

# 基于自信息量算法构建混合式 P2P 网络

池 溢 高志伟

(石家庄铁道大学 石家庄 050031)

**摘 要** 在大多数研究中,推选超级节点时重点考虑节点的性能是否适合作超级节点,而没有考虑当有节点从网络中退出或加入时,网络重新构建的开销。在选择超级节点的算法方面,很少有算法同时涉及节点的容量和在网络中的在线时间这两个参数。本算法在 Alberto Montresor 算法的基础上进行了改进,由信息量概念入手,将节点的容量和在线时间两个因素有机地结合起来,设计了自信息量算法模型。试验结果表明,自信息量模型网络构建的速度和单一使用节点容量构建的速度相仿,但自信息量模型推选的超级节点相对稳定,减少了网络构建的频数,从而减少了网络维护开销。该模型还能够适应一些极端情况,如试验中有 50% 的节点离开时,仍可较快地构建网络,说明该算法具有一定的鲁棒性。

**关键词** 超级节点,节点的容量,在线时间,自信息量算法

**中图分类号** TP393 **文献标识码** A

## Self-information Algorithm for Building Hybrid P2P Network

CHI Yi GAO Zhi-wei

(Shijiazhuang Tiedao University, Shijiazhuang 050031, China)

**Abstract** In most studies, the selection of super-peer only considers whether the performance of nodes is suitable for super-peers, without taking into account the speed of the network to re-build, when a node is out of or accession from the network. In the algorithm study, the on-line time and capacity was less considered at the same time. The algorithm was improved based on Alberto Montresor algorithm. From the concept of the self-information, combined with the node capacity and on-line time, the algorithm of super-peer selection was designed. The results show that the speed of building network for the self-information algorithm is similar with single-used capacity, and selected super-peers were more stable. The self-information algorithm will effectively reduce the frequency of building network, thereby reduce the costs of the network traffic. For some catastrophic situations, for example, removing 50% of nodes from the network, it can be quickly built. This indicates that the network built by our algorithm has robustness.

**Keywords** Super-peer, Node capacity, On-line time, Self-information algorithm

## 1 引言

近几年来 P2P 网络研究迅速升温,各方面的软件层出不穷。与传统的 C/S 模式相比,P2P 技术有非中心化、可扩展、健壮性等优点,因此具有较广泛的发展空间。而混合式 P2P 结构结合了集中式和分布式的优点,已成为 P2P 网络发展的主流趋势<sup>[1,2]</sup>,由此推选合适的超级节点的算法成为目前研究的热点课题之一。

Yang 和 Molina<sup>[3]</sup>探讨了基于超级节点网络设计的一般原理,但该文没有提供具体的超级节点选择算法。Montresor<sup>[4]</sup>提出了一种新的通用机制,其用于建立和维护基于超级节点的覆盖拓扑结构。这种机制是基于著名的 gossip 范例,节点与随机选择的节点之间交换信息,最终推选出性能最好的节点作为超级节点。基于该协议选取的超级节点是大容量的节点,能够处理节点加入和离开系统的连续流动。文献<sup>[5]</sup>提出了一种基于信誉的超级节点选择算法 RASSA,该算法根据节点提供资源的质量计算节点的信誉值,从信誉高的

节点中选择机器性能好、在线时间长的节点作为超级节点。信誉较高的节点指的是那些在系统中提供可信服务的节点,相比信誉较低的节点,其履行超级节点职责的能力更值得信赖,更适合充当超级节点。

H2O<sup>[6]</sup>是一种非结构化超级节点选取方法。超级节点向邻居广播自己的信息,当普通节点发送查询请求时,从已存储的超级节点中选择一个最好的超级节点发送请求。文献<sup>[7]</sup>提出了 e\* 算法,用以构建能感知节点可连接性的 Overlay 网络。e\* 算法着重考虑节点的可连接性,以此来对覆盖网络进行优化,该算法能够明显提高系统在响应时间方面的表现。文献<sup>[8]</sup>增加了对节点范围的考虑,在该协议的实验网络中,节点参数范围在 10000 左右,显然不宜在大型的网络中采用该协议,因而限制了算法的应用范围。文献<sup>[9]</sup>在兴趣相同的节点中推选出超级节点,但在实验中设计的兴趣参数只有 3 种,显然与实际情况不相符,其也没有考虑参数增加后是否对网络构建开销产生影响。

从目前比较流行的推选超级节点算法中不难看出,在大

池 溢(1979—),女,硕士生,主要研究方向为网络构建、网络安全,E-mail:chiyi@foxmail.com.

多数研究中推选超级节点时,都只考虑节点的性能是否适合作超级节点,很少考虑节点从网络中退出或加入时,网络重新构建的速度和重新构建网络的花费开销。随着网络飞速的发展,网络中的节点规模不断增加,网络的动态性是一个不可忽视的问题。在选择超级节点的算法方面,很少有算法同时涉及节点的容量和在网络中的在线时间这两个参数。本项研究由信息量概念入手,将节点的容量和节点的在线时间两个因素有机地结合起来,设计了自信息量算法模型。它充分考虑了网络中节点的扩充和网络的动态性,能更好地适应网络的发展趋势。试验结果表明,利用本文提出的算法推选出的超级节点相对稳定,有效地减少了动态节点对网络构建速度的影响,从而减少网络中的流量花费开销。

## 2 试验模型

本文使用的模拟器为 Java 语言编写的 PeerSim0.3<sup>[10]</sup> 仿真试验平台。以此平台为基础,开发了自信息量协议的仿真程序,采用 Newscast 协议<sup>[11]</sup> 随时捕捉网络中节点的状态。在模拟器中每个节点都是相对独立的,一个节点的加入或离开不会对其他节点产生影响,因此可以很好地模拟现实网络中节点加入或退出的状态。PeerSim 仿真环境不但实现了上述相关特性,而且具有较强的可扩展性和动态性。仿真环境试验是基于组件的,构成这些组件的基本元素是已经定义好的 Java 对象<sup>[10]</sup>,部分数据参数设置如表 1 所列。

表 1 模拟试验的配置文件中参数的初始化值

参数名称	参数值
SIZE	40000
MAXCAPACITY	180
MAXTIME	4000
DEGREE	30
REDUCED_DEGREE	30
ATTEMPTS	30
RATIO	1
LIMIT	0.95
WHEN	30
CRASH	0.90
LAMD	0.02
ALPHA	1.8

自信息量算法协议拓扑结构的主要特征:每个客户端恰好和一个超级节点相连接,超级节点通过随机网络相连。由此协议能够找到一个节点个数较少,而且在线时间较长的超级节点集合,并且这些超级节点能够覆盖剩下的作为客户端的节点。采用这样的拓扑结构可容易地实现文件共享,也可以减少由应用程序带来的流量。

该算法构建的混合式 P2P 网络中,超级节点必须赋予两个互斥角色中的一个,即超级节点或客户端来覆盖网络。节点间的邻居关系取决于该节点的角色,超级节点能与其他超级节点相连,也能够跟它所负责的客户端相连,而客户端只能连接到负责管理它的超级节点。

产生的拓扑结构,以容量和时间的自信息量作为推选标准。节点如果有较大的容量和较长的在线时间,则被认为是好的候选超级节点。于是可以产生一个含有较少数量超级节点的拓扑,且超级节点在网络存在的时间较长。该模型结构采用两层异构拓扑结构,推选出来的超级节点的全部容量足以覆盖所有作为客户端的节点,并且可以有效地限制活跃节点作为超级节点,从而减少因构建网络的次数带来的开销。

需要说明的是,在静态网络中,目标拓扑可获取;但在动态的情况下,实际的拓扑仅仅是近似的。

现实网络中的节点是异构的,它们的计算及存储能力以及连接带宽等都会有所不同。在模拟的过程中,为了能够从客户端加入的节点中分辨出有能力作为超级节点的节点,用  $C_n$  作为节点  $n$  的参数,代表其容量。换句话说,将网络中节点的异构性抽象地表示为容量这一参数,以此来简化模型。在实际网络中此值可在运行中计算,所以在模型中假定每个节点的容量参数是已知的。用  $T_n$  作为节点  $n$  的另外一个参数,代表其在线时间。为了能更好地模拟现实网络,先给参数赋一个初始值,在随后的模拟过程中累计计时。从网络中移除的节点渐渐地被遗忘,新加入的节点通过该节点先前在网络中的口碑来决定该节点的在线时间。

## 3 模型参数及算法

### 3.1 节点的容量

Barabasi 和 Albert 提出的无标度网络模型(BA 模型)<sup>[12]</sup>,被认为是目前最能表现复杂的网络模型,其特征是:择优连接和增长,而且增长是无限的;不考虑节点的寿命,即认为节点的寿命是无限的。分析模型的方法均假设了节点在  $t=1s, 2s \dots$  时间间隔离散地进入系统。

在动态的网络中,节点的容量是有规律可循的,容量有明显的分布特征,通常服从幂分布或均匀分布,此处只讨论节点容量服从幂分布<sup>[4]</sup>的情形。服从幂分布时,节点  $n$  有容量  $x$  的概率  $P(C_n=x)=x^{-\alpha}$ ,其中  $x \in [1, C_{max}]$ ,  $\alpha$  是分布参数,通常参数  $\alpha=2$ 。网络初始化时,按照节点容量的分布,给每个节点赋初值,节点容量为静态变量。

### 3.2 节点的在线时间

为了使模拟结果更接近网络的真实情况,网络初始化时,节点的直接在线时间 and 间接在线时间是相同的。间接在线时间是在初始值的基础上,通过后期对节点在网络中模拟的过程对节点在网络间进行累积。

直接在线时间的计算:网络中节点的加入或退出被认为服从泊松分布,节点的直接在线时间则服从参数为  $\lambda$  的负指数分布<sup>[13]</sup>。概率密度函数

$$f(x) = \lambda e^{-\lambda x}, x > 0 \quad (1)$$

用极大似然估计法估计分布参数  $\lambda$ , 设  $x_1, x_2, \dots, x_n$  是一组随机样本值,代表节点的在线时,似然函数

$$L(\lambda) = \lambda^n \prod_{i=1}^n \exp(-\lambda x_i) = \lambda^n \exp(-\lambda \sum_{i=1}^n x_i) \quad (2)$$

$$\frac{d \ln L(\lambda)}{d \lambda} = \frac{n}{\lambda} - \sum_{i=1}^n x_i = 0$$

解出

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^n x_i} = \frac{1}{\bar{x}} \quad (3)$$

由此可知,只要粗略地估计出平均在线时间,即可由式(3)求出分布参数的估计值。例如本试验中设  $\bar{x}=500$ , 则参数  $\hat{\lambda}=0.002$ 。确定分布参数后,可以产生符合指数分布的随机数作为节点的在线时间,具体方法为:

分布函数

$$F(x_i) = 1 - \exp(-\lambda x_i) \quad (4)$$

在线时间

$$x_i = \frac{\ln(1 - F(x_i))}{-\lambda} \quad (5)$$

由随机变量抽样基本定理,有  $R=F(x)$ , 其中  $R$  是  $[0,1]$  上均匀分布的随机变量。由于  $1-R$  与  $R$  具有相同的分布, 因此式(5)可以写成

$$x = -\frac{\ln R}{\lambda} \quad (6)$$

利用 CommonRandom() 函数产生  $0 \sim 1$  之间的随机数, 即可由式(8)模拟出指数分布的在线时间。

当产生 1000 个随机数时,按降序排列,第 200 个数据 821 作为百分位点,则

$$P(x < 821) = 1 - \exp(-0.002 * 821) = 0.806$$

即小于这一数值的在线时间的概率为 80%, 大于这一数值的在线时间的概率为 20%。这一结果与文献[14]中给出的试验结果基本符合。

间接在线时间的计算: 间接时间的初始值为其邻居节的直接在线时间, 每个节点的邻居节点不只存放其邻居节点的 ID, 也存放其邻居节点的间接在线时间。模型采用的是环驱动模型, 假设每做一次周期循环需要 1s 的时间。在每次循环中都对周围邻居节点进行探测, 若探测到邻居节点 ID 与自己存放的邻居节点 ID 相同, 则在节点的间接在线时间上加 1s。

通过比较直接在线时间和间接在线时间, 在线时间可以由 eBay<sup>[15]</sup> 信誉模型计算得到。

### 3.3 选择超级节点算法的设计

在推选超级节点时, 同时考虑节点的容量和在线时间, 需要制定一个统一的标准。设节点的容量为  $C_n$ , 节点的在线时间为  $T_n$ , 最简单的方法是将两个参数进行线性组合, 得到参数  $\delta(C_n, T_n)$ , 然而这样的参数存在 3 个明显的问题:

- (1)  $C_n, T_n$  的量纲不同,  $C_n$  的单位是度,  $T_n$  的单位是时间;
- (2)  $C_n, T_n$  的数量级不同, 在本次试验中  $C_n$  的量级为  $10^2$ ,  $T_n$  的量级为  $10^4$ ;
- (3) 当  $C_n$  (或者  $T_n$ ) 很大,  $T_n$  (或者  $C_n$ ) 很小时,  $\delta(C_n, T_n)$  可能很大, 显然我们不希望选取这样的节点作为超级节点。

为了解决这一问题, 从自信息量的概念入手设计了一个新的参数。设网络中共有  $m$  个节点, 节点  $n$  具有容量  $C_n$ , 且在线时间为  $T_n$ , 网络中所有节点容量之和  $C_s = \sum_{n=1}^m C_n$ , 所有节点的在线时间之和  $T_s = \sum_{n=1}^m T_n$ , 则节点  $n$  具有容量  $C_n$  时成为超级节点的概率为

$$P_C = C_n / C_s \quad (7)$$

$C_n$  越大, 成为超级节点的概率就越大。同理, 节点具有在线时间  $T_n$  时, 成为超级节点的概率为

$$P_T = T_n / T_s \quad (8)$$

当某节点具有容量  $C_n$  和在线时间  $T_n$  时, 假设  $C_n, T_n$  独立, 则其联合自信息量为

$$I(C_n, T_n) = I(C_n) + I(T_n) = -\log P_C - \log P_T \quad (9)$$

为了避免在公式中出现  $C_s, T_s$ , 将上式变成

$$I(C_n, T_n) = -\log \frac{C_n}{C_s} - \log \frac{T_n}{T_s}$$

$$\text{令 } \delta(C_n, T_n) = \log C_n + \log T_n \quad (10)$$

$$\text{则 } I(C_n, T_n) = -\delta(C_n, T_n) + \log(C_s * T_s) \quad (11)$$

由于在同一时刻  $\log(C_s * T_s)$  为常数, 因此  $I(C_n, T_n)$  的变化只与  $\delta(C_n, T_n)$  有关, 于是可以用  $\delta(C_n, T_n)$  作为选取超级节点的条件。所不同的是, 如果依据自信息量选取超级节点, 则  $I(C_n, T_n)$  越小越好; 如果依据  $\delta(C_n, T_n)$  选取超级节点, 则  $\delta(C_n, T_n)$  越大越好。由于  $\delta(C_n, T_n)$  只与  $C_n, T_n$  有关, 与  $C_s, T_s$  无关, 因此式(10)比式(9)更具可操作性。

## 4 试验结果分析及性能评价

### 4.1 对网络构建速度的影响

应用单一节点容量参数  $C_n$  和自信息量  $\delta(C_n, T_n)$  分布进行试验, 比较两种算法选举出的超级节点对网络构建速度的影响, 结果如图 1 所示, 虚线表示用自信息量算法推选出的超级节点数, 实线表示用节点容量推选出的超级节点数。从图 1 可见, 两个参数的试验结果相当吻合, 当实验运行到第 10 个周期左右时, 两种算法都基本确定了网络中超级节点的数量。采用自信息量算法推选超级节点, 构建网络的速度比单一使用节点容量构建网络的速度慢(从图 1 可以看出, 大约慢 3 个周期左右), 而且网络构建后超级节点的数量略增加, 但对于大型的网络来说, 这些影响是可以忽略的<sup>[4]</sup>。

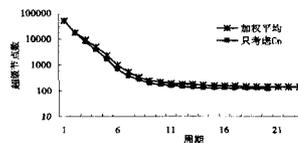


图 1 不同参数对网络构建速度的影响

### 4.2 网络中活跃节点的滤除

图 2 给出节点容量算法和自信息量算法推选出的超级节点的对比情况, 横坐标为节点的在线时间, 纵轴为节点的容量。试验中, 网络节点个数为 1000 个节点。图 2 中, 构建好后的客户节点用符号“+”表示, 方块“□”表示超级节点。

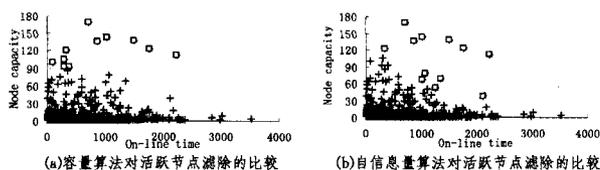


图 2

图 2(a) 给出了只采用容量作为推选超级节点的条件选出的超级节点分布情况。可以看出, 部分在线时间较少的节点被选中, 这些节点在网络中比较活跃, 可能是不稳定的。图 2(b) 为自信息量算法推选出的超级节点分布情况, 大多数超级节点集中在图形的中心部位, 说明推选出的超级节点具有较大的容量和较长的在线时间, 通常这类节点具有较好的稳定性。由此可见, 自信息量算法可以有效地避免选择活跃节点作为超级节点, 于是增加了构建网络的稳定性。

### 4.3 网络中的开销

超级节点加入或退出时需要网络进行重构, 活跃节点频繁加入或退出会增加网络重新构建的次数, 由此增加网络花费。设计自信息量算法的主要目的, 希望推选相对稳定的节点作为超级节点, 以节省这部分网络开销。图 3 给出节点容量(虚线)和自信息量(实线)两种不同算法构建网络时开销的对比试验结果。自信息量算法在推选超级节点时, 推选相对稳定的节点, 因此网络重构次数较少。仅以节点容量作为推选条件所推选出的超级节点活跃度较高, 网络不够稳定, 构建次数比自信息量算法的构建次数多。

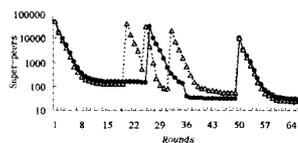


图 3 容量算法和自信息量算法的网络重构次数

图4给出了容量(虚线)和自信息量算法(实线)的开销对比,算法中的开销包括节点间消息交流花费和节点间客户端的转换花费。从图4不难看出,网络构建初期自信息量算法的花费并没有减少,但运行较长周期后,有节点退出网络时,自信息量算法的开销明显要小于只考虑容量的开销。由于自信息量算法可以有效地限制活跃节点的加入,减少构建网络的次数,因此其减少了构建网络的开销。

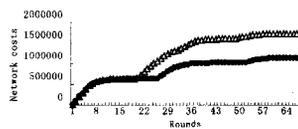


图4 容量算法和自信息量算法构建网络的花费

#### 4.4 算法的鲁棒性

为了考察自信息量算法在动态网络中面对灾难性突发事件的反应,试验运行到第30个周期时,将50%的超级节点从网络中移除,试验结果如图5所示。采用自信息量算法,网络中剩余的节点可以很快地重新选举出新的超级节点,以完成对拓扑网络的重新构建,这说明自信息量算法对于动态网络具有很好的鲁棒性。

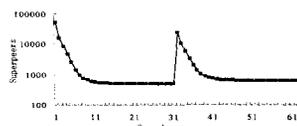


图5 自信息量算法的鲁棒性

#### 4.5 网络规模对算法的影响

现实中的网络规模都很大,网络中节点的数量会达到百万或千万,自信息量算法的构建速度是否受到网络大小的影响也是需要考虑的因素之一。图6给出了自信息量算法从1000到50000个节点对网络构建速度的影响。当网络中节点数在1000至20000之间时,随着节点数量的增加,其运行周期增加,即网络的构建速度随着节点数量的增加而减小。当网络中节点数为20000至50000时,随着网络规模的增加其对网络构建速度的影响逐渐减少,基本上呈对数变化趋势,大约运行到第25个周期就能够构建好一个混合式的拓扑网络。这一现象表明,随着网络节点数量的增加,自信息量算法不会对网络的构建速度产生很大的影响。

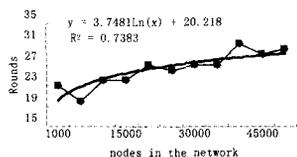


图6 网络规模大小对构建速度的影响

**结束语** (1) 本文将节点在线时间引入到P2P网络超级节点选取的研究中,提出了基于自信息量算法构建混合式的P2P网络模型。该模型能够在动态网络中结合节点的在线时间,通过自信息量算法来选取超级节点。

(2) 自信息量算法构建网络的速度比单一使用节点容量的构建速度稍慢,构建后的超级节点数略有增加,但对于大型的网络来说,这些影响是可以忽略的。

(3) 采用自信息量算法推选出的超级节点具有一定的容量和较长的在线时间,使得构建的网络相对稳定,减少了网络构建的频数,从而减少了网络维护开销。

(4) 算法具有一定的鲁棒性,可以较好地适应灾难性突发事件,例如有50%的节点离开网络,仍可较快地完成网络的构建。

(5) 在模型网络中,初始化网络的在线时间是网络中节点的在线时间。在现实网络中,对超级节点数量的要求以及网络中节点的特性都不尽相同,这些因素都可能造成模型结构和模型参数的变化。所以在实际应用中,可以根据现实网络的特点对模型参数进行修正。

#### 参考文献

- [1] 徐恪,叶明江,胡懋智. P2P技术现状及未来发展[J]. 中兴通讯, 2007,6(13):6-10
- [2] 罗杰文. Peer to Peer 综述[EB/OL]. <http://docs.huihoo.com/p2p/1/index.htm>,2005-11-03
- [3] Yang B, Molina H G. Designing a super-peer network[J]. Proceedings of the 19th International Conference Oil Data Engineering(ICDE'03),2004,49(6):49-60
- [4] Montresor A. A Robust Protocol for Building Superpeer Overlay Topologies[C]//Proceedings of the Fourth International Conference on Peer-to-Peer Computing. 2004,5:1-13
- [5] Liu Y M, Yang S B, Chen W M, et al. The reasarch of the Reputation-Aware SuperNode Selection Algorithm in P2P system. [J]. Journal of the Graduate School of the Chinese Academy of Sciences,2008,25(2):197-203
- [6] Lo V, Zhou Da-yi, Liu Yu-hong, et al. Scalable supernode selection in peer-to-peer overlay network[J]. Second International Workshop of Hot Topics in Peer-to-Peer Systems,2005,7(21):18-25
- [7] Wang W, Jin C, Jamin S. Network Overlay Construction under Limited End-to-End Reachability[J]. Proc of the IEEE INFOCOM, Miami, US,2005,8(22):2124-2134
- [8] Min Su-hong, Holliday J, et al. Optimal Super-peer Selection for Large-scale P2P[EB/OL]. <http://www.cse.scu.edu/jholliday/IsHIT06.pdf>,2006
- [9] Khan S, Tokarchuk L. Interest-based self organization in group-structured P2P networks[C]//Consumer Communications and Networking Conference, declared interest, super-peer network topology. 2009,6:(1-5)
- [10] Jesi G P. PeerSim HOWTO; Build a new protocol for the Peer-Sim 1.0 simulator [EB/OL]. <http://www.metz.supelec.fr/metz/personnel/altier/PagesPerso/Enseignement/3A/SOA/P2P/tutorial1.pdf>,2005-12-24
- [11] Jelasity M, Kowalczyk W, van Steen M. Newscast Computing [J]. Internal report IR-CS-006, Vrije Universiteit Amsterdam, Department of Computer Science,2003,9:213-237
- [12] Barabasi A-L, Albert R. Statistical mechanics of complex networks [J]. Rev Mod Phys,2002,9(74):47-97
- [13] 郭进利. 探讨复杂网络的新途径[J]. 系统工程理论与实践, 2006,10733-40
- [14] Saroiu S, Gummadi P K, Gribble S D. A Measurement Study of Peer-to-Peer File Sharing Systems [EB/OL]. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.7773&rep=rep1&type=pdf>,2003-04-14
- [15] Houser D, Wooders J. Reputation in auctions: theory and evidence from eBay[J]. J Econ Managet Strat, 2007,15(2):353-370