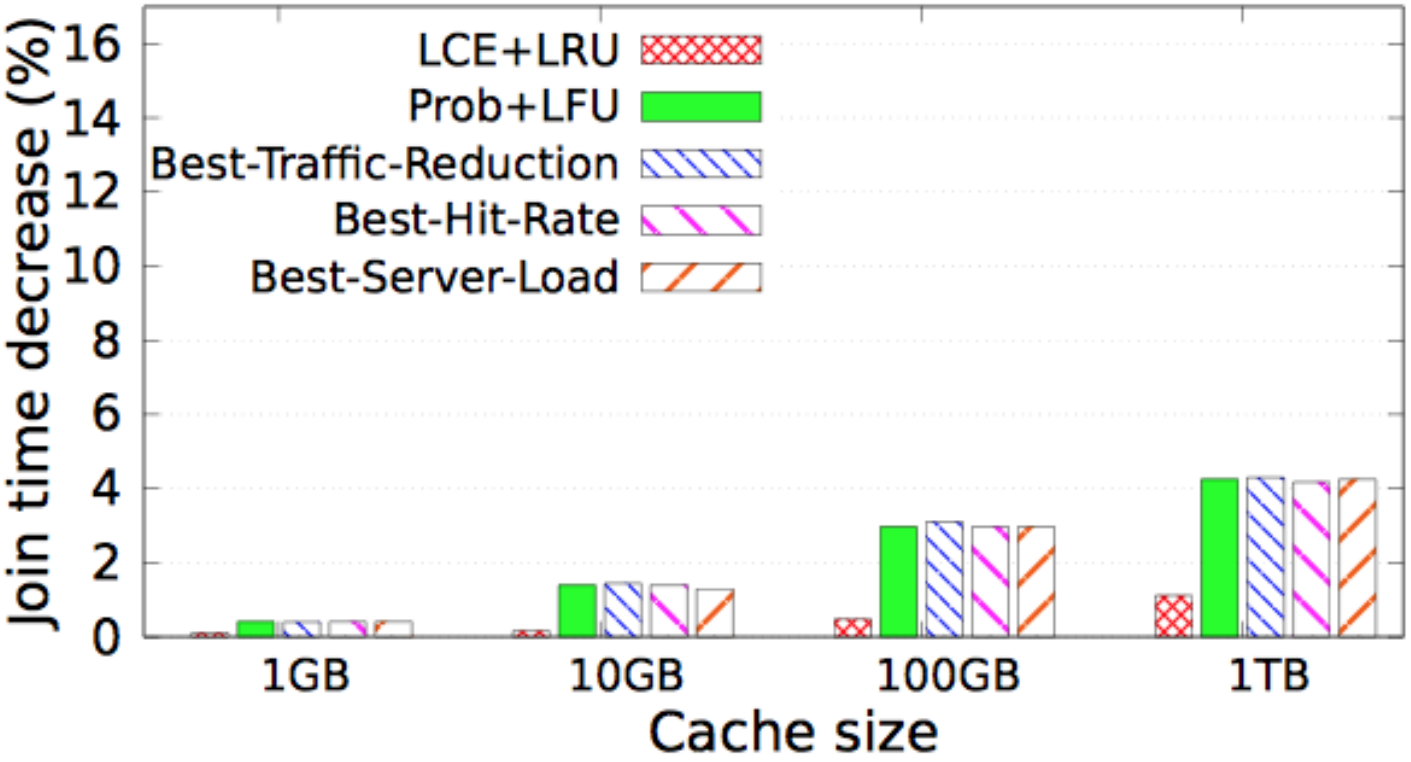
缓冲率


Buffering ratio reduction (%)
0
1
2
3
4
5
6
7
8
LCE+LRU
Prob+LFU
Best-Traffic-Reduction
Best-Hit-Rate
Best-Server-Load
1GB
10GB
100GB
1TB
Cache size

平均下载速度


Ave. bitrate increase (%)
0
2
4
6
8
10
12
14
16
LCE+LRU
Prob+LFU
Best-Traffic-Reduction
Best-Hit-Rate
Best-Server-Load
1GB
10GB
100GB
1TB
Cache size

视频启动时间


Join time decrease (%)
0
2
4
6
8
10
12
14
16
LCE+LRU
Prob+LFU
Best-Traffic-Reduction
Best-Hit-Rate
Best-Server-Load
1GB
10GB
100GB
1TB
Cache size

重要结论：

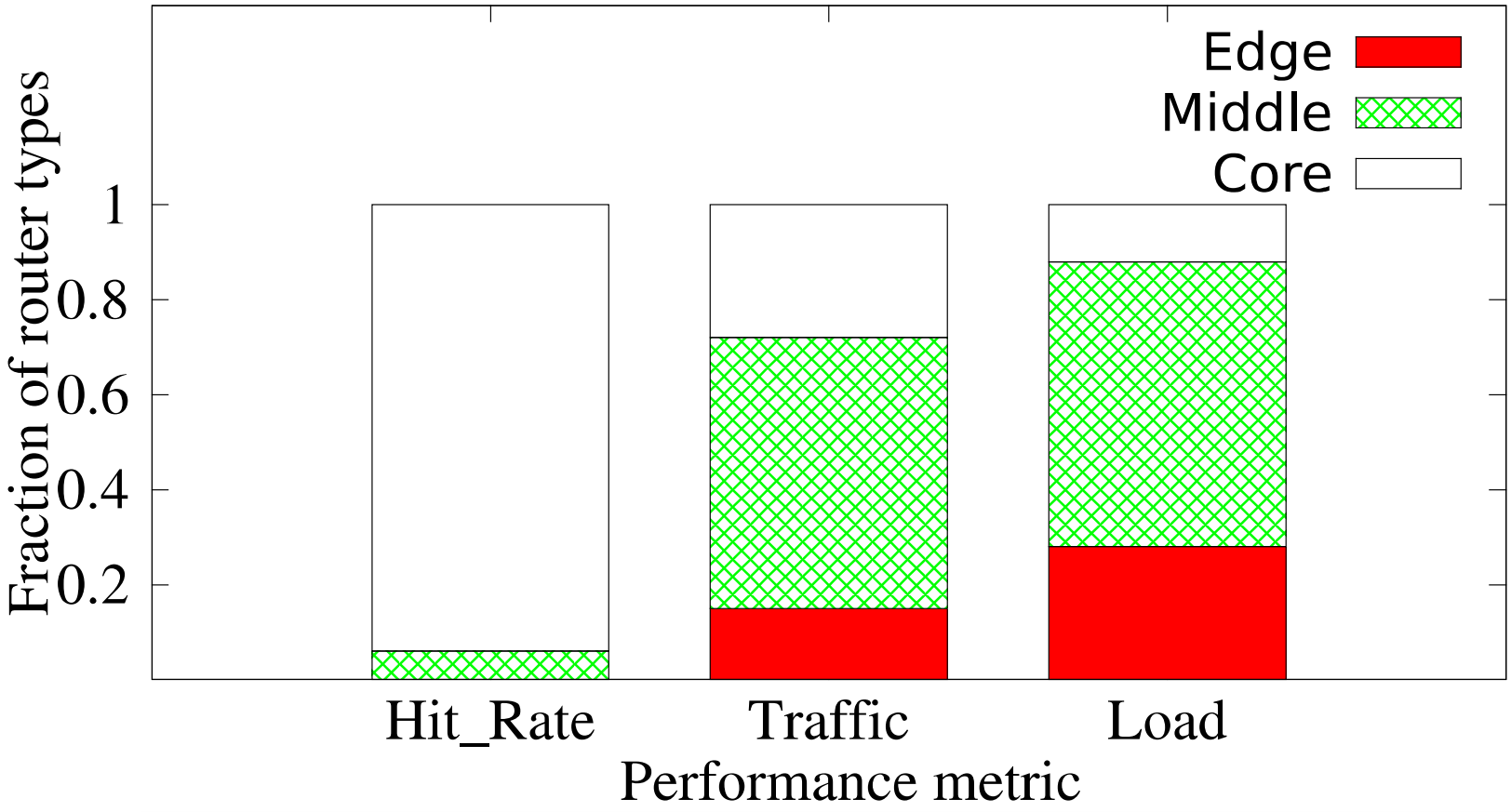
- 视频QoE提升有限（≤12%）
- Optimize traffic reduction = Optimize QoE
  - ISP和视频服务提供商对好策略的看法一致
- Prob+LFU 近似最优

更深入分析

选取100个最优的缓存位置，并分析：

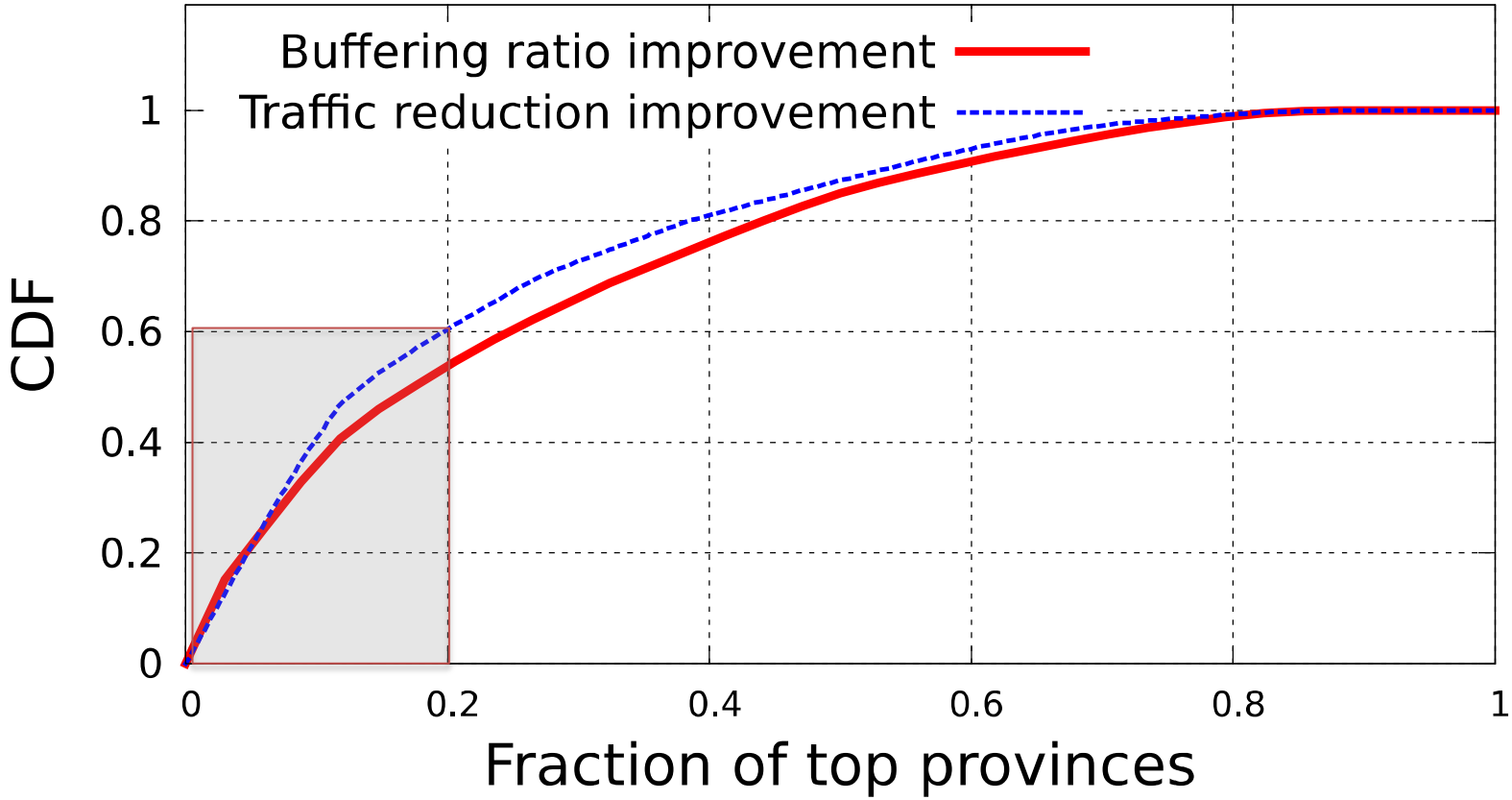
- 网络核心还是汇聚还是接入？
- 地理位置？

网络位置



网络化缓存不适合部署在核心网

地理位置



1. 20 % 的省贡献了60%的改善效果
2. 访问密集、服务器部署少的省份（如：河南）

基于网络感知的应用层流量优化

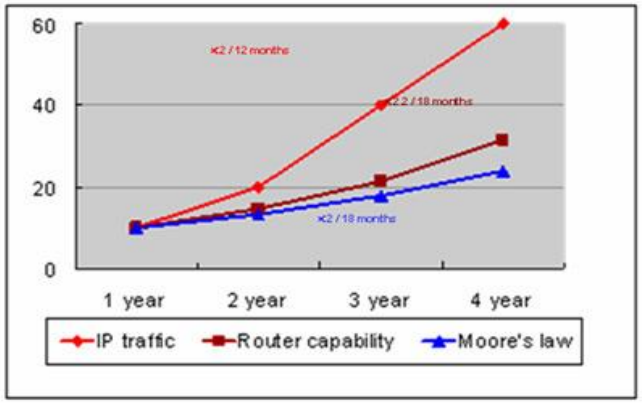
P2P的广泛使用

- P2P技术被广泛用于各种互联网内容分发（音视频、文件下载等）

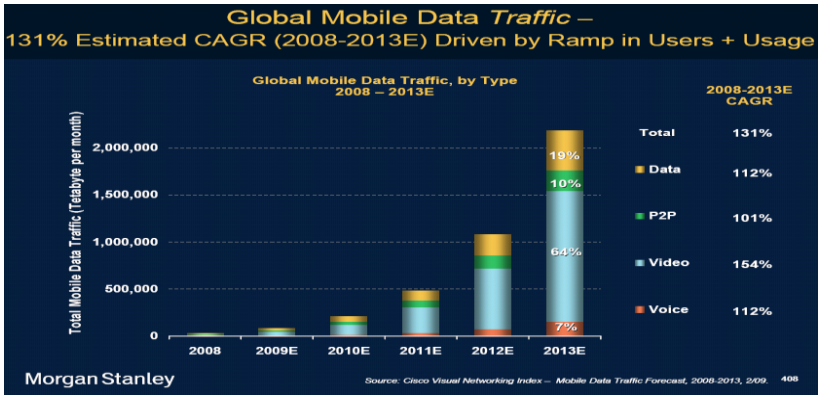

- P2P的优势
  - 去中心话，系统健壮性好
  - 可扩展性好
  - 对服务提供商不收费
  - 对用户多源传输，下载速度更快（QoS）

P2P的问题

- **影响P2P应用推广的一个重要问题就是流量难以管理**
  - 平均传输距离1600公里，5.5个重要网络节点（取自Verizon测试数据，网络运营商的梦魇）
  - 50%~90%本地资源从外部获取



- **资源下载速度不稳定**
  - 节点太远，链路质量差
  - 网络运营商限流、整形



流量本地化

- 通过寻找临近的服务源节点，解决P2P覆盖网络与底层传输网路的失配问题

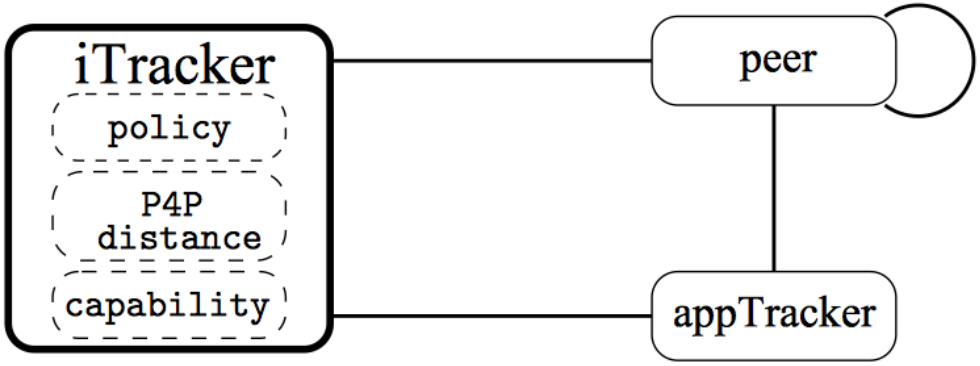
- 基本方案
  - ✓ 应用系统反向工程（探测最优的RTT、网络坐标系统等等），实例：Joost、Kontiki
  - ✓ 缺点：探测不准确、很多信息（网络拓扑、拥塞状况、计费策略等）无法探测

P4P：运营商友好的P2P技术

- 让运营商提供网络状态，指导服务提供商优化服务选择

- 提出者：谢海永博士（华为美研）、杨阳教授（耶鲁大学）

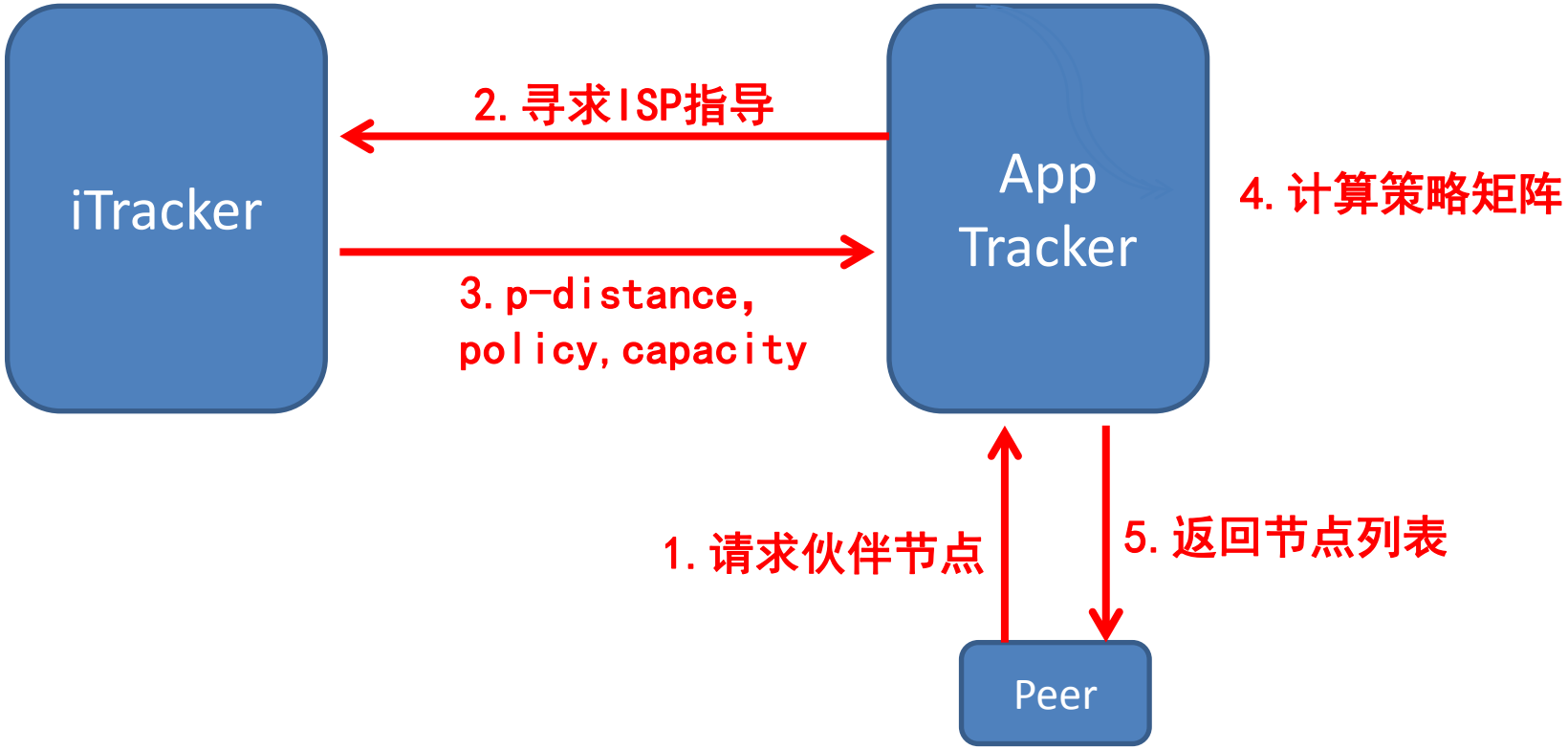
P4P控制架构



引入了iTracker（ISP Tracker）作为网络运营商与应用进行交互的门户，iTracker提供三种形式的网络信息：

- P4P distance：网络状态、拓扑
- Policy：运营商策略，如入口和出口流量比例，拥塞时避免使用的链路等
- Capacity：网络的能力

P4P工作流程



策略矩阵

AN EXAMPLE PGM

	PID1	PID2	PID3	Intra-AS Percentage
PID1	75%	10%	15%	90%
PID2	18%	70%	12%	85%
PID3	10%	10%	80%	90%

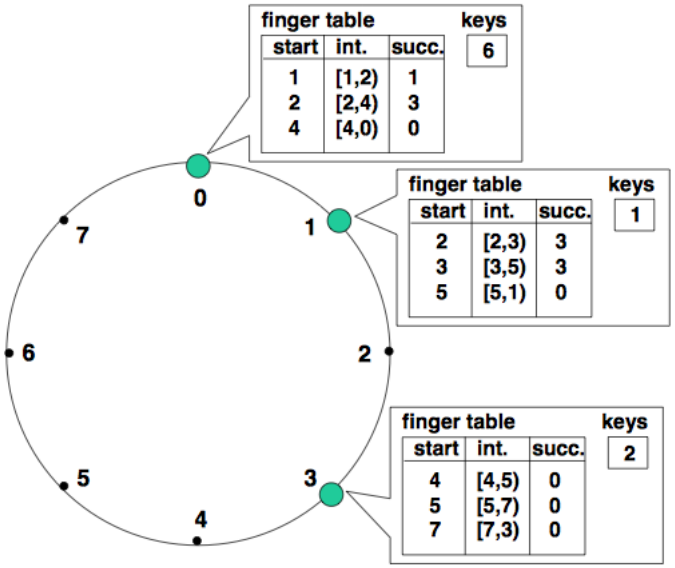
A RP from PID 1 should select:

1) ***90%*75%=67.5%*** peers from PID1
2) ***90%*10%=9%*** peers from PID2
3) ***90%*15%=13.5%*** peers from PID3
4) ***1-90%=10%*** peers from other ASes

PID域：一个接入点的子网
AS域：自治系统

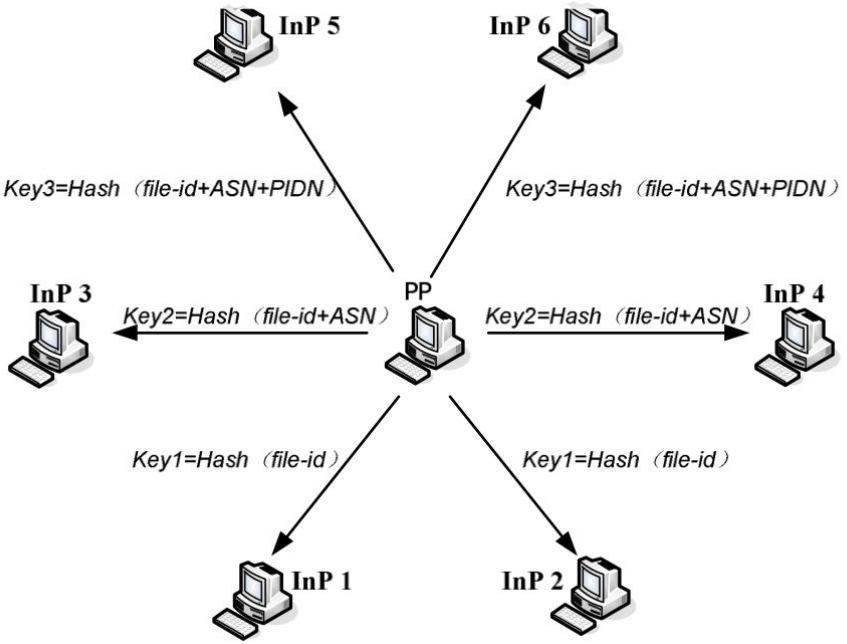
节点可以通过询问iTracker得到自己所属的PID和AS域

DHT网络不存在AppTracker



$$Key1 = Hash(file - id)$$

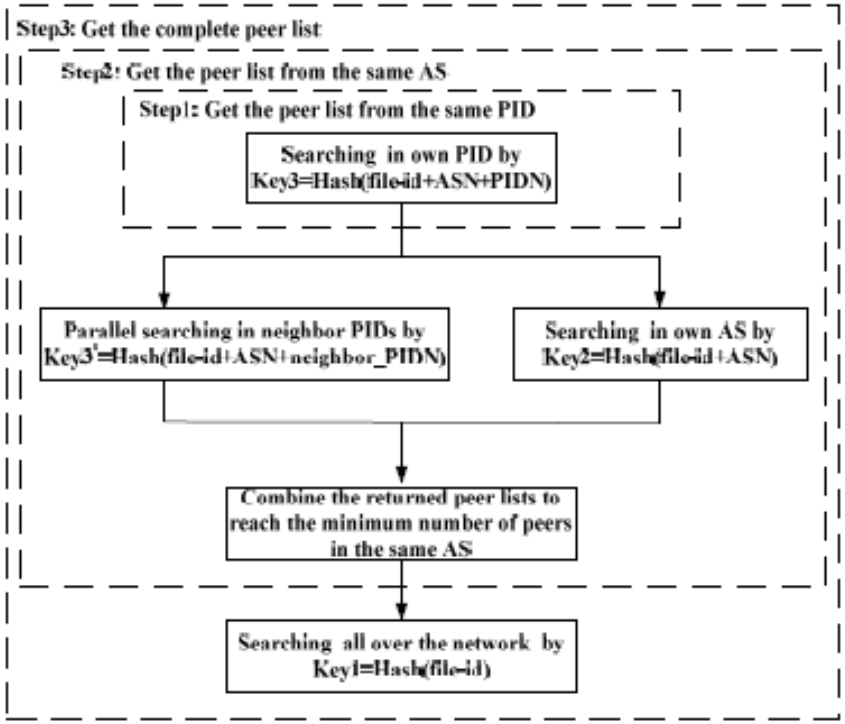
$$Key2 = Hash(file - id + ASN)$$
$$Key3 = Hash(file - id + ASN + PIDN)$$



PP: Publishing Peer
RP: Requesting Peer
InP: Indexing Peer

基于三重哈希的资源查找

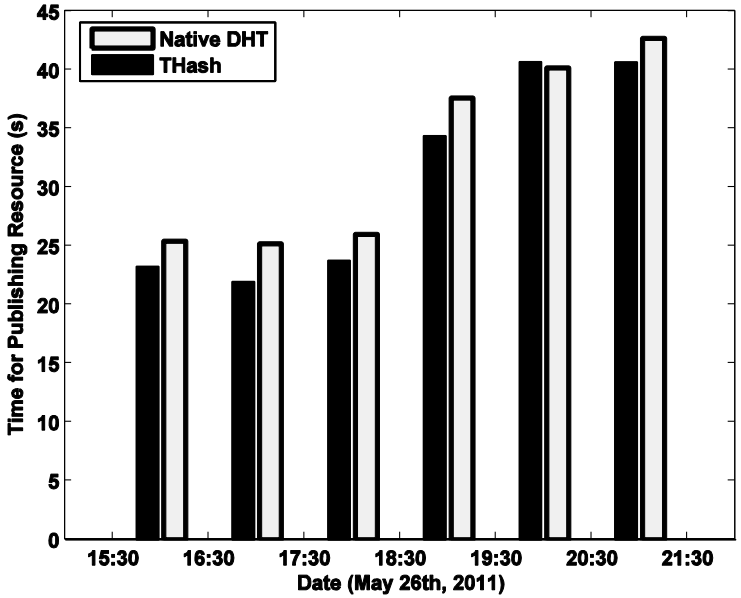
- 并行查找：
  - 自己的PID域
    - Searching by its own *Key3*
  - 同一个AS的邻居PID域
    - Searching by other *Key3** and its own *Key2*
  - 其它AS域
    - Searching by its own *Key1*

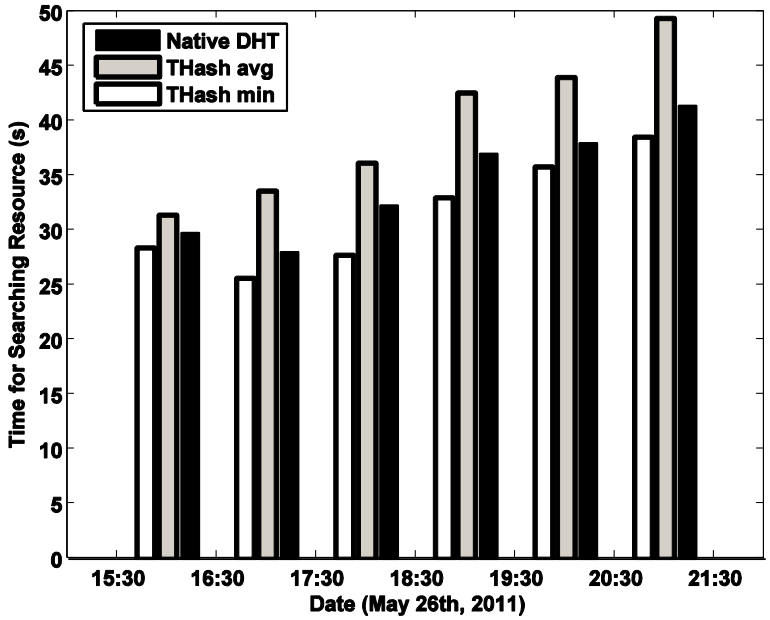


实验

- DHT相关指标
  - 发布时间：Publishing Delay
  - 查找时间：Searching Delay
- 网络优化效（ISP concerned)
  - 同一个域内的节点比例：Intra-domain peer ratio
  - 跨域流量：Inter-domain traffic
- 应用性能：Application performance (User concerned)
  - Downloading performance speedup
- 开销：Overhead
  - 节点重传率：Peer reselection ratio
  - PGM更新开销：PGM update overhead

DHT Related Delays

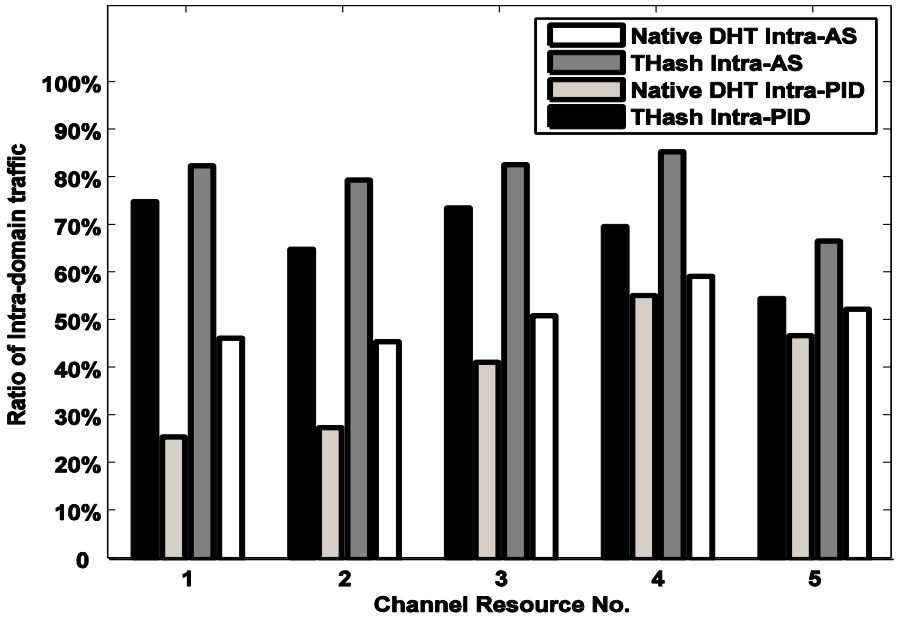


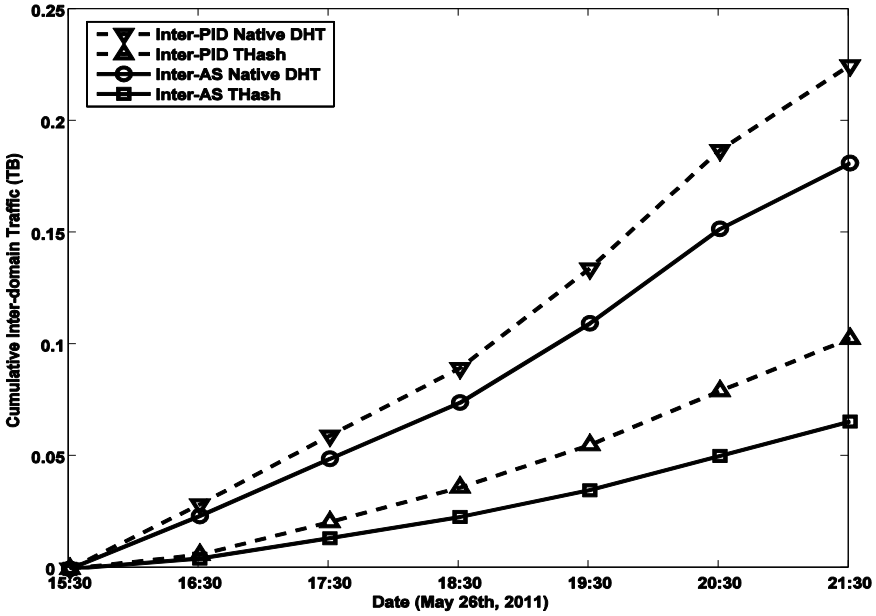


Conclusions:

1) No big difference in the publishing delay between the two schemes
2) THash causes an increase in the average delay for searching resources. However, the delay of the first reply for THash is smaller than the delay of Native DHT.

Network Optimization

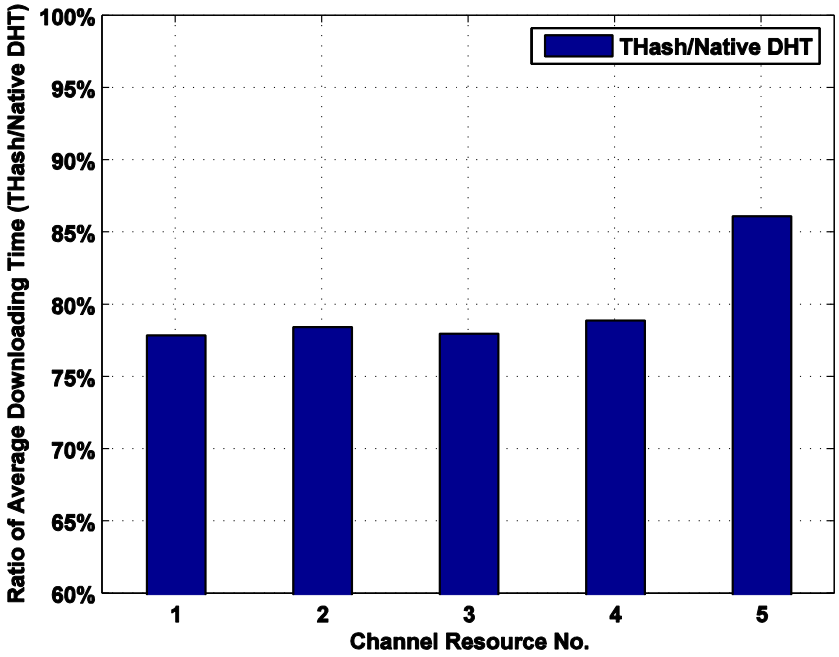




Conclusions:

1. **By introducing the network optimization in THash (PGM) we remarkably limit Inter-AS and Inter-PID traffic.**

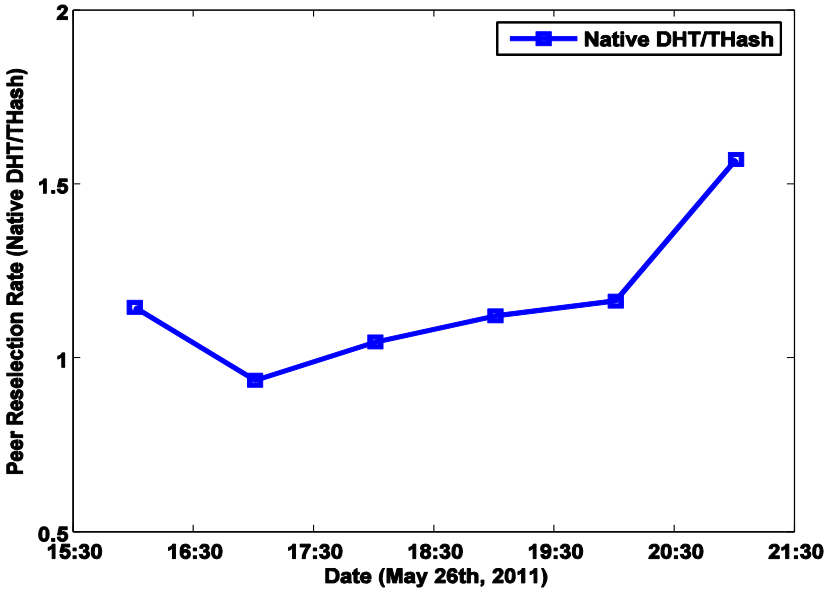
Application Performance

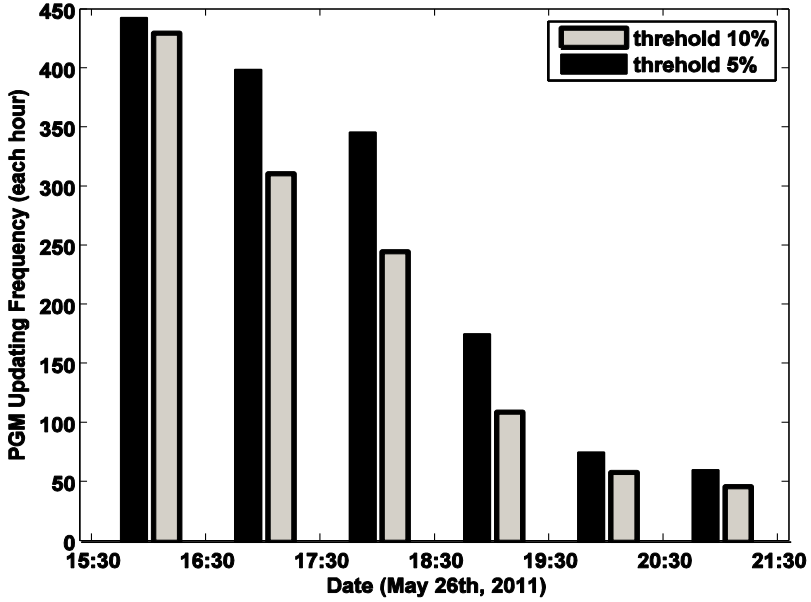


Conclusions:

1. **THash considers not only link utilization in p4p-distance but also load balancing in resource distribution information, thus resulting in higher downloading performance.**

Overhead





Conclusions:

1. In THash the uploading performances of the peers are not badly impacted by the variations of the network conditions, thus reducing the peer reselection rate and resulting in higher stability.
2. When the system enters into a stable state, the overhead of updating and distributing the PGM is negligible compared with the data traffic transmission in the system.

数据中心网络服务质量保障方法

研究背景

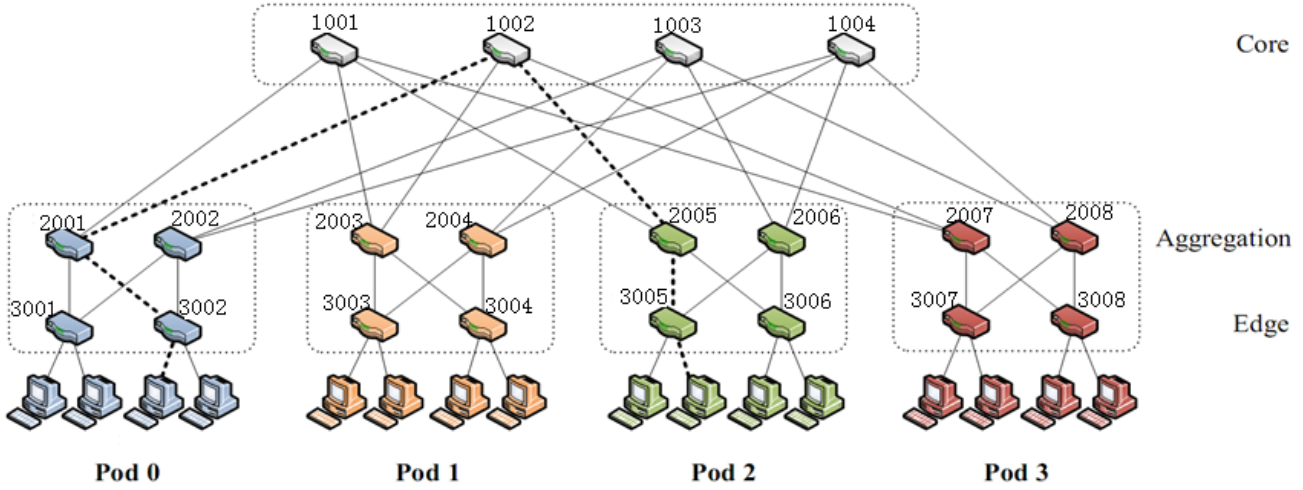
- **云计算是当今IT业的发展热点**
  - 根据CISCO统计，到2016年，全球62%或将近三分之二的总工作负荷将在云端处理[2]
  - 作为云计算的基础设施，云数据中心包括计算资源、网络资源和存储资源三大资源
- **数据中心网络(DCN)是云计算基础设施的核心组成部分，也是云计算的瓶颈所在**[1]
  - 当前的业务模型(如MapReduce，分布式存储，网页搜索等)极大的依赖于节点间的通信

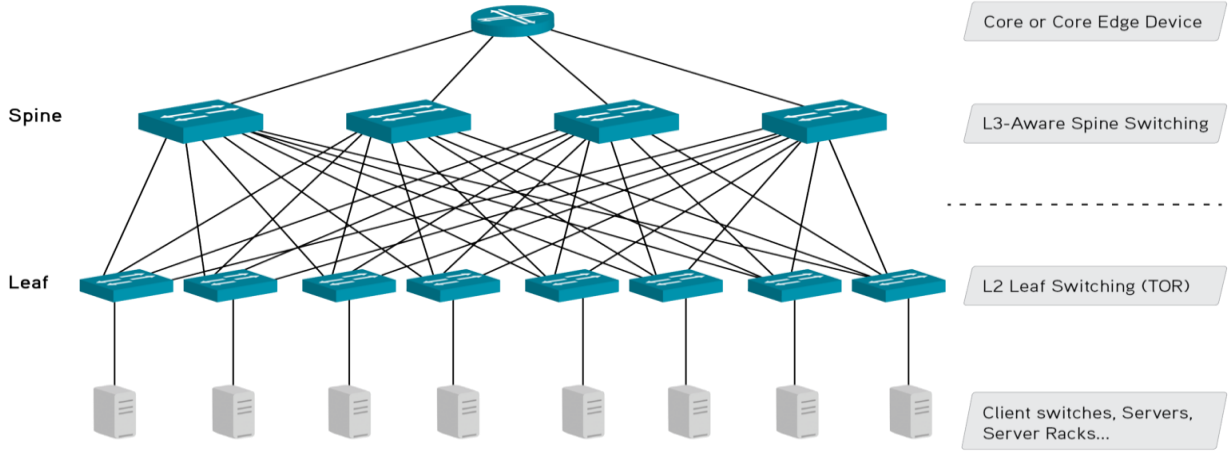
[1] Al-Fares, Mohammad and Loukissas, Alexander and Vahdat, Amin, A scalable commodity datacenter network architecture. SIGCOMM 2008.
[2] Global Cloud Index 2014
[3] 李丹, 陈贵海, 任丰原, 蒋长林, 徐明伟, 数据中心网络研究进展与趋势, 计算机学报, Vol. 37, No. 2, Feb. 2014

DCN

- 数据中心网络，Datacenter Network（DCN）
  - Enterprise DC, Internet DC, Cloud DC
  - 10K~100K PCs, L2 & L3 connection, 1/10Gbps NIC
  - Commercial L2/L3 switch, 40*1G/8*10G
  - Topology: Spline_leaf, Fat-tree,VL2,Bcube
  - Application: web search, big data(MapReduce), recommendation system, social network, ad
  - Multiple equal cost paths between inter-racks
  - 等价路由：ECMP实现负载均衡

研究背景—DCN拓扑


1001
1002
1003
1004
Core
2001
2002
2003
2004
2005
2006
2007
2008
Aggregation
3001
3002
3003
3004
3005
3006
3007
3008
Edge
Pod 0
Pod 1
Pod 2
Pod 3


Core or Core Edge Device
Spine
L3-Aware Spine Switching
Leaf
L2 Leaf Switching (TOR)
Client switches, Servers,
Server Racks...

研究背景—DCN Traffic

- 99.9% TCP connection
- 大流（Elephant Flow）、小流（Short Flow）相结合
- 小流(lasting several RTTs)延时敏感，大流 吞吐量要求高
- 带宽是足够的，只是没有充分利用
- 热点通常发生在核心网

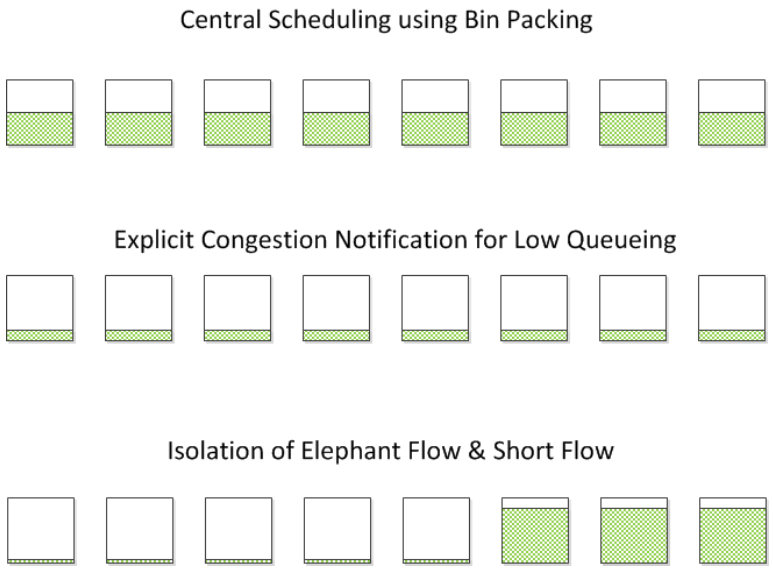
[IMC 2011]

研究背景

- 提供QoS保障是优化DCN的重要途径之一
  - 私有云DCN的QoS保障
    - 保障的对象是**流**，包括大流和小流[4]
    - 小流要求延时小 vs 大流要求吞吐量大[5]
    - 焦点问题之一：保证延时敏感的小流在规定时间内完成，同时提高大流的吞吐量，提供网络利用率—即**同时保障低延时和高吞吐量**
  - 公有云DCN的QoS保障
    - 保障的对象是**租户**
    - 租户的QoS要求各不相同且经常变化
    - 焦点问题之一：通过**带宽保障**的方法，为租户提供QoS保障

同时满足低延时高吞吐量的方法研究

- 研究问题
  - DCN中大流(高吞吐量）和小流（延时敏感）并存，如何同时满足大流的高吞吐量和小流的低延时的QoS要求



研究现状—同时满足低延时和高吞吐量的QoS保障

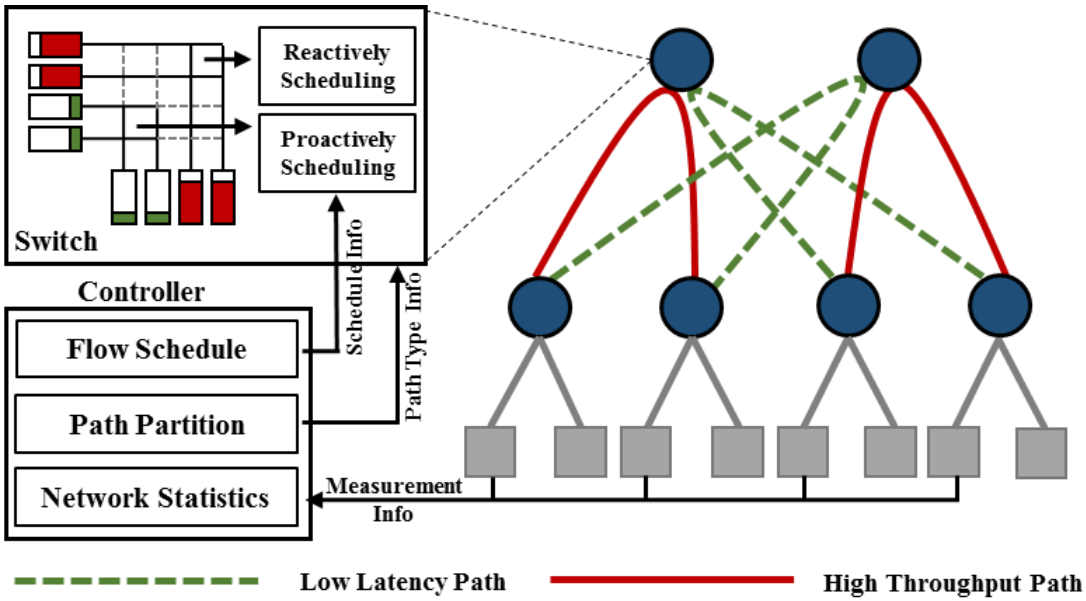
- **基于负载均衡的方法**
  - 文献：[IEEE standard 802.1Qbp, NSDI'10, CONEXT'12]
  - 特征：这类方法主旨是对网络流量进行负载均衡，使网络链路的利用率达到均衡，从而避免网络出现拥堵，从而在总体上降低网络的平均延时，并有效提高了网络的利用率
  - 缺点：无法保障延时敏感的流在规定时间内完成
- **基于拥塞控制和流控的方法**
  - 文献：[SIGCOMM'11, SIGCOM'12, SIGCOM'13, NSDI'12, SIGCOM'14]
  - 特征：这类方法主要利用Active Queue Management (AQM)技术探测网络中的拥堵状况，并利用流控技术控制流速，从而使网络设备保持较低或几乎为0的等待队列，大大降低网络延迟
  - 缺点：网络利用率降低；需要高级的交换机特性支持；部署难度较大

研究现状—同时满足低延时和高吞吐量的QoS保障

- 基于优先级调度的方法
  - 文献：[SIGCOMM'12, SIGCOMM'13]
  - 特征：利用优先级进行流量调度和数据包转发，优先保证延时敏感的流的完成时间
  - 缺点：导致低优先级的流产生“饥饿效应”；需要交换机的高级特性支持
- 小结
  - 现有的方法无法同时提供低延时和高吞吐量的QoS保障

同时满足低延时高吞吐量的方法研究(1/5)

- 研究思路
  - 概括：通过动态路径划分将DCN网络分为高吞吐量路径和低延时路径，并分别将大流和小流在这两种类型的路径上传输和调度。



同时满足低延时高吞吐量的方法研究(2/5)

- 研究思路(续)

①建立路径延时计算模型

- 输入：小流的完成时间要求(如200ms)
- 输出：计算路径的平均延时上限

②设计动态链路划分算法

- P还是NP?
- 近似算法

③设计流调度方法及实现

- 集中式的大流调度，进一步优化链路利用率
- 分布式的小流调度，进一步减小调度延时开销
- 基于VLAN实现

同时满足低延时高吞吐量的方法研究(3/5)

- **路径延时计算模型**

流完成时间(FCT)定义，

$$FCT = K\frac{MSS}{C} + \sum_{i=0}^{l} w_i$$

计算FCT的概率分布

$$\mathbb{P}\{FCT^{-} > y\} = \frac{\gamma(l, \kappa y)}{(l-1)!}$$

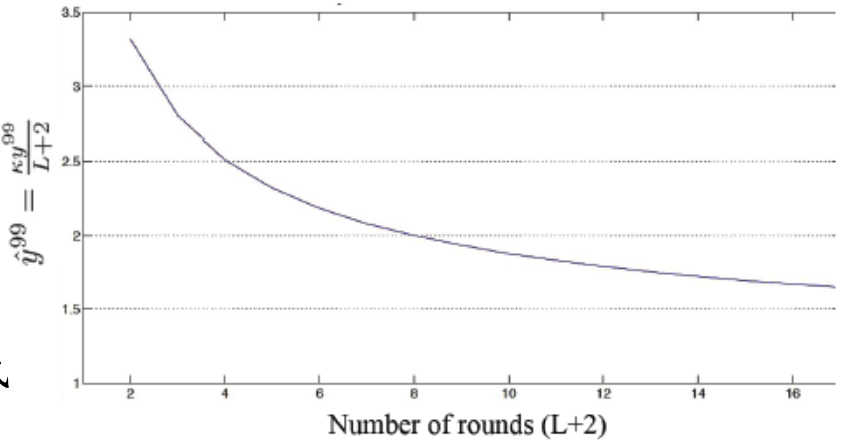
满足99%的流在规定时间内完成

$$\frac{\gamma(L, z)}{(L+1)!} = 0.01$$

$$\hat{y}^{99}(l) = \frac{\kappa y^{99}}{l} \qquad y^{99} = \frac{2.5l}{\kappa}$$

根据M/G/1模型，得到路径时延

$$\mathbb{E}\{W\} = \frac{\lambda}{2(1-\rho)}\frac{\mathbb{E}\{X^2\}}{\mathbb{E}\{X\}} = \frac{1}{\kappa}$$



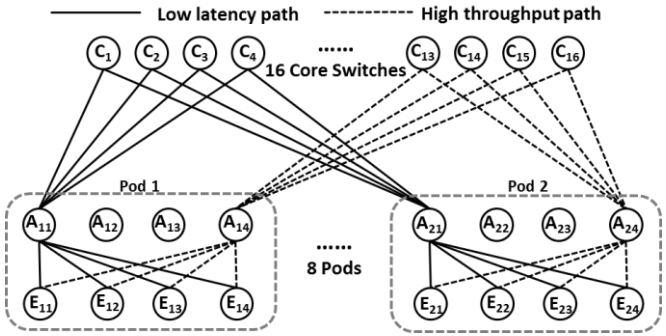
$$\mathbb{E}\{W\} > 2\overline{F_{SLA}^{-}}$$ 延时上限

$$\mathbb{E}\{W\} < 2\epsilon\overline{F_{SLA}^{-}}$$ 延时下限

同时满足低延时高吞吐量的方法研究(4/5)

- 动态路径划分算法
  - 问题定义

GMU问题：根据路径延时的上下界以及实时流量负载，以最小的费用动态增减低延时路径和高吞吐量路径



  - NPC证明

$$\textit{Minimal Cost Dominant Set} \leq_p \textit{GMU}$$

  - 近似算法

1. 基于贪心的近似算法
2. $O(M^2N^2)$

Algorithm 1 Main Control Loop

```
Initialize the fixed LLP and HTP paths
while True do
    D^kT ← getDelays()              ▷ Get delay estimates
    RM ← getRM(D^kT)                ▷ Derive the Request Matrix
    (ℜ, 𝔖) ← DPP(RM)                ▷ Solve the DPP
    notifyNetwork(ℜ, 𝔖)              ▷ Deploy the modified paths
    sleep(T)                        ▷ Wait for next update period
end while
```

Algorithm 2 Dynamic Pat

```
function DPP(RM)
    AR ← The list of
    DR ← The list of
    ℜ ← DeletePaths(I
end function
function DELETEPATHS(DR)
    while DR ≠ ∅ do
        I ← the idle LLP with lowest 𝒩(e) for R_ij in DR
```

算法例子

R_1 --> P_1 = {{e_1,e_2,e_3}, {e_3,e_4,e_6}}

R_2 --> P_2 = {{e_2,e_4,e_6}, {e_1,e_2,e_4}}

R_3 --> P_3 = {{e_1,e_3,e_5}, {e_5,e_6,e_7}}

R_4 --> P_4 = {{e_5,e_6,e_2}}

S = { }

R_1 --> P_1 = {{e_1,e_2,e_3}, {e_3,e_4,e_6}}

R_2 --> P_2 = {{e_2,e_4,e_6}, {e_1,e_2,e_4}}

R_3 --> P_3 = {{e_1,e_3,e_5}, {e_5,e_6,e_7}}

R_4 --> P_4 = {{e_5,e_6,e_2}}

S = {e_1,e_2,e_3 ,e_4 }

R_1 --> P_1 = {{e_1,e_2,e_3}, {e_3,e_4,e_6}}

R_2 --> P_2 = {{e_2,e_4,e_6}, {e_1,e_2,e_4}}

R_3 --> P_3 = {{e_1,e_3,e_5}, {e_5,e_6,e_7}}

R_4 --> P_4 = {{e_5,e_6,e_2}}

S = {e_1,e_2,e_3 ,e_4 ,e_5 }

R_1 --> P_1 = {{e_1,e_2,e_3}, {e_3,e_4,e_6}}

R_2 --> P_2 = {{e_2,e_4,e_6}, {e_1,e_2,e_4}}

R_3 --> P_3 = {{e_1,e_3,e_5}, {e_5,e_6,e_7}}

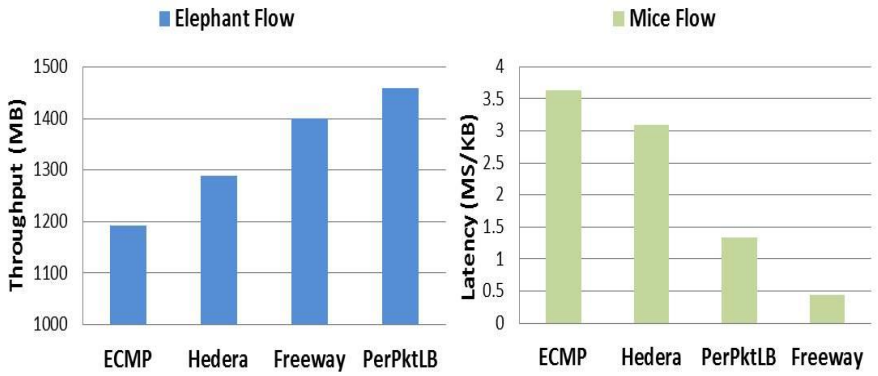
R_4 --> P_4 = {{e_5,e_6,e_2}}

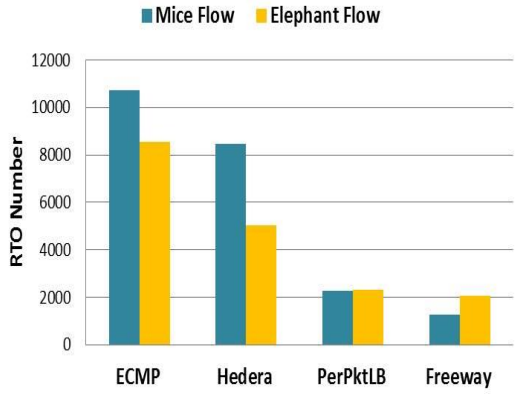
S = {e_1,e_2,e_3 ,e_4 ,e_5 ,e_6 }

同时满足低延时高吞吐量的方法研究(5/5)

- 流调度
  - 路径标记
    - VLAN+TRUNK
    - 无需对现网做任何修改，可直接部署
  - 大流集中式调度
    - 响应式流表加载（Reactively）
    - Bin-Packing
  - 小流分布式调度
    - 流表项预加载(Proactively)
    - ECMP

实验结果





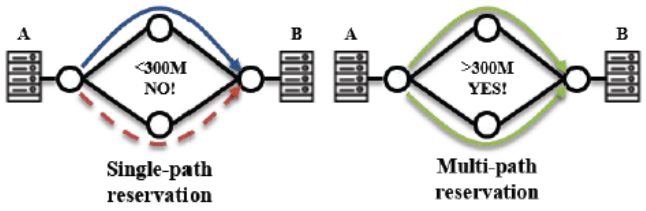
吞吐量：大流的吞吐量相对于传统的ECMP和基于集中式调度的Hedera方法明显提高，略逊于包级别的流量调度的吞吐量，但是包级别的吞吐量会产生TCP的乱序，造成Head of Line Blocking

延时：小流的延时得到了明显优化

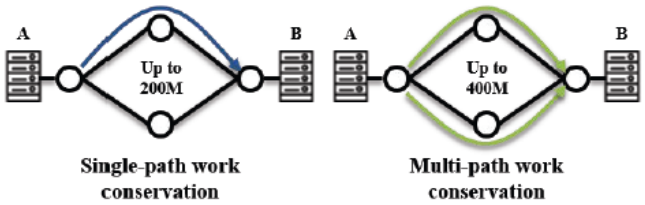
RTO：Freeway中大流和小流的RTO次数也明显减少，节省了大量的网络带宽资源

满足租户不同QoS的多路径带宽保障方法研究

- 研究问题
  - 如何基于DCN的多路径特性为租户提供带宽保障，保障租户的不同QoS
  - 单路径vs.多路径（如右图）
    - 带宽利用率低
    - 不支持灵活负载均衡



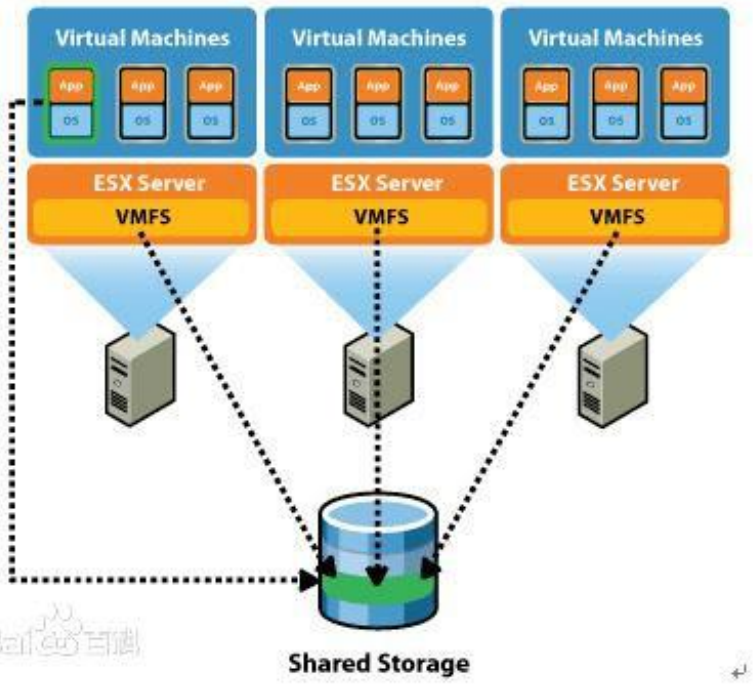
(a) All links' bandwidth usage are 800M/1000M. Request bandwidth from A to B is 300M.

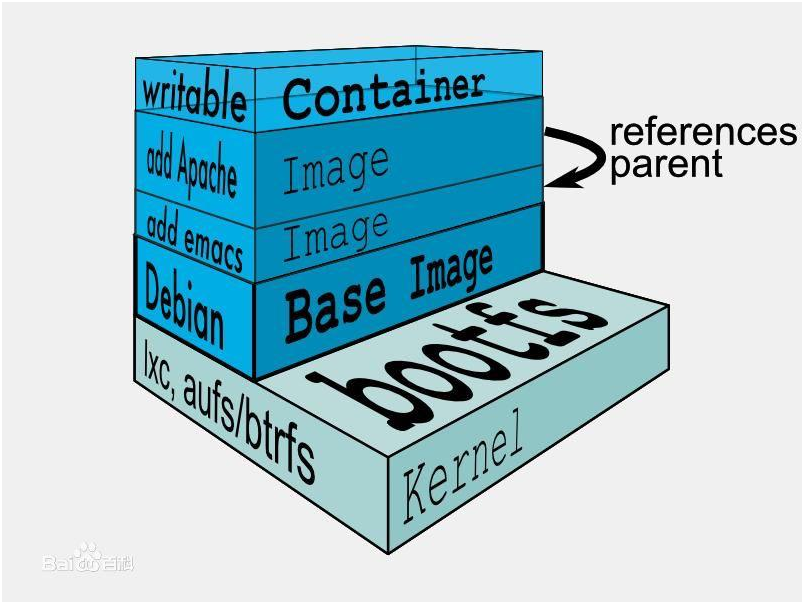


(b) The bandwidth from A to B is successfully reserved. All links' bandwidth usage are currently 800M/1000M

云计算多租户带宽保障

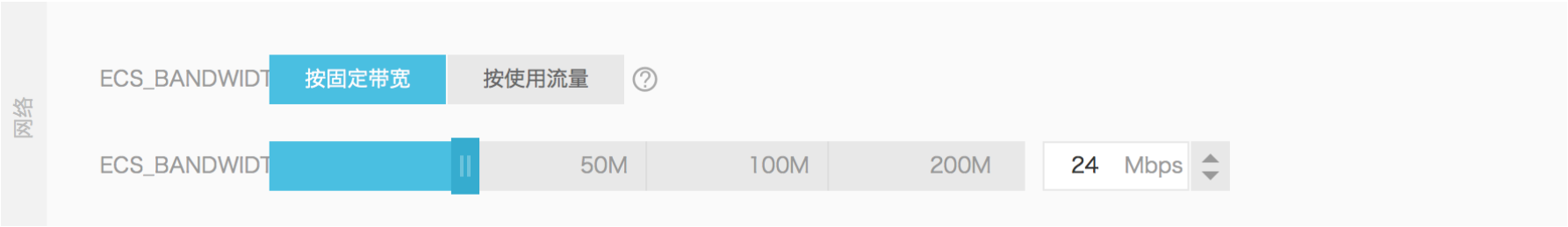
- 公有云平台（如阿里云，亚马逊云等）公开为租户提供弹性虚拟化的计算资源，网络资源和存储资源
- 当前，计算资源和存储资源已经较好地实现资源的虚拟化与隔离化，如KVM，VMWare，XEN，Docker等





云计算多租户带宽保障

- 然而网络资源并没有很好地隔离，而是被各租户共享共享的网络不能提供可估计且可靠的网络服务
- 为租户提供带宽保障是云计算提供QoS的重要方法
- 带宽保障：为租户的各节点之间，提供租户定义的带宽资源

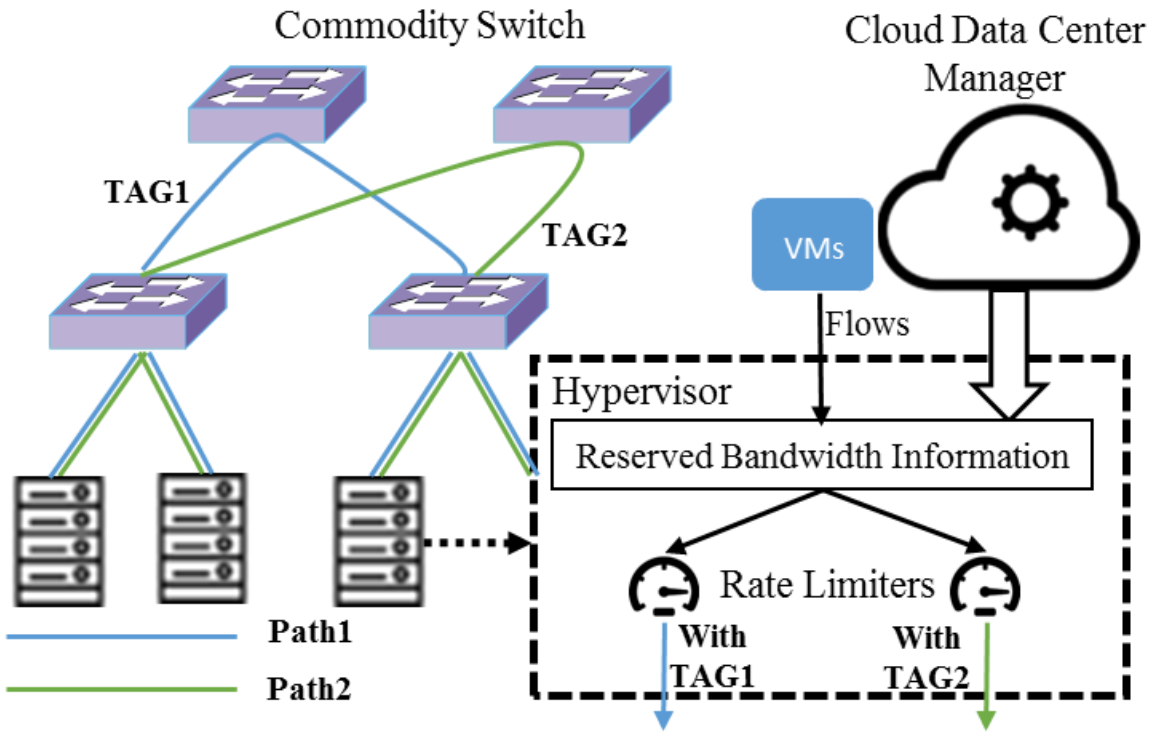


研究现状—面向租户的带宽保障

- 基于单路径的带宽保障
  - 文献：[CONEXT'10, SIGCOMM'11, USENIX WIOV'11, SIGCOMM'13, NSDI'13, HOTCLOUD'13]
  - 特征：从带宽保障模型，自适应性，控制方式到交换机需求和拓扑需求等不同的角度，提出了各种带宽保障方法
- 基于多路径的流调度
  - 文献：[SIGCOMM'11, NSDI'10]
  - 特征：充分利用DCN的多路径的特点，提高网络利用率
- 小结
  - 单路径的带宽保障方法由于没有充分利用DCN的多路径特性，链路利用率低且无法与流量负载均衡配合部署，而多路径的传输机制无法为租户提供严格的带宽保障

满足租户不同QoS的多路径带宽保障方法研究(2/2)

- 研究思路
  - 概括：充分利用DCN的多路径特性，在多条路径上进行带宽分配并且充分利用多路径的动态空闲带宽，并且与流量负载均衡配合使用

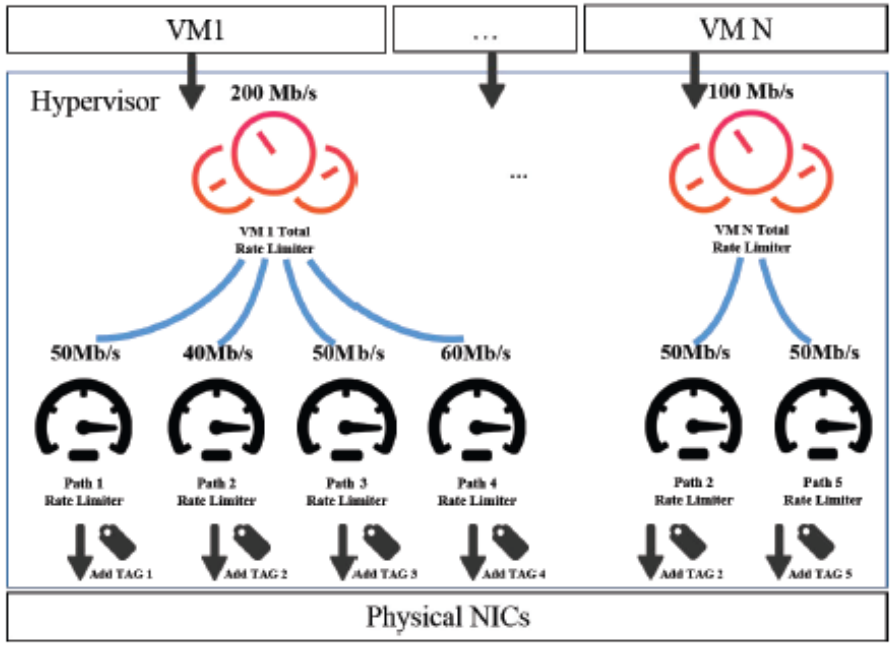


满足租户不同QoS的多路径带宽保障方法研究

- 研究思路

①多路径带宽分配

- 多路径带宽分配算法
- 使用多路径令牌桶在主机端实现流在多条路径上的限速
- 使用源路由，数据包沿既定的路径传输

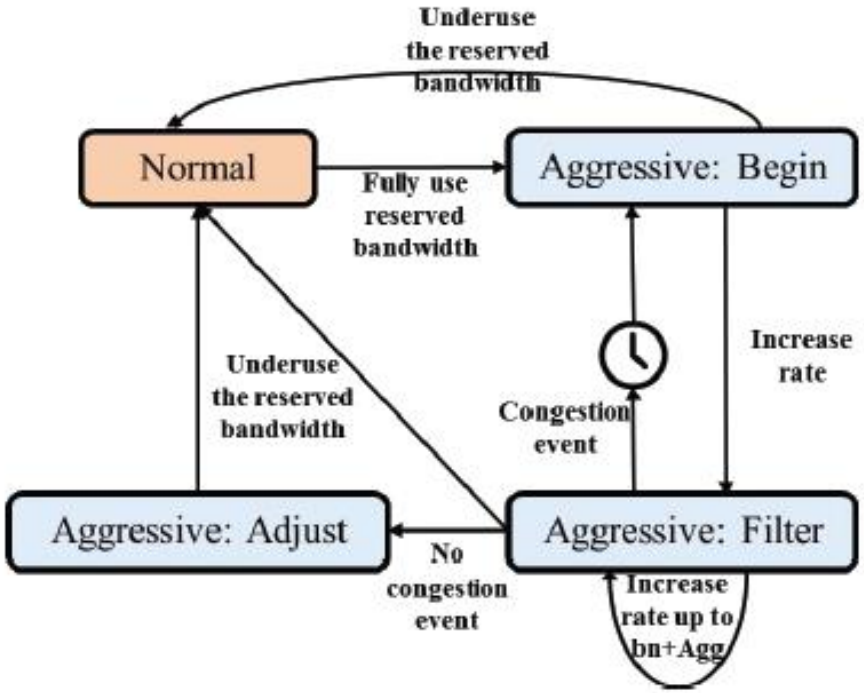


满足租户不同QoS的多路径带宽保障方法研究(2/2)

- 研究思路

②多路径Work Conservation

- 多路径Work Conservation算法，利用状态机分别控制各路径的流速



Work Conservation状态机