

基于节点优先级服务的混合P2P网络性能分析

刘琳, 任子清

燕山大学, 理学院, 河北 秦皇岛

收稿日期: 2022年2月21日; 录用日期: 2022年3月15日; 发布日期: 2022年3月22日

摘要

本文为了抑制P2P网络中的“搭便车”现象, 提高对等网络的服务质量, 请求节点被分为高优先级请求节点和低优先级请求节点。提出了一种具有负顾客和异步休假策略的抢占优先M/M/c排队模型, 通过拟生灭过程和矩阵几何解的方法求解系统的稳态分布。通过数值实验, 分析系统性能指标随系统参数变化的趋势。最后构造两类请求节点的个人效益函数, 结果表明高优先级请求节点的个人效益高于低优先级请求节点个人效益。

关键词

P2P网络, 搭便车, 抢占优先权, 矩阵几何解, 数值实验

Performance Analysis of Hybrid P2P Networks Based on Node Priority Service

Lin Liu, Ziqing Ren

School of Science, Yanshan University, Qinhuangdao Hebei

Received: Feb. 21st, 2022; accepted: Mar. 15th, 2022; published: Mar. 22nd, 2022

Abstract

In order to suppress the “free riding” phenomenon in P2P networks and improve the service quality of P2P networks, the requesting nodes are divided into high priority requesting nodes and low priority requesting nodes. A preemptive priority M/M/c queuing model with negative customers and asynchronous vacation strategy is proposed. The steady-state distribution of the system is solved by quasi birth-and-death process and matrix-geometric solution. Through numerical experiments, the change trend of system performance index with system parameters is analyzed. Finally, the personal expected return function of two types of request nodes is constructed. The results show that the personal expected income of high priority requesting nodes is higher than that

of low priority requesting nodes.

Keywords

P2P Network, Free Riding, Preemptive Priority, Matrix-Geometric Solution, Numerical Experiment

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

P2P 系统是一种分布式系统, 部分或完全不需要中央服务器[1]。是一个依赖于连接端节点的计算能力和带宽的系统, 而不是集中式的。具有分布式控制、自组织和对称通信的特征。P2P 系统有多种类型, 主要用于大规模内容分发、文件共享、平台共享、通信、分布式计算和协作。在 P2P 系统中, 由于使用系统和其他对等点的资源, 对等方需要通过共享其资源来添加到系统中。然而, 在众多 P2P 系统中, 大量对等点不愿共享资源, 只从系统中获取记录而不返回任何记录的客户端, 称为搭便车者。

有几项测量研究证实了 P2P 网络中存在搭便车者。Handurukande 等人[2]在 edonkey 进行的跟踪分析中, 大约 80%的对等点是搭便车的。Adar 等人[3]测量研究结果表明, 网络中 25%的对等方提供了 Gnutella 中约 99%的资源。此外, Asvanund 等[4]发现在 Gnutella 0.6 中存在搭便车现象。搭便车降低了 P2P 系统的有效性、健壮性, 加剧了网络拥塞, 导致系统不稳定。统计数据显示, 在 Gnutella 0.6 中, 25%的节点提供了大约 99%的资源, 7%的节点贡献了比其他节点更多的文件[5]。70%的对等点根本不共享任何文件, 25%的对等点提供了 99%的网络查询命中率[6]。因此, 为了提高 P2P 网络的服务质量, 必须有效抑制搭便车行为。

目前, 许多学者针对搭便车现象都提出了自己的解决方案。Shin 等人[7]提出了一种通用的激励机制 T-Chain, 用于加强参与者之间的合作。T-Chain 使合作成为强制性的, 并允许新来者将他们的第一个接收文件上传到另一个对等方。T 链的简单性促进了参与者之间的合作, 并且可以很容易地适应其他应用或协议。Oualhad 等人[8]提出了一种基于声誉的概率资源分配方案。为信誉较高的节点提供了选择偏好, 同时也为彼此信誉不好的节点之间的交互提供了一定的有限概率。这避免了上述对等点之间的断开连接。Alotibi 等人[9]提出了一种信用方法来防止 P2P 网络中的搭便车者。该方法使用基于点的免费搭车者检测来限制他们的活动, 并通过提供下载优先级来奖励最合作的节点, 以减少他们的下载时间。

本文提出一种节点优先级服务的策略来应对搭便车行。在实际应用中, 请求节点根据其贡献值可分为低优先级请求节点和高优先级请求节点, 高优先级请求节点抢占低优先级请求节点的服务, 以此来减少低优先级请求节点的个人效益, 提高网络运行的稳定性。研究 P2P 网络中具有负顾客和异步休假的抢占优先级 M/M/c 排队模型, 建立了一个三维连续时间 Markov 随机模型。利用 QBD 过程和矩阵几何解构造系统性能指标, 得到了排队模型的稳态分布, 并通过数值实验分析了参数变化对 P2P 网络系统性能指标的影响。定义了个人效益函数, 并详细分析了两类请求节点的个人效益。

2. 混合对等网络

本文所考虑的混合 P2P 网络模型是由集中式网络模型和分布式网络模型结合构成。如图 1 所示, 分布式网络是由簇与簇之间连接组成的, 而簇是一个小型的集中式网络, 节点根据物理位置相近的原则组

织在一起。簇中含有两种节点，一种是普通节点，另一种是超级节点。普通节点是网络中的请求节点，只有发布、查找资源信息和维护资源的基本能力。超级节点是一个服务节点，充当该簇中的资源服务器，接收来自节点的连接和资源处理请求，并将处理结果返回给请求节点。

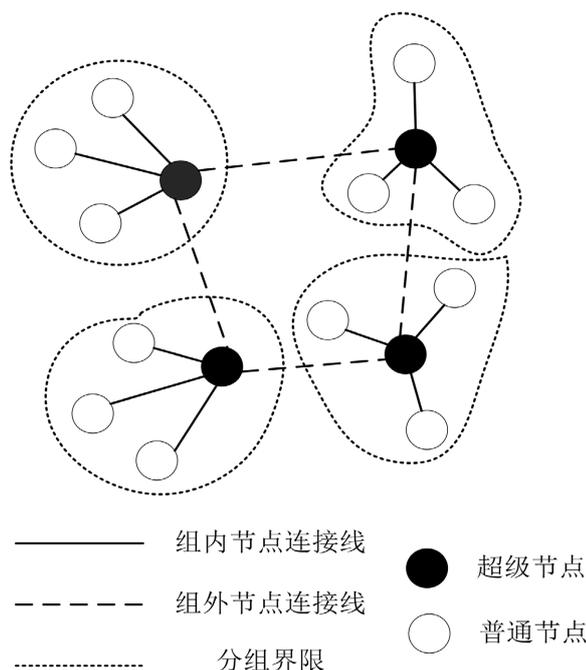


Figure 1. Node dynamic grouping diagram

图 1. 节点动态分组示例

在此模型上，引入节点服务请求队列请求服务的概念。超级节点根据请求节点的贡献值将其划分到相应的队列，包括搭便车请求节点队列和合作请求节点队列。搭便车请求节点加入混合 P2P 网络后的主要目的是下载所需的资源。当它下载完资源时，就会离开网络，当它需要再次下载资源时，它会再次加入这个网络。合作请求节点不仅下载资源，还上传资源为其他节点服务，而且不频繁加入和离开混合 P2P 网络。为了鼓励节点更稳定地留在网络中，提高网络的服务质量，合作请求节点共享资源被优先考虑。合作请求节点队列的优先级高于搭便车请求节点队列。为了降低系统能耗，当没有节点请求服务时，簇中的服务节点会进行异步休假。

3. 建模分析

在混合对等网络中，将搭便车请求节点和合作请求节点抽象成两类顾客，两类请求节点请求服务的过程抽象为顾客到达的过程，将服务节点抽象为服务台，节点为用户服务的过程抽象为服务台服务的过程。

3.1. 模型描述

1) 假设搭便车请求节点和合作请求节点的到达间隔服从参数为 $\lambda_1, \lambda_2 (\lambda_1, \lambda_2 > 0)$ $\lambda (\lambda > 0)$ 的指数分布。合作请求节点对搭便车请求节点具有抢占优先权，即当服务节点被搭便车请求节点占用且无空闲服务节点时，先到达的合作请求节点抢占正在服务的搭便车请求节点的服务，搭便车请求节点退回其队首处于等待状态，当所有服务节点均被合作请求节点占用时，新到达的合作请求节点需要等待排队。搭便

存在左零向量。当 QBD 正常返时, 稳态分布满足如下方程组

$$\begin{cases} (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_c) B[\mathbf{R}] = \mathbf{0}, \\ \sum_{i=0}^{c-1} \boldsymbol{\pi}_i \mathbf{e} + \boldsymbol{\pi}_c (\mathbf{I} - \mathbf{R})^{-1} \mathbf{e} = \mathbf{1}, \\ \boldsymbol{\pi}_i = \boldsymbol{\pi}_c \mathbf{R}^{i-c}, i \geq c. \end{cases}$$

其中 \mathbf{e} 是维数为 $(c+1)(d+1)$ 且所有元素均为 1 的列向量, \mathbf{I} 是维数为 $(c+1)^2(d+1)$ 单位矩阵。

对于上述分析证明过程, 采用了文献[10] [11]中提到的矩阵 - 几何解的方法。矩阵的表达相对复杂, 所以很难直接给出速率矩阵 \mathbf{R} 的分析表达。因此, 采用高斯 - 赛德尔迭代法来求出速率矩阵 \mathbf{R} 的近似解。

3.3. 性能指标

根据上述分析可以得到各性能指标:

1) 搭便车请求节点的平均队长为

$$E(L_1) = \sum_{i=0}^{\infty} iP(L_1 = i) = \sum_{i=1}^{\infty} i \left(\sum_{j=0}^c \sum_{k=0}^d \pi_{i,j,k} \right).$$

2) 合作请求节点的平均队长为

$$E(L_2) = \sum_{j=0}^c jP(L_2 = j) = \sum_{i=0}^{\infty} j \left(\sum_{k=0}^d \pi_{i,j,k} \right).$$

利用排队理论中的利特尔定律, 可以推导出请求节点的平均逗留时间。

3) 搭便车请求节点的平均逗留时间:

$$E(W_1) = \frac{1}{\lambda_1} \sum_{i=1}^{\infty} i \left(\sum_{j=0}^c \sum_{k=0}^d \pi_{i,j,k} \right).$$

4) 合作请求节点的平均逗留时间:

$$E(W_2) = \frac{1}{\lambda_2} \sum_{i=0}^c \sum_{j=1}^c j \left(\sum_{k=0}^d \pi_{i,j,k} \right).$$

4. 数值实验

在实际生活中, 性能指标常随系统参数的变化而变化, 用 MATLAB 进行数值分析来刻画性能指标随系统参数的变化趋势。令 $c=10, \lambda_1=1, \lambda_2=1, \mu_1=1, \mu_2=3, \varepsilon=1$, 图 2、图 3 显示了休假参数 θ 对搭便车请求节点的平均逗留时间 $E(W_1)$ 和具有不同 d 的高优先级请求节点的平均逗留时间 $E(W_2)$ 的影响, 其中 d 是系统允许的最大服务节点数。对于固定 θ , 当 $d=4$ 时, 搭便车请求节点的平均逗留时间 $E(W_1)$ 和合作请求节点的平均逗留时间 $E(W_2)$ 最短。当 $d=\{4,5,6,7\}$, 合作请求节点的平均逗留时间 $E(W_2)$ 随休假参数 θ 的增加而减少。由于休假参数 θ 越大, 平均休假时间越短, 服务节点处于服务状态的概率越高, 合作请求节点的平均停留时间 $E(W_2)$ 越短。当 $d=\{5,6,7\}$ 时, 搭便车请求节点的平均逗留时间 $E(W_1)$ 随着休假参数 θ 的增加而增加。当 $d=4$ 时, 搭便车请求节点的平均逗留时间 $E(W_1)$ 随着休假参数 θ 的增加而减少。由于合作请求节点的抢占服务, 搭便车请求节点的平均逗留时间 $E(W_1)$ 将随着休假参数 θ 的增加而减小。在不同休假参数的情况下, 搭便车请求节点的平均逗留时间的数值结果如表 1 所示, 合作请求节点的平均逗留时间的数值结果如表 2 所示。

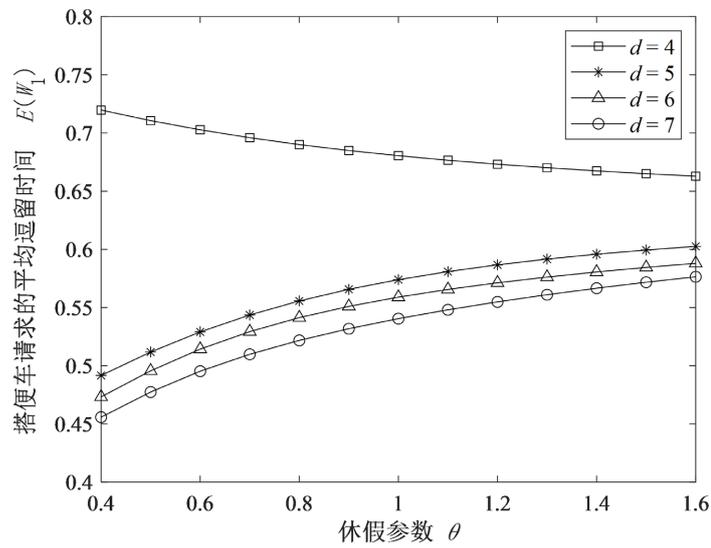


Figure 2. Average sojourn time $E(W_1)$ of free riding request nodes

图 2. 搭便车请求节点的平均逗留时间 $E(W_1)$

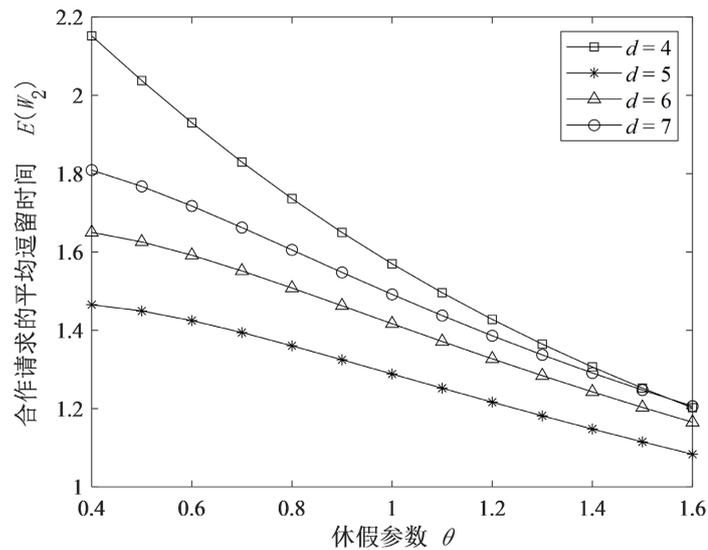


Figure 3. Average sojourn time $E(W_2)$ of cooperation request nodes

图 3. 合作请求节点的平均逗留时间 $E(W_2)$

Table 1. Average sojourn time $E(W_1)$ of free riding request nodes

表 1. 搭便车请求节点的平均逗留时间 $E(W_1)$

	$\theta = 0.4$	$\theta = 0.6$	$\theta = 0.8$	$\theta = 1$	$\theta = 1.2$	$\theta = 1.4$	$\theta = 1.6$
$d = 4$	0.7197	0.7027	0.6900	0.6805	0.6731	0.6674	0.6629
$d = 5$	0.4917	0.5291	0.5557	0.5740	0.5867	0.5958	0.6026
$d = 6$	0.4733	0.5143	0.5413	0.5589	0.5712	0.5806	0.5880
$d = 7$	0.4558	0.4953	0.5218	0.5405	0.5549	0.5666	0.5764

Table 2. Average sojourn time $E(W_2)$ of cooperation request nodes**表 2.** 合作请求节点的平均逗留时间 $E(W_2)$

	$\theta=0.4$	$\theta=0.6$	$\theta=0.8$	$\theta=1$	$\theta=1.2$	$\theta=1.4$	$\theta=1.6$
$d=4$	2.1518	1.9301	1.7361	1.5697	1.4276	1.3062	1.2018
$d=5$	1.4651	1.4246	1.3602	1.2882	1.2163	1.1476	1.0833
$d=6$	1.6501	1.5917	1.5079	1.4167	1.3270	1.2427	1.1651
$d=7$	1.8091	1.7174	1.6051	1.4916	1.3859	1.2908	1.2060

5. 纳什均衡

本节我们将通过建立个人效益函数, 来分析研究两类请求节点的均衡行为。假设搭便车请求节点和合作请求节点在系统中得到服务后的效益为 R_1 和 R_2 , 在系统中单位等待时间产生的损失为 C_1 和 C_2 。搭便车请求节点和合作请求节点从进入系统开始, 到服务结束为止所获得的个人效益函数为 U_1 和 U_2 , 具体表达式如下:

$$U_1 = R_1 - C_1 E(W_1),$$

$$U_2 = R_2 - C_2 E(W_2).$$

假设 $c=7, d=3, \lambda_1=\lambda_2=1, \mu_1=\mu_2=1, \varepsilon=1, R_1=R_2=50, C_1=C_2=10$, 即在两类请求节点的到达率, 服务率, 效益和损失是相等的情况下, 图 4 反映了两类请求节点个人效益随休假参数 θ 的变化大小。两类请求节点个人效益随着休假参数 θ 的增大而增大, 这是因为随着休假参数 θ 的增大, 服务节点处于异步休假概率增大, 请求节点的平均逗留时间减小, 所以请求节点的个人效益增大。我们还可以看出 U_2 的函数线始终在 U_1 函数线的上方, 无论休假参数是多少, 合作请求节点的个人效益都是高于搭便车请求节点的个人效益。表 3 显示了 U_1, U_2 具体值。

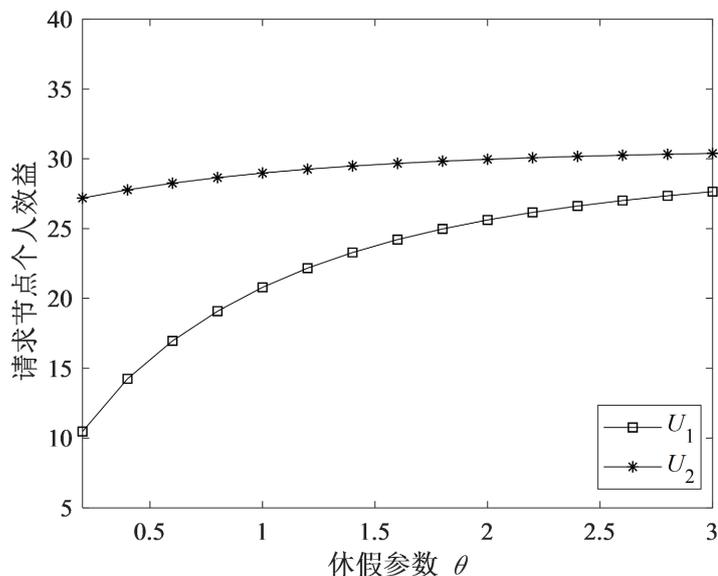
**Figure 4.** Request node individual benefits**图 4.** 请求节点个人收益

Table 3. Two types of request nodes for individual benefits**表 3.** 两类请求节点的个人效益值

	$\theta = 0.5$	$\theta = 1$	$\theta = 1.5$	$\theta = 2$	$\theta = 2.5$	$\theta = 3$
U_1	15.6967	20.7887	23.7696	25.6162	26.8205	27.6441
U_2	28.0229	28.9796	29.5787	29.9614	30.2151	30.3901

6. 结论

本文主要针对混合 P2P 网络中的搭便车现象, 克服其对网络不稳定的影响。提出节点优先级服务的策略, 将请求节点划分成合作请求节点和搭便车请求节点。首先在 M/M/c 排队模型的基础上, 引入负顾客、异步休假和抢占优先权等策略, 构造了相应的排队模型。然后建立三维 Markov 链, 利用拟生灭过程和矩阵几何解法得到了模型的平稳分布, 进而给出系统中请求节点的平均队长、平均逗留时间和系统能耗等重要性能指标。最后利用 MATLAB 数值分析, 描绘出参数变化对混合 P2P 网络系统各性能指标的影响, 并通过构造个人效益函数, 得到了混合 P2P 网络中请求节点的个人效益。结果表明合作请求节点的个人效益高于搭便车请求节点的个人效益, 有效的抑制了搭便车行为, 提高了对等网络的稳定性, 保证了合作请求节点的服务质量。

参考文献

- [1] Oluwatope, A.O., Ojoolayiwola, O.E., Sossou, E.H., et al. (2020) Ayo Game Approach to Mitigate Free Riding in Peer-to-Peer Networks. *Journal of King Saud University-Computer and Information Sciences*. <https://doi.org/10.1016/j.jksuci.2020.09.015>
- [2] Handurukande, S.B., Kermarrec, A.M., Le Fessant, F., et al. (2006) Peer Sharing Behaviour in the eDonkey Network, and Implications for the Design of Server-less File Sharing Systems. *AcmSigops Operating Systems Review*, **40**, 359-371. <https://doi.org/10.1145/1218063.1217970>
- [3] Adar, E. and Huberman, B.A. (2001) Free Riding on Gnutella. *First Monday*, **5**, No. 10. <https://doi.org/10.5210/fm.v5i10.792>
- [4] Asvanund, A., Clay, K., Krishnan, R., et al. (2004) An Empirical Analysis of Network Externalities in Peer-to-Peer Music-Sharing Networks. *Information Systems Research*. <https://doi.org/10.2139/ssrn.433780>
- [5] Yahaya, M.O. (2015) Mitigating Free Riding in Peer-To-Peer Networks: Game Theory Approach. *Nigerian Journal of Technology*, **34**, 375. <https://doi.org/10.4314/njt.v34i2.23>
- [6] Shahriar, I., Qiu, D. and Jaumard, B. (2017) Modeling of Free Riders in P2P Live Streaming Systems. *International Conference on Computing, Networking and Communications (ICNC)*, Silicon Valley, CA, 26-29 January 2017, 729-734. <https://doi.org/10.1109/ICCNC.2017.7876220>
- [7] Shin, K., Joe-Wong, C., Ha, S., et al. (2017) T-Chain: A General Incentive Scheme for Cooperative Computing. *Transactions on Networking*, **25**, 163-174. <https://doi.org/10.1109/TNET.2017.2685560>
- [8] Oualha, N. and Roudier, Y. (2008) Validating P2P Storage Audits with Evolutionary Game Theory. *Lecture Notes in Computer Science*, **5343**, 47-58. https://doi.org/10.1007/978-3-540-92157-8_5
- [9] Alotibi, B., Alarifi, N., Abdulghani, M., et al. (2019) Overcoming Free Riding Behavior in Peer-to-Peer Networks Using Points System Approach. *Procedia Computer Science*, **151**, 1060-1065. <https://doi.org/10.1016/j.procs.2019.04.150>
- [10] 田乃硕, 岳德权. 拟生灭过程与矩阵几何解[M]. 北京: 科学出版社, 2002: 39-77.
- [11] Neuts, M.F. (1981) *Matrix-Geometric Solutions in Stochastic Models*. Johns Hopkins University Press, Baltimore, 36-66.