ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.2, February 2007, pp.381−390
DOI: 10.1360/jos180381

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种令 P2P 覆盖网络拓扑相关的通用方法*

邱彤庆, 陈贵海+

(计算机软件新技术国家重点实验室(南京大学),江苏 南京 210093)

A Generic Approach to Making P2P Overlay Network Topology-Aware

QIU Tong-Qing, CHEN Gui-Hai+

(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210093, China)

+ Corresponding author: Phn: +86-25-83686500, Fax: +86-25-83686500, E-mail: gchen@nju.edu.cn, http://cs.nju.edu.cn/~gchen

Qiu TQ, Chen GH. A generic approach to making P2P overlay network topology-aware. *Journal of Software*, 2007,18(2):381−390. http://www.jos.org.cn/1000-9825/18/381.htm

Abstract: With the help of distributed Hash table, the structured P2P (peer-to-peer) network has a short routing path and good extensibility. However, the mismatch between the overlay and physical network becomes the obstacle in the way of building an effective peer-to-peer system in a large-scale environment. In this paper, a generic, protocol-independent approach is proposed to solve this problem. This method is based on the swaps of peers. By discovering and performing the potential swaps that are beneficial to the match between overlay and physical network, it can reduce the average latency and improve the performance of the system. The experimental results show that the approach can greatly reduce the average latency of overlay networks. Moreover, the cost of overhead is controllable. Besides, if combining this approach with other protocol-dependent ones, the performance can be further improved.

Key words: P2P (peer-to-peer) network; overlay network; topology-aware

摘 要: 利用分布式哈希表,有结构的对等(peer-to-peer,简称 P2P)网络具备了较短的路由长度和较好的扩展性.然而,由此产生了覆盖网络和物理网络之间的不匹配问题,它严重阻碍了在大规模环境下建立有效的对等网络.提出一种通用的、协议无关的方法来解决该问题.该方法基于节点交换机制,通过发现并实施有利于覆盖网络和物理网络匹配的节点交换来降低网络时延、提高性能.实验表明,该方法在明显降低了覆盖网络的平均时延的同时,也保证了额外开销可控.此外,若与其他协议相关的方法相结合,系统性能还可以得到进一步提高.

关键词: 对等网络;覆盖网络;拓扑有关
中图法分类号: TP393 文献标识码: A

* Supported by the National Natural Science Foundation of China under Grant No.60573131 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.2006CB303004 (国家重点基础研究发展规划(973)); the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of the Ministry of Education of China (国家教育部高等学校优秀青年教师教学科研奖励计划); the Natural Science Foundation of Jiangsu Province of China under Grant No.BK2005208 (江苏省自然科学基金)

Received 2005-10-22; Accepted 2006-02-23

近年来,一些 P2P(peer-to-peer)系统相继被提出来,如 CAN(content-addressable network)[1],Chord[2],Pastry[3],Tapestry[4],Viceroy[5],Cycloid[6]等,它们为大规模 P2P 应用提供了自组织的基础设施.这类有结构的 P2P 系统是以分布式哈希表(distributed Hash table,简称 DHT)为基础的.它们使用一致性哈希函数(consistent Hash function)将数据和节点都均匀地映射到一个键值空间(key space,或称标号空间 identifier space).键值为 k 的数据存放在标号也为 k 的节点上;如果该节点不存在,数据就存放在标号与 k 邻近的节点上.通过合理的组织,对于任意数据的查询都可以在有限步(通常为 $O(\log n)$内,其中 n 为系统内节点个数)内完成.然而,由于每个节点被随机映射到一个节点标号,这样的映射过程丢失了很多物理网络的性质.因此,构建起来的逻辑上覆盖网络往往和物理网络不一致.物理上相距很远的点可能成为逻辑上的邻居;相反,物理上邻近的节点可能在覆盖网络中相距很远.通常把这种现象称为失配(mismatching),而相应的解决目标就是建立拓扑有关(topology-aware)的覆盖网络.

为了解决上述问题,通常的方法包括两个步骤[7]:(1) 搜集网络邻近信息;(2) 利用该信息构建或调整覆盖网络结构.为了说明目前相关工作的局限性,下面从这两个方面展开讨论.

(1) 搜集邻近信息

为了解决失配的问题,节点必须了解在物理上它和哪些节点相邻近.相应地,要搜集下层物理网络的邻近信息.一般有两种搜集信息的方式:地标聚类(landmark clustering)和泛洪或启发式搜索(flooding or heuristic searching).地标聚类的基本思想是:如果两个节点相邻近,那么它们与那些地标节点的距离应该相似.Ratnasam 等人[8]就利用该思想来优化 CAN.这种搜集方式的主要问题在于信息精确度不够高.同时,地标节点也存在着单点失效的问题[9].泛洪或启发式搜索的方法与 P2P 系统中搜索的过程相似,但目的不是为了查找数据,而是测量节点之间的时延.相对而言,这种方法的精度更高.可是,如果不合理地加以限制,这种探测性的搜索会产生很大的开销.因此,搜集邻近信息的主要问题在于如何在开销较小的情况下搜集到足够的信息.

(2) 利用邻近信息

当搜集完足够的信息之后,下一步就是如何利用这些信息构建或调整覆盖网络结构.大致上可以分为 3 种方法[10]:邻近路由(proximity routing)、邻近邻居选择(proximity neighbor selection,简称 PNS)和地理布局(geographic layout).邻近路由即在路由时,若下一跳有多种选择,则选择时延最短的一个.CAN[1]就使用了这样的优化方式.邻近邻居选择是在构建路由表时,从多个候选节点中选择距离最近的作为路由表项.Pastry[11]本身都有邻近邻居选择的机制.地理布局则是希望在地理位置上邻近的节点其标号也相似.Topologically-Aware CAN[8]就是这样的一个例子.然而,以上的方法都存在一个共同的问题,我们称其为协议相关,即没有哪一种方法可以很自由地应用到各种有结构的 P2P 系统中.比如,邻近路由要求下一跳有多种选择,邻近邻居选择也要求有多个候选节点作为路由表项,这样的限制使得它们无法应用到像 Chord[2]这样确定性的路由系统中.类似地,地理布局的方法只使用于 CAN 这样的系统[12],因为只有在 CAN 中,节点标号相似才意味着它们之间在逻辑上是邻近的.

为了解决上述问题,本文提出了一种令 P2P 覆盖网络拓扑有关的通用方法.通过周期性地调整节点标号,既保证了网络结构不变,又逐步降低了网络的平均时延,使得覆盖网络和物理网络逐步匹配.这种方法是协议无关的,可以在各种有结构的 P2P 系统中使用.实验表明,我们使用的方法能够显著降低网络平均时延,并且可以在其他协议相关方法的基础上进一步提高系统性能.

本文第 1 节描述基本思想和基本方法.第 2 节介绍一些开销控制机制.第 3 节通过实验来验证方法的有效性.第 4 节讨论其他相关工作.最后,第 5 节对全文作总结.

1 基本方法

1.1 基本思想

一个 DHT 网络(简称 DHT)是在物理网络的基础上构建的应用层覆盖网络.它可以看作是一个有向图 $G=(V,E)$,其中,V 是系统中节点的集合,E 是节点间逻辑连接的集合.集合 E 中存在边(u,v)(或者记为 $e_{uv}\in E$),表示节点 u 知道直接发送消息到节点 v 的方式(一般就是知道节点 v 的 IP 地址).若$(u,v)\in E$,则称节点 u 是节点 v 的

前驱节点,节点 v 是节点 u 的后继节点.两节点互为邻居节点.使用 DHT 的优点显而易见:首先,使用标号可以方便管理一个大规模的系统;其次,由于映射的过程是随机的,因此每个节点都保持了匿名性.但是,这样构建的覆盖网络无法与物理网络相匹配.图 1 显示了一个由 4 个节点构成的物理网络.其中,节点之间的连线用数字标识,表示了节点之间的时延.对应的两种覆盖网络如图 2 所示.圆周代表了整个标号空间,用虚线表示节点之间的逻辑连接*.假设覆盖网络上邻居节点之间的距离是它们之间的最短路径的长度.例如图 2(a)中 $A\rightarrow D$ 的距离以图 1 中 $A\rightarrow B\rightarrow D$ 来计算,等于 12.为了说明图 2(a)的覆盖网络是失配的,我们假设有一个查询从节点 A 到节点 C.它的时延是 10($A\rightarrow B\rightarrow C$)或者 23($A\rightarrow D\rightarrow C$),都远大于物理网络中的时延 3.出现这种情况的根本原因是,每个节点都和其标号捆绑在一起,当一个节点加入系统时,它在覆盖网络中的位置就不变了.我们的基本想法就是在保证 DHT 有效性的前提下,使得节点标号可变.为此,要满足以下的具体要求:

- 结构的保持.对节点标号的改变不应当改变 P2P 系统的结构.在本文的开始部分,我们提到大多数方法的共同问题就是它们都依赖于特定的协议.如果标号的改变破坏了原有的结构,就不可能不使用特定协议的信息来重构它.
- 安全性.对节点标号的改变不能是任意的,否则,系统将变得十分脆弱,很容易被黑客利用.同时,希望保持原有系统匿名性的特征.
- 可控的开销.节点标号的改变带来的开销应该是可控的.如果开销过大,该方法就不能达到很好的性能.

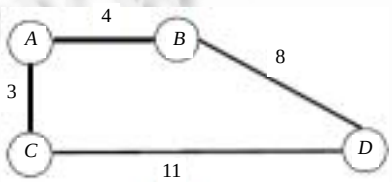
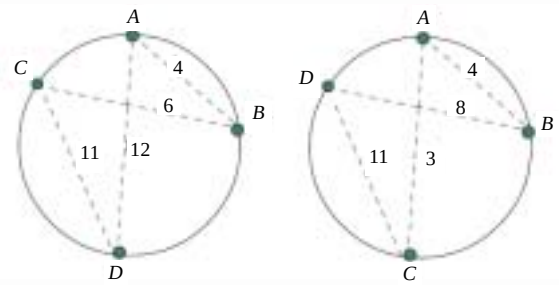


Fig.1 The physical network with four nodes

图 1 由 4 个节点构成的物理网络



(a) Mismatch overlay network (b) Overlay network after swap

(a) 失配的覆盖网络 (b) 交换后的覆盖网络

Fig.2 Overlay network

图 2 覆盖网络

1.2 交换操作

改变节点标号的基本操作是“交换”.交换节点标号是一种可以有效调整覆盖网络的方法.例如,在图 2(a)中交换节点 C 和 D 的标号,重构的覆盖网络就与物理网络完全匹配了(如图 2(b)所示).为了具体说明交换的方法,表 1 中列出一些表达的符号.我们假设节点 u 和节点 v 之间可能发生交换.节点 u 的邻居节点集定义如下:

$$N(u)=\{i|i\in V\wedge((u,v)\in E\vee(i,u)\in E)\} \quad (1)$$

Table 1 The notation table

表 **1** 符号表

Notation	Meaning
V	The set of all nodes in the system
E	The set of all edges in the overlay network
t_0	The time before the swap of node u and v
t_1	The time after the swap of node u and v
$N_{t_i(u)}$	The neighbor set of node u at t_i
$d(i,j)$	The latency between node i and j
L_{t_i}	The accumulated latency value of overlay
AL	The average latency of the overlay

* 为了方便说明,这里假设连接是无方向的.

为了简化对邻居节点的处理,将路由表作一定的扩展,不但记录后继节点,而且记录前驱节点.实际上,有许多 P2P 系统出于容错性的考虑,已经选择性地记录了一些前驱节点.因此,这种扩展带来的路由表规模的扩大是有限的.有一些路由对称的系统(如 CAN)甚至不需要任何扩展.在开始时,系统中每个节点 u 获取它的所有邻居节点的地址列表和初始的局部时延信息 $\sum_{i\in N_{t_0}(u)} d(u,i)$.然后,每隔一段固定的时间 T,节点 u 就周期性地探测一个随机节点 v.包含 TTL 字段的数据包用来实现这样的探测.开始时设置 $TTL=k$,每经过一个节点,TTL 减 1.当 $TTL=0$ 时,节点 v 被选中.此后,节点 u 和节点 v 交换它们的地址列表和初始信息.两节点分别计算交换后的局部时延信息 $\sum_{i\in N_{t_1}(u)} d(u,i)$ 和 $\sum_{i\in N_{t_1}(v)} d(v,i)$.然后,它们交换新的时延信息并独立计算差值 $Diff$,

$$Diff = \sum_{i\in N_{t_0}(u)} d(u,i) + \sum_{i\in N_{t_0}(v)} d(v,i) - \sum_{i\in N_{t_1}(u)} d(u,i) - \sum_{i\in N_{t_1}(v)} d(v,i) \tag{2}$$

如果 $Diff\le 0$,表明节点 u 和 v 之间的标号交换不能达到降低平均时延的效果,所以没有进一步的操作.如果 $Diff>0$,节点 u 和 v 就要发生交换操作,即交换节点的标号和路由表信息.此外,它们还要通知自己的邻居节点更改路由表信息,并重新计算初始的局部时延信息.由于我们扩展了路由表,这样的通知操作可以很方便地实现.即使没有扩展路由表,交换也可以看作是一系列的加入(join)和离开(leave)操作的集合.而这两个基本操作在任何 P2P 系统中都已经实现了.当然,使用加入和离开操作会引入一些不必要的维护开销.从这里也可以看到,对路由表的扩展不是必须的,而是出于方便实现和降低开销的考虑.

1.3 有效性分析

首先来证明节点交换的方法保持了原有的结构.

定理 **1**. 令 $G=(V,E)$表示 t_0 时刻的覆盖网络.不失一般性,假设节点 v_x 和 v_y 和在 $t_0\to t_1$ 时间段发生了交换(其中,x,y 为标号;$v_x,v_y\in V$),则在 t_1 时刻得到的图 $G'=(V',E')$和 t_0 时刻的图 $G=(V,E)$同构.

证明:易见$|V|=|V'|,|E|=|E'|$.先构造 V 和 V'之间的一一映射:

对于$\forall i\ne x,y,v_i\in V$ 对应于 $v_i\in V'$.

对于 $x,y,v_x\in V$ 对应于 $v_y\in V',v_y\in V$ 对应于 $v_x\in V'$.

再构造 E 和 E'之间的一一映射:设未交换的节点组成集合 V_1,交换的节点组成集合 V_2.

对于$\forall v_i,v_j\in V_1,e_{ij}\in V$ 对应于 $e_{ij}\in V',e_{ji}\in V$ 对应于 $e_{ji}\in V'$.

对于$\forall v_i\in V_1,\forall v_j\in V_2$,即 $v_j=v_x$ 或 $v_j=v_y,e_{ix}\in V$ 对应于 $e_{iy}\in V',e_{xi}\in V$ 对应于 $e_{yi}\in V'$;

同样地,$e_{iy}\in V$ 对应于 $e_{ix}\in V',e_{yi}\in V$ 对应于 $e_{xi}\in V'$.

对于$\forall v_i,v_j\in V_2,e_{ij}\in V$ 对应于 $e_{ji}\in V',e_{ji}\in V$ 对应于 $e_{ij}\in V'$.

由此可见,图 G 和图 G'同构. □

推论 **1**. 经过任意多次节点标号的交换,最终得到的覆盖网络对应的图和原图同构.

证明:因为同构关系是可传递的,由定理 1 可以很容易地推出,经过任意多次交换,仍然同构. □

既然经过任意次交换所得到的图和原图同构,就表明原有的结构得到了保持.原有的路由协议可以不加改动地应用在新的覆盖网络上.因此符合第 1.1 节提到的结构保持的要求.另外,由于仅仅是随机地交换节点标号,而不是任意改变标号,所以匿名性得到了保证,同时不容易被黑客利用.

其次要分析的是,这样的节点标号交换能否得到比较理想的覆盖网络.我们使用 stretch 作为衡量覆盖网络和物理网络匹配程度的标准.它是逻辑上的平均时延和物理平均时延的比值.而物理平均时延相对固定,因此主要是看逻辑平均时延的大小.以下讨论的都是逻辑时延.设任意两个节点之间时延之和为 $d(i,j)$,则平均时延(average latency,简称 AL)可以表示为

$$AL = \left(\sum_{i\in V}\sum_{j\in V} d(i,j)\right) \Big/ n^2 \tag{3}$$

下面分析两个节点 u 和 v 发生交换对平均时延的影响.假设交换前后节点数不变,只需考虑累积时延 L_{t_i} 的变化.下面两个公式显示了这种变化.

$$L_{t_0} = C + \sum_{i \in N_{t_0}(u)} \alpha_i d(u,i) + \sum_{i \in N_{t_0}(v)} \beta_i d(v,i) \tag{4}$$

$$L_{t_1} = C + \sum_{i \in N_{t_1}(u)} \gamma_i d(u,i) + \sum_{i \in N_{t_1}(v)} \delta_i d(v,i) \tag{5}$$

在公式(4)和公式(5)中,C 代表交换前后不变的部分.和式中的系数$\alpha_i,\gamma_i,\beta_i,\delta_i$ 代表计算任意两节点时延时用到对应链路的次数.注意到, $N_{t_0}(u)=N_{t_1}(v)$, $N_{t_0}(v)=N_{t_1}(u)$.此外,假设每条链路访问的次数均等,则有 $\alpha_i \approx \gamma_i$, $\beta_i \approx \delta_i$.通过公式(4)–公式(5)计算差值,可以得到:如果 $Diff \geq 0$,则 $L_{t_0} > L_{t_1}$,从而表明交换可以达到降低 stretch 的作用.值得注意的是,以上是一个近似的分析.事实上,当节点的位置发生变化时,对应的访问次数也可能随之发生变化.这就是为什么不是每一次交换都会提高整体网络的效率的原因.在实验部分也会看到这一点.

2 可控的开销

在第 1.1 节提到了 3 点要求,本节主要针对最后一点,即开销的控制.可以将节点交换带来的额外开销分为 4 个方面:(1) 探测邻居节点;(2) 探测要交换的节点;(3) 交换路由表以及其他相关信息;(4) 交换节点上的数据.我们认为,探测邻居节点的开销是十分有限的,并且可以在构造 P2P 覆盖网络时“捎带”实现.所以,下面给出减少其他 3 方面开销的方法.

2.1 自适应的探测

开销(2)和开销(3)主要与节点的交换次数相关.在基本方法中,每个节点都是每隔一定的时间做周期性的探测.事实上,当整个覆盖网络已经优化并且平均时延趋于稳定时,周期性的探测就变得耗时且意义不大.在理想情况下,当整个网络的平均时延变化不大时,就可以延长探测周期甚至停止探测.由于分布式系统的限制,通常只能基于局部信息作出上述判断.这里给出一种自适应的探测策略来降低探测和节点交换的次数.

从局部的角度看,每个节点都出自它的邻居节点组成的“环境”中.如果一个节点的邻居更换频繁,那么它所处的环境是“不稳定”的.因此要进行节点的探测和交换;相反地,如果它的邻居节点在一定时间段内没有发生变化,则可以认为它所处的环境相对“稳定”.这里使用参数 *activity* 来描述节点的状态,并作为判断是否发起周期探测的依据.开始时,参数 *activity* 被设置为一个初值 *initial value*.如果一个节点发生了交换,则表明它进入了一个新的环境,其 *activity* 就会增加,以保证探测继续进行.此外,它还要通知新的邻居节点增加各自的 *activity* 值.而每经过一个周期,*activity* 就要减少.详细的伪代码如算法 1 所示.

算法 **1**. 自适应探测算法.

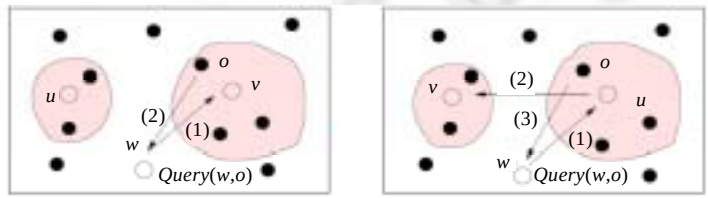
```
activity=initial value
while activity≥threshold do
    probe one node using TTL
    if swap is necessary then
        exchange the node ID and routing tables
        activity=activity+1
        notify neighbors to increase activity
    end if
    activity=activity−1
    wait for fixed interval
end while
```

其中的两个参数 *initial value* 和 *threshold* 对探测数都有影响.恰当的参数值应当根据具体系统作具体设置.在实验中,两者的初值都设为 0.结果表明,在没有过多牺牲方法有效性的前提下,交换数目可以得到明显降低.

2.2 数据项的移动

在一个实际的 P2P 系统中,每个节点负责键值空间的一个部分.当节点 *u* 和 *v* 交换标号后,它们拥有的数据项也应当进行交换.这个过程一般会带来很大的额外开销.我们受到 Baumann 等人在移动 Agent 领域工作[13]的

启发,提出一种“阴影策略”来降低数据移动带来的开销.图 3 说明了该策略.初始时,节点 u 和 v 都有一些数据项(在图 3(a)中用阴影区域表示).数据项 o 在节点 v 上.一个查询(w,o)从节点 w 发起,要查询数据项 o.查询过程如图 3(a)所示.经过若干跳,查询必然被传到节点 v 上,由节点 v 向节点 w 返回数据项 o.在节点 u 和 v 交换标号之后,逻辑上它们所负责的数据项区域也发生了交换.但这里我们不是立即交换它们的数据项,因此数据项 o 仍然在节点 v 上.两节点都记录对方节点的“阴影”,其中记录了对方节点的地址信息和该阴影的生存期.在生存期变为 0 之前,所有关于尚未交换的数据项的查询都被两节点重新定位到正确的位置.例如,在图 3(b)中,查询(w,o)被节点 u 重新定位到节点 v.仍旧由节点 v 向节点 w 返回数据项 o.当节点趋于稳定,生存期变为 0 时,数据项才会被交换.生存期的值与节点的状态有关,这在第 2.1 节中已经进行了讨论.考虑到在一段时间内,对于每个节点可能会参与多次节点交换,相应的阴影状态会发生不一致的情况,一种解决的方法就是引入简单的同步机制,即在发生交换的两个节点完成数据交换之前(在阴影没有失效之前),不允许它们与其他节点进行交换.很明显,这样的方法使得一次查询的反应时间加长了.因此,应当根据具体应用需要,在数据移动和查询开销之间作出权衡.也正是因为数据的大小、分布等因素都与具体应用相关,在这里只提出一个解决策略,而不在实验中考虑.



(a) Before swap (b) After swap

(a) 交换前 (b) 交换后

● Data item ○ Node

Fig.3 Shadow scheme

图 3 阴影策略

3 模拟实验

3.1 模拟方法

实验使用 GT-ITM(Georgia tech internetwork topology models)拓扑生成器[14]生成 transit-stub 模式的物理网络.事实上,一共生成两种不同类型的网络拓扑:一种称为 ts-large 类型,它包含 70 个 transit 区域,每个 transit 区域有 5 个 transit 节点,每个 transit 节点上附有 3 个 stub 区域,每个 stub 区域又有 2 个 stub 节点;第 2 种称为 ts-small 类型,它仅包含 11 个 transit 区域,但是每个 stub 区域有 15 个 stub 节点.从直观上看,ts-large 类型拓扑比 ts-small 具有更宽的主干和更稀疏的边界网络.除了有关物理拓扑的实验以外,我们总是以 ts-large 作为基础,因为它更近似于 Internet 中节点分散,每个边界网络包含少量节点的情况.同时,stub-stub,stub-transit 和 transit-transit 这 3 种连接分别赋予 5ms,20ms 和 100ms 的时延.在物理网络建成之后,从中选择一定数量的节点作为覆盖网络中的节点,选取的节点数为 n={300,600,1200}.实验选择 Chord 作为模拟的基础平台.这是因为 Chord 本身的一些特点决定了有些方法并不适合它,这在本文的开始部分已经提到.表 2 列出了所有实验使用的参数和默认值.

Table 2 Parameters selection in the experiment

表 2 实验参数的选择

Parameter	Meaning	Default value (type)
ts	The transit-stub model	ts-large
n	The number of nodes in Chord	600
TTL	Time-to-Live for probing	2
T	The time interval for node swapping	1 min.
δ	The percentage of nodes join and leave simultaneously	0
t	The time interval that δ% nodes join and leave	1 min.
activity threshold	One parameter of adaptive probing	0
threshold	One parameter of adaptive probing	0
b	One parameter for PNS version Chord	2

3.2 交换的有效性

实验使用 stretch 作为衡量覆盖网络和物理网络匹配程度的标准.它随着时间发生变化,时间间隔 T 被设置为 1 分钟.图 4 显示了 TTL 的变化对 stretch 的影响.此时,节点规模为 600.共有 4 种不同的情况:在集中式的情况下,节点可以选择任意其他节点作为交换的对象;而在分布式的情况下,分别设置 TTL={1,2,4}.TTL=1 表示仅探测邻居节点;TTL=2 表示探测邻居的邻居;TTL=4 表示探测距离自己约为网络半径长度的节点[**].从图中可以看到,仅探测邻居节点不能有效降低 stretch,而其他 3 种方法效果相近.其原因在于邻居之间的信息十分有限,不能反映这个覆盖网络的情况.考虑到集中式的方法很难在分布环境中使用,可以说只有当 TTL≥2 时的探测可以在 P2P 系统中达到较为理想的效果.再进一步考虑到探测的开销,TTL=2 是一个比较好的选择,它也被用于下面的一系列实验中.此外,从图 4 中可以发现,stretch 并不总是减小的.这与我们在第 1.2 节中的分析是一致的.

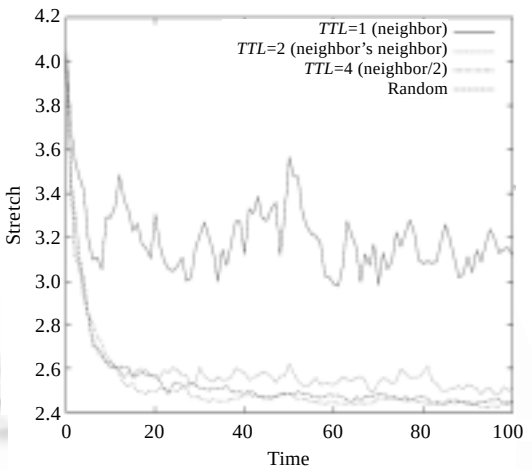


Fig.4 Varying TTL

图 4 改变 TTL

图 5 显示了节点规模的影响.当节点规模变大时,交换的优化效果减弱.这种现象可以从两个方面来解释:首先,因为 TTL 被限制为 2,而网络规模变大,探测的信息相对变得更加局部化;其次,覆盖网络规模变得越大,它就越接近之前构造的物理网络,因而优化的效果变得不太明显.物理拓扑结构的影响如图 6 所示.两种物理拓扑的结构,都由大约 2 200 个节点组成.很明显,在 ts-large 上有更好的优化结果.这是由于在 ts-large 的拓扑结构中,只有少数的 stub 节点在一个 stub 区域中.也就是说,两个相距较远的 stub 节点作交换的几率相对较高.而这样的交换能够比较明显地提高系统的整体性能.正如前面提到的,ts-large 类型的物理网络和 Internet 更为接近,因此可以说,我们的方法在大规模网络中会有较好的性能.

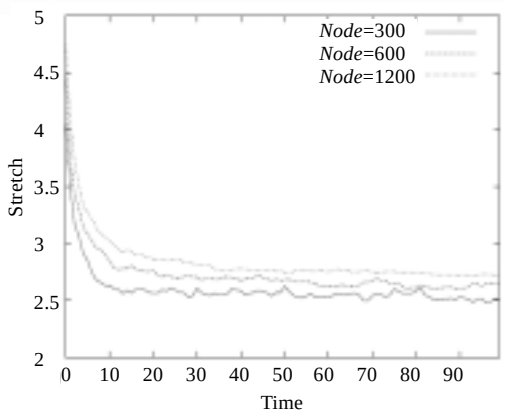


Fig.5 Varying the system scale

图 5 改变系统规模

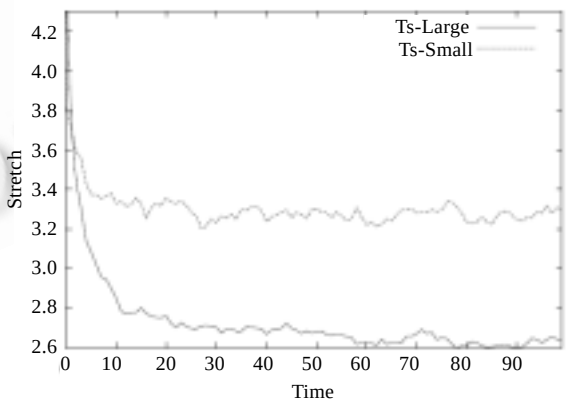


Fig.6 The impact of physical network

图 6 物理拓扑的影响

3.3 与其他方法联合

正因为基于节点标号交换的方法是协议无关的,因此它可以很容易地在其他同类型方法的基础上进一步

** 因为节点规模是 n=600,Chord 的网络直径大约是 $d=\log_2 n\approx 8$.

提高性能.ChordFingerPNS[15]是一种拓扑有关的 Chord.系统中节点的每个 Finger 有$(b-1)\log_b(n)$个可供选择的项.每个节点在它的直接后继节点 finger 列表中通过邻近邻居选择(PNS)的方法构造路由表.简单地说,它通过更改 Chord 协议,加入了邻近邻居选择(PNS)的机制.由于候选邻居节点是有限的,所以,这样的方法不能达到理想的匹配效果.我们比较了 3 种不同的情况:原始的 Chord,ChordFingerPNS 和加入了节点交换机制的 ChordFingerPNS.从图 7 中可以看到,节点交换机制可以进一步将性能提高 20%左右.虽然这里只给出了邻近邻居选择的例子,但实际上,我们的方法可以与其他任何使拓扑敏感的方法联合起来使用.

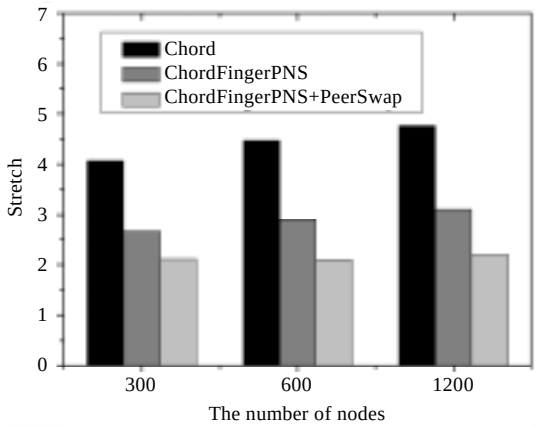


Fig.7 Combined with ChordFingerPNS

图 7 和 ChordFingerPNS 的联合使用

3.4 动态环境

动态性是 P2P 系统中的一个十分重要的特性.尽管目前有不少针对 P2P 系统动态性的研究[16],但还没有一个标准的模型来刻画节点动态的行为.在模拟实验中,我们简单地假定在一个固定的时间间隔 t 内有δ%的节点加入,同时有δ%的节点离开.设置 t=1min.,δ={0,1,5}.图 8 显示了实验结果.从图 8 中可以看到:当δ=5 时,stretch 的波动比较大.但是,节点的加入/离开并不总是使覆盖网络的匹配程度降低,因为节点的加入和离开有可能起到和节点交换类似的效果.还应注意到的是,波动的程度应该与两个参数 T 和 t 的相对大小有关.如果节点交换比节点加入/离开要频繁,波动自然就会减小.总的来说,尽管有波动,但是节点交换的方法在动态环境中依然十分有效.

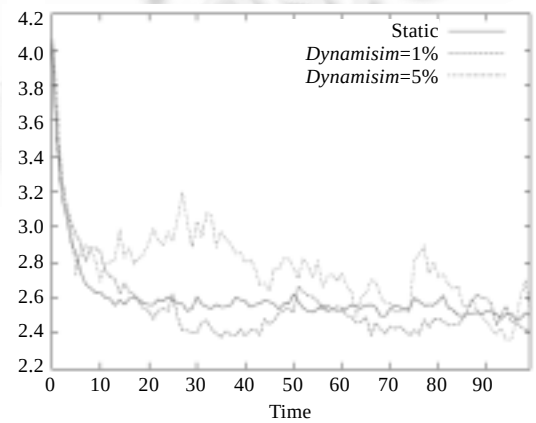


Fig.8 The stretch in dynamic environment

图 8 动态环境中的 Stretch

3.5 自适应的探测

不考虑数据项的移动,最大的开销与探测以及交换的次数密切相关.图 9 表明了自适应探测方法的有效性.固定的方法(fixed method)是每隔一段固定的时间进行一次探测,而自适应的方法以 *activity* 为标准判断是否进行探测.两个参数 *initial value* 和 *threshold* 均设为 0.图中横坐标代表 stretch,纵坐标代表交换次数.每个点代表在 1 分钟的时间间隔内,整个系统的交换次数和交换完成后对应的 stretch 值.从图中可以看出,自适应的探测方法明显降低了交换次数.当然,stretch 降低的效果也有所减弱.这一点可以从点的分布看出.stretch 较大的点,固定的方法比自适应的方法少一些.然而,相比较而言,交换次数的降低更为明显.因此,在将 stretch 和交换次数做出权衡时,应当选择自适应的方法.

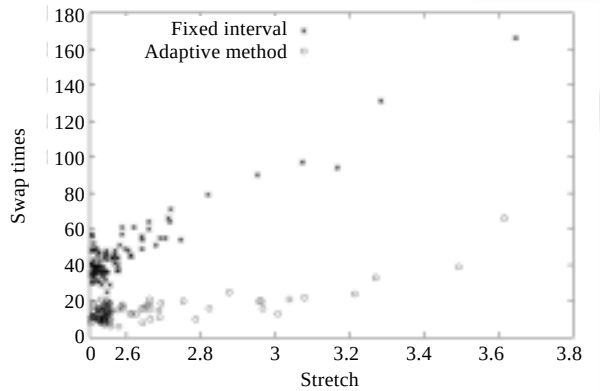


Fig.9 The tradeoff between swap times and stretch

图 9 Stretch 和交换次数的权衡

4 相关工作

大多数关于使得覆盖网络拓扑有关的工作已经在本文的开始部分作了介绍.在众多方法中,与我们的工作最为相关的是一种称为“SAT-match”的方法[17].它的基本操作是“跳跃”,即节点通过泛洪机制发现与其相近的节点,而后“跳”到该节点附近.本质上说,它也是一种变换节点标号的方法.然而,这种方法存在一些缺陷:首先,“跳跃”任意改变了节点标号,无法保证结构不变,也具有安全的隐患;其次,文中没有提到关于控制开销,特别是数据项移动的问题.另外,值得一提的是,Liu 等人提出的使得覆盖网络拓扑有关的方法[18],这种方法通过有限的局部信息来自适应地调整覆盖网络的连接.但是,这种方法应用在无结构的 P2P 网络中,对有结构的 P2P 系统并不合适.

5 总 结

本文提出一种方法来解决覆盖网络和物理网络失配的问题.这种方法是协议无关的,适用于所有基于 DHT 的有结构的 P2P 系统.此外,本文还提出了自适应探测和阴影策略来降低节点交换带来的开销.理论分析和实验都表明,节点交换是一种十分有效的方法.模拟实验还表明,它可以在其他方法的基础上进一步提高系统性能.在不牺牲方法有效性的前提下,自适应探测可以大量减少节点交换的数目,进而大幅度降低额外开销.

当然,本文所提出的方法还存在一些问题.比如,对于节点之间内容的交换,还需要进一步面向应用的实验来验证.此外,我们还考虑将该方法应用于非结构的对等网络.以上问题留待在今后的工作加以解决.

References:

[1] Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content-addressable network. In: Govindan R, ed. Proc. of the ACM SIGCOMM. New York: ACM Press, 2001. 161−172.

[2] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup protocol for Internet applications. In: Govindan R, ed. Proc. of the ACM SIGCOMM. New York: ACM Press, 2001. 149−160.

[3] Rowstron A, Druschel P. Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In: Guerraoui R, ed. Proc. of the 18th IFIP/ACM Int'l Conf. on Distributed Systems Platforms (Middleware 2001). Heidelberg: Springer-Verlag, 2001. 329−350.

[4] Zhao BY, Huang L, Stribling J, Rhea SC, Joseph AD, Kubiatowicz J. Tapestry: A resilient global-scale overlay for service deployment. IEEE Journal on Selected Areas in Communications, 2004,22(1):41−53.

[5] Malkhi D, Maor M, Ratajczak D. Viceroy: A scalable and dynamic emulation of butterfly. In: Ricciardi A, ed. Proc. of the 21st Annual Symp. on Principles of Distributed Computing. New York: ACM Press, 2002. 182−192.

[6] Shen HY, Xu CZ, Ghen G. Cycloid: A constant-degree and lookup-efficient P2P overlay network. In: Panda DK, Duato J, Stunkel C, eds. Proc. of the 18th Int'l Parallel and Distributed Processing Symp. (IPDPS 2004). New York: IEEE Press, 2004. 26−30.

[7] Xu Z, Tang C, Zhang Z. Building topology-aware overlays using global soft-state. In: Panda DK, Duato J, Stunkel C, eds. Proc. of the 23rd Int'l Conf. on Distributed Computing Systems (ICDCS 2003). New York: IEEE Press, 2003. 500−508.

[8] Ratnasamy S, Handley M, Karp R, Shenker S. Topologically-Aware overlay construction and server selection. In: Proc. of the IEEE INFOCOM. New York: IEEE Press, 2002. 1190−1199.

[9] Winter R, Zahn T, Schiller J. Random land-marking in mobile, topology-aware peer-to-peer networks. In: Proc. of the 10th IEEE Int'l Workshop on Future Trends of Distributed Computing Systems (FTDCS 2004). New York: IEEE Press, 2004. 319−324.

[10] Ratnasamy S, Shenker S, Stoica I. Routing algorithms for DHTs: some open questions. In: Druschel P, ed. Proc. of the 1st Int'l Workshop on P2P Systems (IPTPS 2002). Berlin: Springer-Verlag, 2002. 45−52.

[11] Castro M, Hu Y, Rowstron A. Exploiting network proximity in distributed hash tables. In: Proc. of the Int'l Workshop on Future Directions in Distributed Computing (FuDiCo 2002). Berlin: Springer-Verlag, 2002. 52−55.

[12] Waldvogel M, Rinaldi R. Efficient topology-aware overlay network. ACM Communications Review, 2003,33(1):101−106.

[13] Baumann J, Hohl F, Rothermel K, StraBer M. Mole—Concepts of a mobile agent systems. World Wide Web, 1998,1(3):123−137.

[14] Zegura EW, Calvert KL, Bhattacharjee S. How to model an Internetwork. In: Proc. of the IEEE INFOCOM'96. New York: IEEE Press, 1996. 594−602.

[15] Dabek F, Li J, Sit E, Robertson J, Kaashoek MF, Morris R. Designing a DHT for low latency and high throughput. In: Proc. of the 1st USENIX Symp. on Networked Systems Design and Implementation (NSDI 2004). San Francisco: USENI Press, 2004. 85−98.

[16] Ge Z, Figueiredo R, Jaisal S, Kurose J, Towsley D. Modeling peer-to-peer files sharing systems. In: Proc. of the IEEE INFOCOM 2003. New York: IEEE Press, 2003. 2188−2198.

[17] Ren S, Guo L, Jiang S, Zhang X. SAT-Match: A self-adaptive topology matching method to achieve low lookup latency in structured P2P overlay networks. In: Proc. of the 18th Int'l Parallel and Distributed Processing Symp. (IPDPS 2004). New York: IEEE Press, 2004. 83−91.

[18] Liu Y, Liu X, Xiao L, Li L, Zhang X. Location awareness in unstructured P2P systems. IEEE Trans. on Parallel and Distributed Systems, 2005,16(2):163−174.

邱彤庆(1981－),男,江苏泰州人,硕士生,主要研究领域为对等计算.

陈贵海(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为对等计算,网络系统,传感器网络,并行计算.