

文章编号: 2096-1472(2016)-07-06-04

一种基于节点能力区分的Pareto最优网络编码组播方法

郑力明¹, 符永铨², 李晓冬³

(1.武警警官学院电子技术系, 四川 成都 610213;

2.国防科技大学计算机学院并行与分布处理国防科技重点实验室, 湖南 长沙 410073;

3.武警警官学院科研部, 四川 成都 610213)

摘要:快速数据分发在突发事件响应, 军事领域等具有重要的应用。针对异构用户节点群体下快速数据分发问题, 提出基于能力区分的拓扑构建和速率控制的网络编码组播协议CORE。CORE利用能力区分的自适应层次化拓扑构建鼓励节点提供高的上传带宽并优化系统范围吞吐量; 利用直方图的方式对基于网络编码的数据传输进行流量控制, 降低冗余数据的传输; 基于分布式的速率控制实现Pareto最优的下载速率分配。实验结果表明CORE具有良好的可扩展性, 能够充分利用异构节点的上传能力, 提供区分的下载带宽分配, 较高的数据传输吞吐量、低端到端网络延迟, 能够提供异构网络环境下分发时间紧迫的数据分发服务。

关键词: 数据分发; 网络编码; 能力区分; 速率分配

中图分类号: TP311

文献标识码: A

An Optimal Pareto Network Coding Multicast Method Based on Node Capacity Differentiation

ZHENG Liming¹, FU Yongquan², LI Xiaodong³

(1. Department of Electronics Technology, Armed Police Officer Academy, Chengdu 610213, China;

2. National Key Laboratory for Parallel and Distributed Processing, School of Computer,

National University of Defense Technology, Changsha 410073, China;

3. Ministry of Scientific Research, Armed Police Officer Academy, Chengdu 610213, China)

Abstract: Fast data distribution has been significantly applied in emergency response, military and other fields. Aiming at the problem of fast data distribution under the different user node groups, the paper proposes a network coding multicast protocol, CORE, based on the capability-differentiation topology construction and rate control. The adaptive hierarchical topology is constructed to provide high upload bandwidth and optimize the system throughput rate. The histogram is applied to control the traffic of data transmission based on network coding, which can reduce redundant data transmission. The optimal allocation of Pareto is achieved through distributed rate control. The experimental results show that, with good scalability, CORE can make full use of the ability to upload heterogeneous nodes, and provide a differentiated download bandwidth allocation, higher data transmission throughput and lower end-to-end network delay. CORE can also provide urgent data distribution service in heterogeneous network environments.

Keywords: data distribution; network coding; capacity differentiation; rate allocation

1 引言(Introduction)

覆盖网通过将数据传输负载分布到用户节点极大提高了数据分发^[1]过程的扩展性, 然而基于覆盖网的数据分发存在“搭便车”(free loader)的问题^[2]: 一些节点消耗覆盖网的数据传输能力却不能提供足够的上传能力。已有的基于激励机制的覆盖网数据分发协议关注在传统存储转发环境下用户间的协作^[3-5], 然而在利用网络编码技术的快速数据分发环境下面临如何在适合网络编码组播的拓扑结构中实现能力区分; 如何调度网络编码传输速率实现优化的下载带宽分配等问题。同时网络编码环境下已有的基于速率控制的优化方式假设存在独立的服务节点^[6-8], 需要额外的部署开销。

针对异构节点群体实现快速数据分发的问题, 提出了基于能力区分的方式进行拓扑构建和速率控制的网络编码组播协议CORE(Capacity-differentiation Optimal RESilient multicast based on network coding)。模拟测试显示CORE协

议具有高度的扩展性, 较高的吞吐率和低端到端延迟等优点。

2 相关工作(Related work)

由于覆盖网环境下用户节点提供上传能力需要消耗本节点的网络传输能力, 研究发现覆盖网中存在大量的不提供上传带宽而仅获取数据的用户节点, 导致请求信息被转发到系统中提供数据下载服务的少量节点(称为“tragedy of the commons”)^[2]。针对覆盖网环境下用户节点服务区分和激励问题, 已有的研究包括: 文件共享环境下单个节点上传带宽分配机制^[3], 多个覆盖网数据传输会话优先调度和带宽分配^[4], 集中式的基于微分方程建模提供静态组播环境下不同服务质量^[5]等。另一方面, 为了利用不同能力的节点, Florida大学的Zhang等在结构化覆盖网环上利用异构节点上传能力进行任意源组播(any source multicast)^[9], 没有考虑服务区分等问题。然而在基于网络编码快速数据分发环境下, 面临新的问题包括: 如何构建节点能力感知的高效拓扑结构以提高数据

分发的吞吐率，如何在适合网络编码数据分发的拓扑结构中实现能力区分；如何调度数据传输速率实现优化的下载带宽分配。

在网络编码组播环境下资源优化的研究基于网络流模型，典型工作如Toronto大学的Li等^[6]假定链路的速率和容量为固定，利用Langrangian对偶方式最大化系统范围吞吐率；MIT的Wu等^[8]为每个组播链路流设定开销函数，通过寻找最优的编码子图(每个会话在每个链路的传输数据量)来最小化网络编码组播环境下的传输开销；加州理工的Chen^[7]等利用速率控制优化系统范围的特定性能目标，通过源节点调整传输速率，以及转发节点在多个组播会话间调度的调度实现最大化系统效用。

3 系统模型(System model)

将用户节点组成的网络建模为有向图 $G(V, E)$ ， V 为节点集合， E 为边集。节点集 V 包含两类节点：源节点 S 和接收节点 T (接收节点作为中间转发节点)。 $V = S \cup T, |S|=1$ 。边集 $E = \{e_1, e_2, \dots, e_{|E|}\}$ ，每个边 e_i 具有最大带宽容量 BW_i 。假定每个节点 i 具有一个网络坐标 x_i ，同时缓存源节点的网络坐标 x_s (假定网络坐标为真实的且相对稳定，虚假坐标处理可通过发送额外的探测数据包实现)。每个节点 i 具有一个标识符 $i.id$ ，以及一个层次level，标识符在节点参与系统的生命期内唯一，然而节点 i 的level值可能因网络的演化而变动。节点 i 在加入时声明自己的上传带宽 b_{ui} 和下载带宽 b_{di} 。

为了显示能力区分的有效性，给出下列示例(如图1(a)一图(d)所示，假定所有节点在线)，其中每个节点参数配置利用三元组 $(U_i, Up_i, Down_i)$ 表示，分别代表接收节点 i 的优先度，上传带宽和下载带宽(单位为Mbps)，接收节点参数配置为：

$x_1(1.5, 0.6, 0.7), x_2(1.7, 0.7, 0.9), x_3(0.8, 0.3, 0.7), x_4(1.0, 0.4, 0.6), x_5(1.3, 0.6, 0.8), x_6(1.2, 0.4, 0.6)$

设定源节点 S 最大子女数为2，上传带宽为1Mbps。图1中有向边代表数据传输方向，边上的值为该逻辑连接的传输带宽。考察下载带宽公平分配和按照优先度分配，前者下载节点平均获取父节点带宽，后者设定节点优先度越大，获取下载带宽越高。图1(a)为无能力区分拓扑构建，下载带宽公平分配；图1(b)为无能力区分拓扑构建，带宽根据节点优先度分配；图1(c)为能力区分拓扑构建，下载带宽公平分配；图1(d)为无能力区分拓扑构建，带宽根据优先度分配。

结果显示图1(a)平均吞吐率为0.34Mbps，图1(b)平均吞吐量为0.34Mbps，图1(c)平均吞吐率为0.41Mbps，图1(d)平均吞吐率为0.43Mbps。图1(c)和图1(d)吞吐率的提高显示能力区分的拓扑构建可以有效地利用当前节点的上传带宽，从而提高系统范围节点的吞吐率；同时图1(d)通过基于能力区分的速率控制提升了高优先度节点的下载带宽，从而提高其上传能力，因而在图1(c)的基础上进一步增加了系统范围的吞吐率。

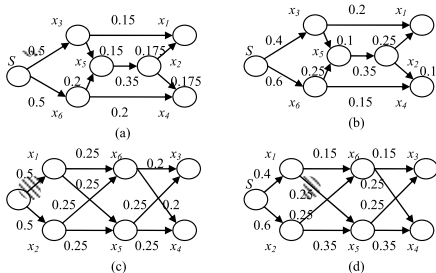


图1 不同拓扑吞吐率

Fig.1 Different topologies throughput

CORE的目标是鼓励节点提高上传带宽，提高数据传输的吞吐率，降低数据完成时间；缓解单源节点数据分发的瓶颈，提高下游节点的传输带宽；充分利用网络编码消除请求特定数据块的优势，降低因完成数据传输的接收节点突然退出导致的性能下降问题。

CORE以能力区分的方式将节点分为多个不相交的层次，每个节点根据优先度大小确定加入的层次，高贡献的节点位于较高的层次以降低距源节点的端到端延迟，源节点分发数据时按照层次由高到低的方向传输数据。

4 能力感知的近似Butterfly网络构造(The ability to perceive the approximate butterfly network structure)

为了提高基于网络编码方式下系统范围节点吞吐率，降低数据传输的端到端延迟，同时基于节点贡献能力实现区分的服务，构建一个类似Butterfly网络的有向无环的拓扑结构(Approximate Butterfly Network, ABN)。

任意层次 L 的一个节点 i 具有三类连接：

- (1)数据连接(General Link)：与层 $L-1$ ($L+1$)的父节点和层 $L+1$ (L 不为最底层)的子女节点建立的传输编码数据的连接。
- (2)同层连接(Level Link)：同一层次节点间建立的逻辑连接以维护不同层间容量范围的严格不相交。
- (3)Butterfly连接(Butterfly Link)：与层 $L-1$ 和层 $L+1$ 的各一个随机节点建立的逻辑连接，分别称为上、下Butterfly连接，负责引导新节点加入和无偏抽样对应层的节点。

图2给出了一个四层的ABN拓扑，虚线代表同层连接，实线为数据连接(部分数据连接作为Butterfly连接)。

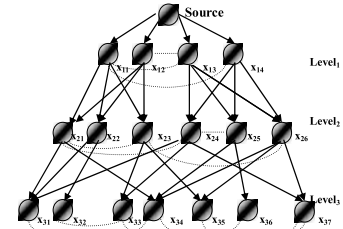


图2 近似Butterfly拓扑网络

Fig.2 The approximate butterfly network structure

4.1 自适应分层

在层次数为1时，拓扑结构转化为所有节点随机连接的mesh网络，而在层次数为 N 时，拓扑结构转化为线性连接。理想的层次数需要保证任意节点距源节点所有路径的最大延迟最低，即：

$$\min \sum_{i \in N} \sum_{P \in Path_i} \sum_{(j,k) \in P} d(j,k)$$

启发式算法包括分割当前最底层和合并当前层两部分：

- (1)分割。若当前最低层到达最大容量后，在新的节点到来时，建立一个新的节点层。
- (2)合并。若第 $L+1$ 层的实际节点数目 $< C_{L+1}/2$ ，且在第 $L+1$ 层所有节点加入第 L 层后，第 L 层节点数目 $< C_L$ ，则将层 $L+1$ 与层 L 合并。

4.2 基于滑动窗口路由机制判定层次容量范围

为了轻量级的估计每层的容量范围，实现基于能力区分的自适应分层，提出基于滑动窗口方式动态的维护每层容量范围。任意节点 i 在时刻 t 维护一个当前层容量范围的滑动窗 $w_i^t = [\min_i^t, \max_i^t]$ ， w_i^t 缓存已知的当前层最小和最大的优先度及对应的节点地址 $Addr_{\min_i^t}, Addr_{\max_i^t}$ 。为了动态的维护当前层容量范

围, 节点*i*利用gossip方式^[10]与同层的其他节点*j*交换 $\min'_i, \max'_i$ 值, 然后更新滑动窗口:

$$\min'_i = \min_{j \in E} \min\{\min'_i, \min'_j\};$$

$$\max'_{i+1} = \max'_{i+1} = \max\{\max'_i, \max'_j\}$$

节点*i*在接收到查找当前层容量范围时, 将查找请求转发到 $\min'_i, \max'_i$ 对应的地址 $Addr_{\min'_i}, Addr_{\max'_i}$, 然后上述转发过程递归执行, 最终返回当前节点已知的最小和最大优先度的节点。

5 基于直方图的连续网络编码传输机制(Based on the histogram of a continuous network coding transmission mechanism)

直方图构建: 每个父节点*i*维护一个缓存BlockCache_{*i*}, 保存已接收的编码数据块集合 $B_i(t)$ 及对应的编码系数向量矩阵 $M_i(t)$, 记为二元组 $\langle B_i(t), M_i(t) \rangle$ 。由于节点*i*在时刻*t*发送到任意子女节点的最大线性独立编码块数目为 $|M_i(t)|$, 为了降低传输冗余线性相关数据至子女节点的可能性, 同时进行流量控制, 父节点*i*为每个子女节点*j*维护一个允许传输编码块数目 k'_{ij} :

$$k'_{ij} = |M_i(t)| - r'_{ji}$$

其中, r'_{ji} 为节点*i*在时刻*t*已发送到节点*j*的编码块数目, 随着父节点传输编码块数目而递增。这样父节点维护一个允许传输到每个子女节点数据块数目的直方图, 如图3所示。父节点在接收到新的编码数据时, 更新每个子女节点对应 k'_{ij} , 然后以push方式连续传输至多 k'_{ij} 个编码数据块至对应子女节点*j*。

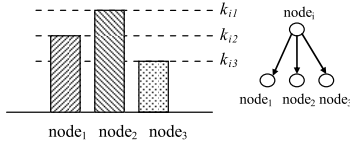


图3 直方图构建

Fig.3 Histogram to build

6 Pareto最优的速率分配机制(Pareto optimal rate allocation mechanism)

6.1 问题建模

假设节点*j*上传和下载能力分别为 P_j, Q_j , 设节点*j*从父节点*i*处获取数据得到效用值为 U^{ij} , U^{ij} 为 x^{ij} 的非减函数, x^{ij} 为父节点*i*沿 (i, j) 链路的传输带宽, 假定 $U^{ij}(x^{ij})$ 为严格的凹函数且二阶可微^[11]。设节点集合*V*中节点按序编号为1, ..., *N*, 边集为*E*。

为保证高优先度的节点获得高的分配带宽, 设定效用函数 U^{ij} 随任意节点*j*优先度增加而增加; 为反映不同节点获得带宽的满足程度, 设定效用函数 U^{ij} 随节点*j*下载能力增大而降低; 为了降低端到端延迟, (i, j) 链路延迟越低, U^{ij} 越大, 设定 H_j, I^{ij} 分别为节点*j*优先度和 (i, j) 链路延迟, 得到一个效用函数表达式:

$$U^{ij} = H_j \log(1 + x^{ij} / (Q_j I^{ij})) \quad (1)$$

除源节点以外所有节点均为接收节点, 接收节点构建自适应的层次传输网络编码数据。构造如下最大化所有节点效用的线性规划问题:

P:

$$\max \sum_{(i,j) \in E} U^{ij}(x^{ij}) \quad (2)$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{j:(i,j) \in E} x^{ij} \leq P_i, \forall i \in N \\ & \sum_{i:(i,j) \in E} x^{ij} \leq Q_j, \forall (i,j) \in E \\ & x^{ij} \geq 0, \forall (i,j) \in E \end{aligned} \quad (3)$$

任何凸规划*P*的最优解 $\{x^{ij}, \forall (i,j) \in E\}$ 给出了一个组播会话

中最优的带宽分配策略。通过最大化上述效用函数, 凸优化*P*保证高优先度的节点获得较高的带宽, 同时, 通过对数函数(严格的凹函数)保证了不同节点带宽分配的公平性^[12]。

6.2 梯度下降方法

首先将原问题*P*的(2)式变为等价的最小化问题: $\min - \sum_{(i,j) \in E} U^{ij}(x^{ij})$, 然后将Lagrangian因子 μ^i 与原问题*P*约束(3)关联, 得到目标函数式(4):

$$\begin{aligned} & - \sum_{(i,j) \in E} U^{ij}(x^{ij}) + \sum_{j:(i,j) \in E} \mu^i \left(\sum_{i:(i,j) \in E} x^{ij} - Q_j \right) \\ & = - \sum_{i:(i,j) \in E} \sum_{j:(i,j) \in E} (U^{ij}(x^{ij}) - \mu^i x^{ij}) - \sum_{i:(i,j) \in E} \mu^i Q_j \end{aligned} \quad (4)$$

现在得到原问题*P*的Lagrangian对偶问题:

$$\text{其中} \quad \max_{\mu \geq 0} L(\mu) \quad (5)$$

$$\begin{aligned} L(\mu) & = \min_C \left(- \sum_{i:(i,j) \in E} \sum_{j:(i,j) \in E} (U^{ij}(x^{ij}) - \mu^i x^{ij}) - \sum_{j:(i,j) \in E} \mu^i Q_j \right) \\ & = \max_C \left(\sum_{i:(i,j) \in E} \sum_{j:(i,j) \in E} (U^{ij}(x^{ij}) - \mu^i x^{ij}) \right) \\ & \quad - \sum_{j:(i,j) \in E} \mu^i Q_j \end{aligned} \quad (6)$$

*C*约束条件如下:

$$\begin{aligned} & \sum_{j:(i,j) \in E} x^{ij} \leq P_i, \forall i \in N \\ & x^{ij} \geq 0, \forall (i,j) \in E \end{aligned}$$

上述公式(6)中的Lagrangian子问题可以被分解为多个凸规划的子问题, 每个子问题可以被节点*i* ($\forall i \in N$)独立的解决, 即:

$$\max \sum_{j:(i,j) \in E} (U^{ij}(x^{ij}) - \mu^i x^{ij}) \quad (7)$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{j:(i,j) \in E} x^{ij} \leq P_i \\ & x^{ij} \geq 0, \forall (i,j) \in E \end{aligned} \quad (8)$$

基于上述子问题的分解, 可以分布式的解决原问题*P*的Lagrangian对偶问题。

梯度下降过程: 首先设置初始化的非负Lagrangian因子 $\mu^i[0], \forall i \in N$, 然后在梯度下降过程的第*k* ($k=1, 2, \dots$)次迭代时, 给定当前的Lagrangian因子 $\mu^i[k]$, 基于water filling方法解决上述公式(7)的*N*个子问题以获得当前的 $x^{ij}[k], \forall j:(i,j) \in E$, 然后更新Lagrangian因子为

$$\mu^i[k+1] = \left[\mu^i[k] + \theta[k] \left(\sum_{j:(i,j) \in E} x^{ij}[k] - Q_j \right) \right]^+, \forall j:(i,j) \in E \quad (9)$$

其中, “+”代表将 $\mu^i[k+1]$ 映射到非负的实数, θ 为单步增加长度的序列。

7 实验测试与分析(The experiment and analysis)

设定每个节点的父节点和子女节点数目分别均匀随机的从6—12选择, 每个节点利用Butterfly连接无偏抽样新的节点, 每个节点定期(设定10s)无偏抽样同层一个节点交换滑动窗口, 每个节点下载和上传能力分别均匀随机的在0.5—4.5Mbps和0.5—1.5Mbps确定, 覆盖网链路延迟为1—100ms均匀随机数。设定每层最大容量比例MAX为0.5以保证每个节点具有至少两个父节点, 下列结果均为10次平均值。

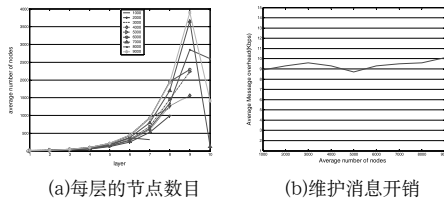
7.1 拓扑结构构造和维护

在不同平均在线节点数目(1000—9000)下, 测试CORE加入过程的分层能力, 考察不同层次的节点分布以及拓扑结构维护开销等性能指标, 结果如图4(a)和图4(b)所示。图4(a)显示各个层次的平均节点数目随层次数目的提高近似以指数形式增加, 有效地降低了系统中层次数目; 图4(b)显示在不

同平均在线节点数目下拓扑结构维护消息带宽开销较小,平均<10kbps,而且随着节点数目增大维护消息开销保持相对稳定。从图4(a)和图4(b)可知CORE能够有效分层,且拓扑结构维护开销较低。

7.2 滑动窗口路由性能

在不同平均在线节点数目下,测试滑动窗口路由机制的性能。统计600秒时间间隔系统范围滑动窗口的性能。考察查找路由路径长度和准确获得当前层容量范围的查找比例(称为命中率)等两个指标。将所有层次性能指标平均,结果如图5(a)和图5(b)所示。图5(a)显示滑动窗口路由机制能够迅速收敛到全局的容量范围,这是由于滑动窗口基于gossip方式扩散当前层次容量范围;图5(b)显示当前滑动窗口确定的对应节点以极高概率(>99.9%)为当前层容量范围上、下限对应节点,说明滑动窗口路由机制能够在网络波动下精确的获取本层的容量范围。从图5(a)和图5(b)可知滑动窗口路由机制能够高效的确定当前层范围。

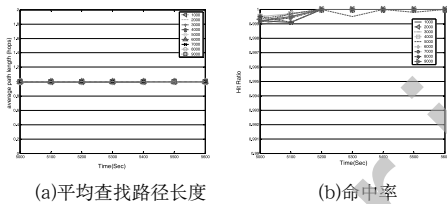


(a)每层的节点数目

(b)维护消息开销

图4 加入过程的分层能力及维护消息开销

Fig.4 Join the process ability of layered and maintenance message overhead



(a)平均查找路径长度

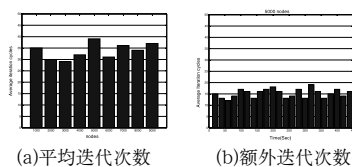
(b)命中率

图5 平均查找路径长度及命中比例变化

Fig.5 The average search path length and hit ratio change

7.3 速率控制收敛性

检验动态网络环境下速率控制方法的收敛速度。设定收敛阈值 $\Delta=0.001$ 。结果如图6(a)和图6(b)所示。图6(a)显示速率控制方法在不同平均节点数目下速率控制方法能够以较低迭代次数收敛,平均的迭代次数<40;且平均迭代次数随节点数目增加变化较小,这是由于速率控制方法为基于局部优化达到全局的收敛,每个父节点均仅与子女节点交换更新消息。图6(b)显示系统所有父节点平均的额外迭代开销较小,平均迭代轮数<20,而且在不同时间额外迭代次数变化幅度较低,这是由于父节点执行额外的速率控制迭代过程时均基于当前子女节点的价格幅度 μ_i ,降低了价格幅度 μ_i 因初始化过程带来的额外迭代过程。



(a)平均迭代次数

(b)额外迭代次数

图6 速率控制方法收敛速度及动态调整开销

Fig.6 Convergence speed and dynamic adjustment overhead rate control method

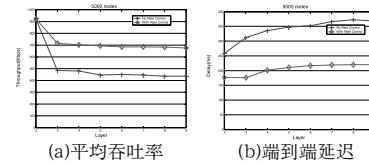


图7 不同层次的吞吐率及端到端延迟

Fig.7 Different levels of throughput and end-to-end delay

7.4 系统吞吐率及端到端延迟

统计平均在线节点数目5000时,不同层次的吞吐率以及端到端延迟,此时节点分为九层,从第二层开始统计(第一层为源节点)。在基于直方图的网络编码数据传输方式和基于能力分层下,采用速率控制 and 没有速率控制的方法下的性能,结果如图7(a)和图7(b)所示。

图7(a)显示基于能力分层下平均吞吐率随层次数增加首先近似指数降低,然后接近平稳状态。这是由于采用能力感知的分层方式,高上传带宽的节点位于上层,因此下层节点可以获得更多的下载带宽。图7(b)显示端到端延迟随着层次数增大而上升,然而基于能力分层和速率控制下端到端延迟随着层次数增大上升幅度远小于其余方式,且在层次数>5后的端到端延迟接近平稳状态,这是由于速率控制机制同时考虑节点间的延迟,增加低延迟节点的下载带宽,在全局范围提升高优先级节点获取数据的平均速率。从图7(a)和图7(b)可知CORE通过速率控制机制实现了按照能力区分方式提供带宽,同时有效提高了系统范围的吞吐率。

8 结论(Conclusion)

针对异构节点环境下实现基于覆盖网的快速数据分发问题,提出了基于能力区分的拓扑构建和速率控制的网络编码组播协议CORE。CORE协议通过充分利用异构节点能力并提供区分的服务质量鼓励节点提供高的上传能力,同时提高全局范围的吞吐率,降低节点获取数据的延迟。模拟测试显示CORE协议具有高度的扩展性,较高的吞吐率和低端到端延迟,以及较低计算开销等优点。

参考文献(References)

- [1] Zheng Z, Wang Y, Ma X. PeerChatter: A Peer-to-Peer Architecture for Data Distribution over Social Networks[J]. Information-An International Interdisciplinary Journal, 2011, 15(1): 259-266.
- [2] E. Adar, B. Huberman. Free riding on Gnutella[J]. FirstMonday, 2010, 5(10): 33-38.
- [3] Richard T. B. Ma, Sam C. M. Lee, John C. S. Lui. Incentive and Service Differentiation in P2P Networks: A Game Theoretic Approach[J]. IEEE/ACM Transactions on Networking, 2012, 14(5): 978-991.
- [4] Chuan Wu, Baochun Li. Diverse: Application-Layer Service Differentiation in Peer-to-Peer Communications[J]. IEEE Journal on Selected Areas in Communications, 2011, 25(1): 222-234.
- [5] F. Clevnet, P. Nain, K. W. Ross. Multiclass P2P Networks: Static Resource Allocation for Bandwidth for Service Differentiation and Bandwidth Diversity[C]. In Proceedings of the Performance, 2009.
- [6] Zongpeng Li, Baochun Li, Lap Chi Lau. On Achieving Maximum