对等网络副本最佳占有率缓存管理策略

王 芬 谢长生 陆正武 王宇德 詹 玲
(华中科技大学计算机科学与技术学院外存储国家专业实验室 武汉 430074)

摘 要 随着对等网络应用的不断深入,如何减少时间延迟,减轻集中性带宽负载,提高服务质量,已经成为研究的一个重点。提出了CORPC缓存管理方案。该方案通过使用流媒体片段的流行度来定义媒体片段副本数可占用的最佳系统缓存容量,综合考虑流媒体片段已有的副本容量、流媒体片段的热度、系统节点存储容量,使用启发式贪婪算法来实现缓存准入和缓存替换机制。该方案兼顾了不同热度的媒体片段的服务质量。模拟环境的测试结果表明,随着节点缓存空间的增加,系统服务质量得到改善。

关键词 对等网络,缓存,热度,副本,容量

Replica-optimized P2P Cache Management Strategy

WANG Fen XIE Chang-sheng LU Zheng-wu WANG Yu-de ZHAN Ling
(National Storage System Laboratory, College of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan 430074, China)

Abstract According to the in-depth research on P2P stream media application, how to improve quality of service is a main aspect in research work. We proposed a CORPC caching management scheme, which defines a best caching capacity of chunk's replicas in a P2P system. Based on media chunk popularity, replica number and peer capacity, CORPC scheme adopts heuristic greedy algorithm to implement cache admission control and replacement policy. This scheme can improve quality-of-service (QoS) of chunks with lower popularity while with little sacrifice of chunks with high popularity. Experiment results show that system QoS is improved obviously with an increase of peers' cache capacity.

Keywords Peer-to-Peer network, Cache, Popularity, Replica, Capacity

1 引言

近年来,对等网络在文件共享、流媒体播放、协同计算等方面应用得越来越广泛。典型的对等网络文件共享系统有Gnutella[1],Kazaa[2]等。共享文件通常较小,流媒体应用相对于文件共享,具有数据量大的特点。为解决时间延迟,提高服务质量QoS,通常采用对流媒体文件进行分段,并在代理服务器或对等网络中的各个peer节点分别存储系统中部分流媒体片段的方法。由于代理服务器的数量有限,对于大数据量的流媒体访问仍旧存在缓存容量和网络带宽限制,因此,目前的研究集中于在各个节点中缓存流媒体片段。如何高效地利用各个节点的缓存空间,开发好的缓存准入策略和替代策略,对流媒体片段的缓存进行良好的管理,既保证不同热度的流媒体片段的访问性能,又对缓存空间进行最大限度的利用,是目前研究的一个重点。

2 相关工作

流媒体系统对服务的可靠性、实时性要求较高,系统的缓存管理策略直接影响了系统的服务质量QoS。P2P系统中节点的异构性[3]使得对缓存管理不仅需要考虑流媒体片段的访问流行度,还需要考虑不同节点的特性,如处理速度、带宽、存储空间等。

2.1 P2P流媒体系统的缓存策略

在文献[4]中Hefeeda等人提出了基于流媒体片段的缓存策略。该缓存策略重点考虑的是各个peer节点的服务带宽和缓存空间,而对流媒体片段的访问热度则考虑较少。在文献[5]中Guo等人创建了结构化P2P流媒体系统PROP。系统中的缓存管理策略考虑了流媒体片段的流行度和使用价值。该算法的缺陷有两点:第一是没有考虑流行度位于两端的流媒体片的服务质量;第二是没有考虑到系统总体存储容量变化对使用价值的影响。在文献[6]中Ying等人提出了基于缓存需求的缓存管理策略。该算法未考虑流媒体片段在系统中已存在副本的状况,会导致热度高的流媒体片段占有系统的缓存空间越来越大。这种方法只考虑了热度较高的流媒体片段,不能保证热度较低的流媒体片段的服务质量。在文献[7]中Bin等人在GridCast上分别使用了SVC缓存方法和

到稿日期:2008-11-03 返修日期:2009-01-07 本文受国家973重大基础项目(G1999033006),国家自然科学基金项目(60173043,70771043)资助。

王 芬 女,讲师,主要研究方向为计算机系统结构、存储系统等,E-mail:wangfen@hust.edu.cn;谢长生 教授,博士生导师,主要研究方向为NAS与SAN存储体系、iSCSI存储设备、图像存储与处理和计算机系统结构;陆正武 博士生,主要研究方向为计算机存储系统;王宇德 博士生,主要研究方向为计算机存储系统;詹 玲 博士生,主要研究方向为计算机系统结构。

MVC 缓存方法。研究表明，副本的使用对于系统效率和负载均衡具有重要的作用。该文中的 MVC 方法未根据流媒体片段的热度、节点的缓存空间等进行设计。对缓存空间占满时进行直接清空，未设计缓存替换算法。

2.2 CORPC 流媒体缓存策略

本文中提出的 P2P 流媒体缓存管理策略 CORPC(Cache management based on Optimization of Replica, Popularity & Capacity)，通过定义的最佳副本缓存容量对流媒体片的副本数进行调整，并根据当前副本数、片段热度、系统缓存空间设计缓存准入算法和缓存替换算法，保证了不同热度的流媒体片段的服务质量，并最大限度地利用了缓存空间。

3 系统构造

结构化对等网络具有节点状态变化频繁、系统结构不稳定的特点[8]。本系统采用非结构化对等网络混合式 P2P 架构，系统架构如图 1 所示。各个 peer 节点按照物理网络分成多个自治的簇(cluster)，每个簇中均存在一个性能较高的超级节点 SP。超级节点 SP 上保存该簇内所有的 peer 节点信息和在本簇内缓存的所有流媒体片段的信息。各个簇内部采用集中目录式 P2P 模式，而整个 P2P 网络中各个不同的簇之间再通过纯 P2P 的模式将超级节点连接起来。Tracker 服务器是系统 SP 的索引节点，它保存整个网络中可以利用的超级节点信息。在系统中还存在多个引导节点，以帮助新加入的节点成功地加入对等网络。

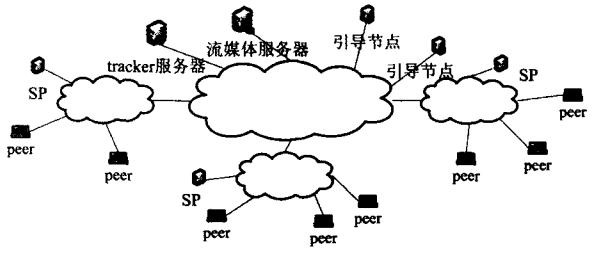

图 1 混合式 P2P 流媒体系统架构图

4 CORPC 缓存管理机制

4.1 关键值的定义

本文中使用的符号定义如下：

ID：流媒体片段 ID，是一个 128bits 的值。由(文件名＋片号)通过哈希函数得到，该值在整个 P2P 流媒体系统中是一个唯一的标识符。

A_i：流媒体片段 i 在一个时间单元内的访问次数。时间单元由系统给出。例如：可以是 24h，24×7h 等。

RP_i：流媒体片段 i 的前一个时间单元内的相对流行度。

R_i：流媒体片段 i 在对等网络中的副本数。

S_i：流媒体片段 i 的大小。

BR_i：流媒体片段 i 的最佳缓存大小。它根据流媒体片段的相对流行度进行赋值。

P_i：流媒体片段 i 的缓存优先级。

假设系统中的流媒体片段为 m 个($m>1$)，peer 节点数为 N，每个节点提供的缓存空间大小均为 C。

· 相对流行度 RP_i

流媒体片段 i 的相对流行度 RP_i(Relative Popularity)为在一个时间单元内片段 i 的访问次数与系统中所有流媒体片段访问次数的比值，它反映的是对流媒体片段 i 访问的相对热度。RP_i 的计算公式如下。

$$RP_i=A_i/\sum_{i=1}^{m}A_i \text{ 且 } \sum_{i=1}^{m}RP_i=1 \quad \text{故 } RP_i<1$$

· 最佳缓存大小 BR_i

当流媒体片段 i 的相对流行度 RP_i 增加时，它的副本在系统缓存空间中占有的比例就应该加大，而当相对流行度 RP_i 减小时，则它占有的系统缓存空间的比例就应该缩小。因此，流媒体片段 i 的最佳缓存大小 BR_i(Best Replica size)由片段 i 的访问相对热度和系统缓存空间共同确定。系统中所有节点提供的总体缓存大小为 CN，即节点数与节点缓存空间 C 的乘积。最佳缓存大小 BR_i 为相对流行度与总体缓存空间的乘积，表示如下：

$$BR_i=RP_i\times CN \text{ 且}$$

$$\sum_{i=1}^{m}BR_i=CN$$

缓存管理时，当各个流媒体片的缓存大小符合各自的最佳缓存大小时，流媒体片的缓存分布就与各自的相对流行度相符合了。这样就可以保证具有不同相对热度的流媒体片在系统中存在合适的副本数。

· 优先级 P_i

它由两方面确定：流媒体片 i 已有副本所占缓存空间大小与最佳缓存大小 BRi 的比值 λ_i；流媒体片的相对流行度 RP_i。下面分别说明。

(1)流媒体片 i 已有副本所占缓存空间大小与最佳缓存大小 BR_i 的比值 λ_i

$$\lambda_i=R_i*S_i/BR_i$$

当流媒体片 i 的缓存副本数为 0 时，λ_i 取极小值 0。当副本无限接近系统所有缓存空间大小 CN 时，λ_i 取极大值 $1/RP_i$。因此，$0\leqslant\lambda_i<1/RP_i$。

λ_i 与优先级 P_i 的关联如表 1 所列。表 1 表明了当流媒体片段 i 的相对流行度 RP_i 固定不变时 λ_i 与优先级 P_i 的关系。当 $\lambda_i=0$ 时，流媒体片未被 peer 节点缓存，则需要大量增加副本数，此时优先级最高。当 $0<\lambda_i<1$ 时，则 $R_i*S_i<BR_i$，流媒体片 i 的副本数还未达到最佳副本数，需要增加它的副本数，因此副本淘汰率较低，流媒体片优先级较高。$\lambda_i=1$ 时，此时流媒体片 i 的副本数为最佳副本数。$\lambda_i>1$ 时，则 $R_i*S_i>BR_i$，流媒体片 i 的副本数超过了最佳副本数，需要减少。$\lambda_i\to1/RP_i$ 时，需要大量减少副本数，被淘汰的概率最高，优先级最小。显然，当相对流行度 RP_i 不变时，随着 λ_i 的增加，优先级 P_i 减少。当 λ 取最小值 0 时，要增加的副本数最大，为 BR_i/S_i。当 $\lambda_i\to1/RP_i$ 时，需减少的副本数为 $(1-RP_i)*CN$。

表 1 λ_i 与优先级 P_i 的关联

λ_i	已有副本数	副本数需求	副本淘汰率	优先级 P_i
$\lambda_i=0$	0	需大量增加	无	最高
$0<\lambda_i<1$	未达到最佳数量	需增加	较小	较高
$\lambda_i=1$	最合适副本数	0	0	中等
$1<\lambda_i<1/RP_i$	过量	需减少	较高	较低
$\lambda_i\to1/RP_i$	超量	需大量减少	高	低

(2)流媒体片的相对流行度 RP_i

相对流行度 RP_i 越高，说明单位时间内访问的频度越大，越需要缓存，因此优先级 P_i 越高。

优先级 P_i 与 λ_i 成反比，与相对流行度 RP_i 成正比。$P_i=$

$f(\alpha RP_i/\beta\lambda_i)$，其中，$\alpha$ 和 β 分别是 RP_i 和 λ_i 的调节因子，反映在优先级的计算中 RP_i 和 λ_i 各自的权重。当对相对流行度的考虑更多一些时，则 $\alpha>\beta$；当对副本容量和缓存空间考虑更多一些时，则取 $\alpha<\beta$。

4.2 缓存初始分配机制

流媒体服务器建立后，按分片策略对流媒体文件进行分片，假设流媒体片段 i 的数据容量为 S_i。在建立初期，所有流媒体文件都无访问时，对流媒体片 i 的最佳缓存大小 BR_i 赋予初始值，使得每个流媒体片段根据数据容量按比例占有系统缓存容量，此时有：

$$BR_i=(S_i/\sum_{i=1}^{m}S_i)\times CN$$

其中，$S_i/\sum_{i=1}^{m}S_i$ 表明片段 i 的数据容量占有比例。

采用 PUSH 方法，主动将流媒体片推到各个节点上去，使得各流媒体片段的副本容量达到各自的最佳缓存容量。

4.3 缓存准入机制

当有新 peer 节点加入对等网络时，它会向系统贡献自己的缓存空间。节点请求对流媒体数据的访问时，被请求访问的流媒体片在本地 peer 节点上可能有两种存在方式。第一种是仅在内存中存在，当播放完毕时释放内存空间。第二种是在硬盘开辟的缓存空间缓存，被缓存的流媒体片可共享给其他节点使用。本研究采用 CORPC 在 peer 节点加入时考虑最大限度地利用节点缓存空间，同时对优先级高、已有副本容量离最佳副本存储容量差距大的流媒体片段优先存储。

4.3.1 问题描述

系统中存在 m 个流媒体片段。对于流媒体片段 i，存在一个二元组$\{S_i,P_i\}$，该二元组中 S_i 为流媒体片的数据容量，P_i 为优先级。新加入的 peer 节点的缓存空间为 C。假设 peer 节点请求访问的流媒体片段为第 n_j 片，它的优先级为 P_j。目前要在 m 个流媒体片段中，选取部分流媒体片段放入到节点硬盘缓存空间，并使得在满足选中的流媒体片段容量大小不超过 peer 节点缓存空间 C 的前提下，$S_i\times P_i$ 取得最大值。

4.3.2 建立模型

对所有 m 个流媒体片段根据优先级使用快速排序法进行排序，形成集合 $N=\{n_1,n_2,n_3\cdots,n_{i-1},n_i,n_{i+1}\cdots n_m\}$。该集合是一个有序数列，按照优先级由高到低依次排序。因此对流媒体片集合 N，有优先级集合 $P=\{p_1,p_2,p_3,\cdots,p_{i-1},p_i,p_{i+1}\cdots p_m\}$，且对任何 $l,m\in[1,m]$，若 $l<m$，则 $p_l\geqslant p_m$。对于流媒体片 n_i，存在一个二元组$\{S_i,P_i\}$。peer 节点的缓存空间为 C。考虑所有优先级大于等于 P_j 的流媒体片，它们形成集合 $J=\{n_1,n_2,n_3\cdots,n_{j-1},n_j\}$，显然，集合 J 是集合 N 的一个子集。令 $Z=\sum_{i=1}^{j}s_j$，则存在 $Z>C$ 和 $Z\leqslant C$ 两种情况。下面分别就这两种情况进行讨论。

1) 当 $Z>C$ 时，需要在集合 J 中选择部分流媒体片段放入到该缓存空间中，在满足选中的流媒体片容量和不超过节点缓存空间 C 的前提下，使得 $\sum S_i\times P_i$ 取得最大值。该问题可转换成 0-1 背包问题。问题为从集合 J 中求解最佳装载$\{n_i\}$，在 $\sum S_i<C$ 前提下，使 $\sum S_i\times P_i$ 取得最大值。

2) 当 $Z\leqslant C$ 时，要从 m 个流媒体片段中寻求最佳装载。这仍旧是一个背包问题。问题为从集合 N 中求解最佳装载$\{n_i\}$，在容量和不超过 C 的前提下，使 $\sum S_i\times P_i$ 取最大值。

4.3.3 求解算法

使用启发式贪婪算法 CacheLoading 来对该 0-1 背包问题进行求解。算法设计思路是：每次从剩余的流媒体片段中选择可以装入的优先级 P_i 最高的流媒体片段进行装载，直至所有流媒体片段均装入缓存区或是缓存区不再能够容纳其他任何一个流媒体片段。算法 CacheLoading 调用快速排序法 QuickSort(w,t,n)，对 n 个流媒体片段按照优先级 w 由高到低进行排序，形成排序寻址表 t。排序后，对 $\forall i\in[1,n-1]$，$\exists w[t[i]]\geqslant w[t[i+1]]$。使用原地分割法实现 QuickSort，其空间复杂度为 O(log$n$)，时间复杂度为 O($n$log$n$)。

流媒体片段装入缓存区问题的启发式贪婪算法 CacheLoading 中的符号表述及伪代码描述如下：

```
// i 表示第 i 个流媒体片段 1≤i≤n
// x[i] = 1 表示流媒体片段 i 被装入缓存区
// x[i] = 0 表示流媒体片段 i 未被装入缓存区
// c 是缓存区剩余容量，CS 是缓存区大小
// w 是流媒体片的优先级
// t 是排序寻址表
Procedure CacheLoading(x[ ],w[ ],c,n) {
[1] 初始化排序寻址表 t，为 t 分配内存，大小为 n+1；
[2] 调用 QuickSort(w,t,n)；//对流媒体片段排序
[3] for (i←1 to n)
[4]     初始化 x[i] = 0；
[5] 对 c 赋初值 CS；
[6] for (i←1 to n)
[7]  if (c≥0) and (当前的 w[t[i]]≤缓存区剩余容量 c)
[8]     then {
[9]             对 x[t[i]] 赋值 1；//装载入缓存区
[10]            将 c← w[t[i]]赋给 c；//计算剩余容量
[11]            } // end if
[12]    对 i 赋值 i+1；// end for
[13] 释放 t 占用的内存；
    }
```

该算法总的时间复杂性度 O($n$log$n$)。

4.3.4 解集描述

根据上述 CacheLoading 实现算法可知，当 $Z>C$ 时，实际上是从所有优先级大于等于 P_j 的流媒体片段中获取部分片段对缓冲区进行装载。当 $Z\leqslant C$ 时，所有优先级大于等于 P_j 的流媒体片段均被装入缓冲区中。

peer 节点使用 PULL 方式获取请求访问流媒体片 n_j。对于解集合$\{n_i\}$中的其他流媒体片，则使用 PUSH 的方式，将流媒体片段推入 peer 节点的缓存空间。

4.4 缓存替换机制

当一个 peer 节点请求访问的流媒体片段 j 时，若缓冲区已满，则要考虑是否需要删除缓冲区中的一个或多个流媒体片段，并将流媒体片段 j 替换进来。假设在当前 peer 节点的缓存区中流媒体片段集合为$\{n_1,n_2,n_3\cdots,n_k\}$，定义集合 $K=\{n_1,n_2,n_3\cdots,n_k,n_j\}$。问题为如何从集合 K 中选择要装载的流媒体片段 n_i，在满足 $\sum n_i\leqslant C$ 的前提下，$\sum S_i\times P_i$ 取最大值。该问题一样可转换为 0-1 背包问题。其中，C 为背包容积，集合 K 为物品集，P_i 为物品价值。

这里采用与缓存准入机制相同的算法思路，唯一的区别在于此处的物品集是本地节点的流媒体片段集 K。因此，缓

（下转第 293 页）

minimum statistics and soft decision for robust noise power estimation in speech enhancement[J]. IEEE Signal Processing Letters,2004,11:95-98

[11] Amrt A,Mitiche A,Dubois E. Reliable and fast structure-oriented video noise estimation[J]. IEEE Proc. Int. Conf. on Image Processing,2002,1:840-843

[12] Starck J L , Murtagh F . Automatic noise estimation from the multiresolution support[J]. Publications of the Astronomical Society of the Pacific,1998,110:193-199

[13] Meer P,Jolion J M,Rosenfeld A. A fast parallel algorithm for blind esitmation of noise variance[J]. IEEE Trans. Pattern Anal. Machine Intell. ,1990,12:216-223

[14] Rank K,Lendl M,Unbehauen R. Estimation of image noise variance[J]. IEE Proc. Vis. Image Signal Process,1999,146:80-84

[15] Olsen S L. Estimation of noise images : An evaluation[J]. CVGIP. Graph models Image Process. ,1993,55:319-323

[16] Martin D R,Fowlkes C C,Malik J. Learning to detect natural image boundaries using local brightness,color,and texture cues [J]. IEEE Trans. Pattern Anal. Machine Intell. ,2004,26:530-549

(上接第 257 页)

存替换机制的算法时间复杂度为 O($k\log k$)。每个 peer 节点的缓存空间容量是有限的,且流媒体片段通常会以 M 为基本单位。因此 k 是一个不大的数字,相对于目前计算机的处理速度而言,时间复杂度 O($k\log k$)所需要的时间是较少的。

5 性能分析

系统拓扑采用构造工具 GT-ITM[8] 生成 transit-stub 层次模型。系统生成 3 个 transit 域,transit 域的平均骨干节点数为 4,骨干节点平均连接 2 个 stub 域。每个 stub 域的平均 host 节点数为 3 个。系统中每个 host 节点均为一个 peer 节点,系统节点数为 72 个。在 transit 域中选择一个骨干节点作为流媒体服务器,放置 100 个媒体文件,媒体文件的数据容量为 150M～1G,编码率为 300～500kbps。在 P2P 流媒体系统中,对流媒体片段的访问服从 Zipf 分布,模拟环境中,假设对流媒体片段的访问服从 θ=0.20 的 Zipf 分布。节点对媒体文件的访问请求以一个恒定的速率到达。假设节点以恒定的速率加入和离开系统,测试节点缓存大小为 10M～200M 变化下的系统性能。在系统性能稳定时进行取值,节点缓存大小变化时的流媒体片段副本数、服务器连接数(反映服务器负载状况)、启动延时的变化如图 2、图 3、图 4 所示。

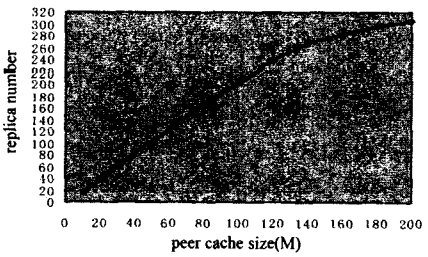

图 2 缓存大小不同时的副本数

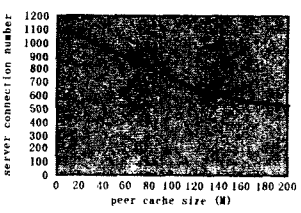

图 3 不同缓存大小时的服务器连接数

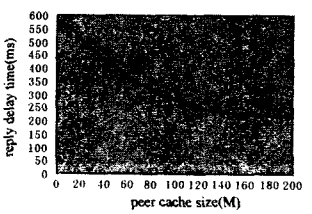

图 4 不同缓存大小时的请求响应延迟

图 2 显示的是 peer 节点数恒定情况下,peer 节点缓存大小从 10M～200M 变化时的系统流媒体片段副本数状况。从图 2 可以看出,随着 peer 节点缓存空间的加大,流媒体片段的副本数不断增加。

图 3 显示的是 peer 节点缓存大小变化时服务器连接数的情况。该图表明节点缓存空间加大,服务器的连接负载情况就会明显减少。在缓存从 10M 变化至 120M 时,服务器端的连接数显著降低,之后降低速度逐渐放缓。

图 4 显示的是 peer 节点缓存大小变化时服务请求响应延迟的状况。该图说明节点缓存空间加大,流媒体视频播放响应延迟则缩短。在缓存从 10M 变化至 100M 时,响应延迟缩短速度快,之后缩短逐步缓慢。

综合解读图 2、图 3、图 4 可看到,随着系统整体缓存空间的加大,流媒体片段副本数随之增加,系统的整体响应时间延迟缩短,骨干网络通信负载降低。表明系统缓存空间大小对流媒体服务质量有较大的影响,因此,有效利用缓存空间可以有效地改善流媒体的服务质量。

结束语 本文提出了一种对等网络混合式拓扑结构下的 CORPC 缓存管理方案。该方案在考虑流媒体片段流行度的同时,最大限度地利用了节点缓存空间。通过缓存空间分配算法,兼顾了流行度较高和较低的流媒体片段的服务质量。将来的工作包括进一步完善该策略,在缓存预取等方面做进一步研究。

参 考 文 献

[1] http://www.gnutella.com,Gnutella website

[2] http://www.kazaa.com,KaZaA website

[3] Saroiu S,Gummadi P,Gribble S. A measurement study of peer-to-peer file sharing systems[C]//Proceedings of MMCN02. San Jose,CA,USA,January 2002

[4] Hefeeda M M , Bhargava B K , Yau D K Y. A hybrid architecture for cost-effective on-demand media streaming[J]. Computer Networks,2004,44(3)

[5] Guo,Chen L S,Zhang X. Design and evaluation of a scalable and reliable P2P assisted proxy for on demand streaming media delivery[J]. IEEE Transactions on Knolwedege and Data Engineering,2006,18(5):669-682

[6] Ying L , Basu A. pcVOD: Internet Peer-to-Peer Video-On-Demand with Storage Caching on Peers[C]//Proceedings of DMS. 2005

[7] Cheng Bin,Stein L,Jin H,et al. Towards cinematic internet video-on-demand[C]//EuroSys. 2008:109-122

[8] Calvert K,Zegural E. GT2ITM:Georgia Tech Internetwork Topology Models[OL]. http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html,2006-10