基于 P2P 计算模式的自组织网络路由模型*

李祖鹏[1,2+], 黄建华[2], 唐 辉[2]

[1](空军工程大学 电讯工程学院,陕西 西安 710077)

[2](国家数字交换系统工程技术研究中心,河南 郑州 450002)

A P2P Computing Based Self-Organizing Network Routing Model

LI Zu-Peng[1,2+], HUANG Jian-Hua[2], TANG Hui[2]

[1](Telecommunication Engineering Institute, Airforce Engineering University, Xi'an 710077, China)

[2](National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China)

+ Corresponding author: Phn: +86-27-84200342, E-mail: Lizp@mail.ndsc.com.cn, http://www.ndsc.com.cn

Received 2004-03-11; Accepted 2004-09-08

Li ZP, Huang JH, Tang H. A P2P computing based self-organizing network routing model. *Journal of Software*, 2005,16(5):916−930. DOI: 10.1360/jos160916

Abstract: Building a virtual network topology named P2P overlay network on top of Internet's physical topology layer based on P2P computing mode can lead to the effective building of a full-decentralized internet based self-organizing network routing model—hierarchical aggregation self-organizing network (HASN). The target and architecture of HASN are described in this paper, as well as a detailed description of the P2P decentralized naming, route discovering and updating algorithm-HASN_Scale. Simulation results testify the performance of HASN.

Key words: peer-to-peer network; self-organizing network; structured overlay network; hierarchical routing table

摘 要: 通过使用 peer-to-peer(P2P)计算模式在 Internet 物理拓扑基础上建立一个称为 P2P 覆盖网络(P2P overlay network)的虚拟拓扑结构,有效地建立起一个基于 Internet 的完全分布式自组织网络路由模型——分级集中式自组织网络路由模型(hierarchical aggregation self-organizing network,简称 HASN).分别描述了 HASN 路由模型的构建目标和体系结构,并详细分析了 HASN 采用的基于 P2P 计算模式的分布式命名、路由发现和更新算法 HASN_Scale,并在仿真实验的基础上,对 HASN 路由模型的性能进行了验证.

关键词: 对等网络;自组织网络;结构化覆盖层网络;分级路由表

中图法分类号: TP393 **文献标识码**: A

自组织网络技术(self-organizing network)[1,2]和计算机对等互联网(peer-to-peer network,简称 P2P)技术[3,4]的结合是本文主要的研究方向.事实上,确自组织网络本身就具有对等架构特性,网络中每个用户节点都兼备了独立路由和主机功能,不存在一个网络中心控制点,用户节点之间的地位是平等的,其路由协议通常采用分布式

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA111141 (国家高技术研究发展计划(863))

作者简介: **李祖鹏**(1976−),男,博士生,讲师,主要研究领域为自组织网络和分布式计算系统;**黄建华**(1961−),男,教授,博士生导师,主要研究领域为计算机网络技术,分布式计算系统;**唐辉**(1979−),男,硕士生,主要研究领域为分布式计算系统.

控制方式,因而具有很强的鲁棒性和抗毁性.而自组织网络的这些特性恰恰就是 P2P 网络技术的核心所在.在对 P2P 技术的研究中发现,通过应用 P2P 计算模式,在 Internet 物理拓扑基础上,建立一个称为 P2P 覆盖层网络的虚拟拓扑结构[5],我们可以有效地建立起一个基于 Internet 的自组织网络模型.

目前,基于 P2P 计算模式的 Napster[6],Freenet[7]和 Gnutella[8]等网络模型已被广泛地应用.然而,中央集权式(centralized)的 Napster 利用中央服务器负责目录管理的服务会因为受服务器的限制,存在服务质量无法提高和单点崩溃(single point of failure)的问题;而非中央集权式(decentralized)的作法,如 Gnutella 及 Freenet,由于没有中央服务器,在搜寻数据时是以 flooding 的方式将消息散布在网络上,存在着消息泛滥的问题,也使得系统的可扩展性(scalability)无法提升.因此,结构化覆盖网(structured overlay network,简称 SON)路由协议,如 CAN[9],Chord[10],Pastry[11]和 Tapestry[12]所提出的算法则为改善 P2P 网络的可扩展性而被提了出来[13].这些模型的共同点就是利用杂凑(hashing)的方式,将数据和节点运算成一个键值(key),利用键值来完成数据的放置与维护.但由于这些算法并没有考虑网络实际拓扑结构,因而即使是邻近的两个节点仍有可能因为杂凑的结果,而必须经过很长的搜寻路径才能取得数据,严重降低了路由的效率[14].

通过将基于 SON 的 P2P 路由技术和自组织网络路由技术研究有效地结合,本文提出分级集中式自组织网络路由模型(hierarchical aggregation self-organizing network,简称 HASN).HASN 系统模型是一个基于 Internet 拓扑架构和 P2P 计算模式的自组织网络路由模型,构建该模型的目的是为了在 HASN 基础上搭建一个应用于广域网络(WAN)的高性能、高可用、负载均衡、动态的自组织网络平台.该平台位于应用层,通过在 Internet 物理拓扑基础上建立一层基于 P2P 覆盖网络的虚拟拓扑结构,并在其上使用基于 P2P 计算模式的路由协议,从而有效地建立起一个具有完全分布式结构的自组织网络路由模型.

1 分级集中式自组织网络路由模型(HASN)

1.1 HASN模型层次结构

本系统的设计目标是针对下一代网络技术(NGN)的发展对分布式、动态大规模自组织网络应用的需求,建立一个基于 P2P 计算模式的自组织网络路由模型,图 1 列出了该系统模型的层次结构,该模型可以用于提供文件资源定位服务、信息管理服务、文件资源传输服务以及通信安全服务等基本的信息服务功能.

构建 HASN 模型的目的,主要是为了通过引入基于网络实际拓扑的网络分层和组群划分机制,对 Internet 中所有节点进行区域性分组,并配合分布式哈希散列算法(也称为杂凑算法,distributed hash tables,简称 DHT)建立起一个具有完全分布式结构、高度可扩展性和鲁棒性的自组织网络体系架构.

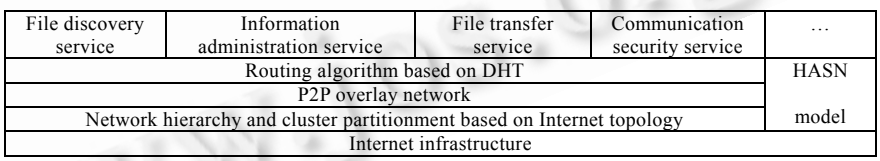


Fig.1 Basic architecture of HASN

图 1 HASN 模型的基本层次结构

1.2 HASN模型体系结构

HASN 模型是一个基于两级不同架构覆盖层(overlay)的统一实体.这两级覆盖层分别为网络拓扑层(network topology layer,简称 NTL)和 DHT 网络查询层(network discovery layer,简称 NDL).其中,网络拓扑层(NTL)主要实现基于网络物理拓扑的聚集群体(cluster)划分和树型体系结构构建功能,NTL 的树型体系结构如图 2(a)所示;DHT 网络查询层(NDL)主要实现基于 DHT 算法的对等点路由和定位功能,NDL 的体系结构如图 2(b)所示.

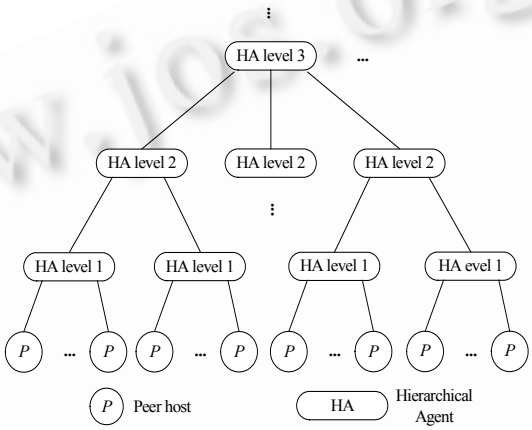
1.2.1 网络拓扑层(NTL)

HASN 模型中引入了一个重要的组件——分级代理(hierarchical Agent,简称 HA).在 HASN 模型中,分级代

理是每一个聚集群体(cluster)的核心.由于 HASN 模型采用的是完全分布式和自组织结构,网络中的任何一个节点都可能基于其物理位置的不同和加入网络时间的先后而成为不同等级的分级代理 HA,因此,在 HASN 的每一级中都有大量 HA,它们可能位于世界的不同位置.物理距离较近的对等点将被归并成组并连接至第 1 级的分级代理(HA level 1,简称 HA1),由其统一管理;物理距离较近的 HA1 将被归并成组并连接至第 2 级的分级代理(HA level 2,简称 HA2),由其统一管理,依此类推.

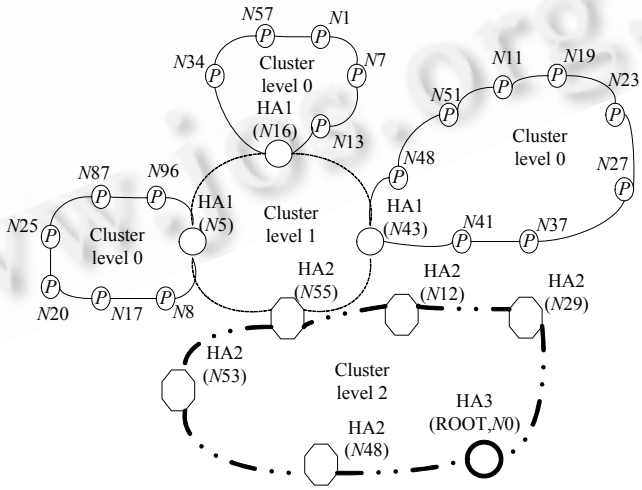
NTL 层网络初始化和组群划分过程具体描述如下:

(1) 当一个新节点 N 希望加入网络时,它首先要与网络中的一个已知成员节点联系,并获取该节点地址信息,该节点将充当新加入节点的引导节点(bootstrap node).在 HASN 中,新节点 N 通过向周围的节点广播发送网络查询消息的方法(最简单的方法可以使用 Ping 消息)发现物理距离较近的网络成员节点.其他成员节点在接收到该消息后,将返回一个应答消息 Pong,应答消息中还应包含该成员节点的身份信息,即该成员节点是普通节点或者是分级代理 HAi,其中 i=1,2,…,H(H 为 NTL 树型结构的高度).为了降低网络开销,该广播消息的初始化 TTL 值将基于网络规模设置为一个较小值,仅当无任何节点返回应答时,再将该 TTL 值逐倍增大.



(a) Architecture of HASN's topology layer

(a) HASN 模型网络拓扑层(NTL)体系结构



(b) Architecture of HASN's DHT discovery layer

(b) HASN 模型网络查询层(NDL)体系结构

Fig.2 Architecture of HASN based on different overlay

图 2 基于不同覆盖层的 HASN 体系结构

(2) N 接收到各成员节点的应答消息 Pong 后,将计算到各个成员节点的物理距离(最简单的方法是通过比较平均往返时延 RTT).假设 HASN 中每一级 HA 物理距离的门限值(threshold)分别为λ_k,其中 k=0,1,2,…,H−1(H 为 NTL 树型结构的高度).N 将基于节点的身份从返回节点列表中选取其引导节点,具体步骤如下:

(i) 如果返回节点列表中存在普通成员节点,选取返回节点列表中距离 N 最近的普通节点 M,假定 M 距离 N 的物理距离为 S_0,则 N 将比较 S_0 与λ_0 的大小关系.如果 $S_0<\lambda_0$,则 N 将加入该成员节点 M 所属 HA1 下属的聚集群体中,组群划分过程结束;否则,进入(ii).

(ii) 令 k 的起始值为 1,选取返回节点列表中距离 N 最近的第 k 级分级代理 HAi,假定 HAk 距离 N 的物理距离为 S_k,其中 k=1,2,…,H,则 N 将比较 S_i 与λ_i 的大小关系.如果 $S_k<\lambda_k$,则 N 将加入该分级代理 HAk 所属 HA(k+1)下属的聚集群体中,且该节点 N 将升级成为一个第 k 级分级代理,组群划分过程结束;否则,进入(iii).

(iii) k 值加 1 后重复(ii)过程,直到节点 N 加入某一级 HA 或者到达顶级 HA 为止(即 k=H,H 为 NTL 树型结构的高度).如果到达顶级 HA,则节点 N 将自动升级成为一个第(H−1)级分级代理.

综上所述,HASN 是一个具有完全对等式架构的自组织系统,在网络中没有预先设置任何形式的中央服务器,网络构建采取自顶向下的组网方式,以按需方式由高至低先生成各级分级代理,然后再依次加入物理距离接近的节点.在 HASN 路由模型中,HA 的主要功能在于,为新加入节点初始化路由表提供导向作用(参见后文 HASN 路由算法描述).而实际上,我们在自组织网络模型中引入 HA 的另一个目的是,在 HA 的基础上,为动态分布式网络引入安全机制、用户行为监控和评估机制,由于该研究内容非本文论述重点,因而在此不作详述,相关工作可参考文献[15].

1.2.2 网络查询层(NDL)

在网络拓扑层(NTL)结构基础上,HASN 模型将通过建立 DHT 网络查询层(NDL)结构来实现对消息的路由.由图 2(a)可见,HASN 模型在 NTL 上建立起一个个聚集群体,与之对应,我们在 NDL 中引入了聚集级别(cluster level)的概念,具体定义如下:

- 第 n 级聚集(cluster level k,简称 CL_k):对应 HASN 模型中的不同等级聚集群体.其中,每个 HA_1 下的子对等级将组成一个第 0 级聚集群体(cluster level 0,CL_0);所有的 HA_1 将组成一个第 1 级聚集群体(cluster level 1,CL_1),依此类推.
- 全聚集(full cluster,简称 FC):对应 HASN 网络中的所有节点.

根据前面所述的分级方法,假设当前 NTL 树型结构的高度为 H,NDL 查询协议描述如下(其中,NDL 层使用的路由算法 HASN_Scale 将在下一部分详细叙述):

(1) 对于 CL_i 中的一个节点,它发起的查询将首先在 CL_i 内的 Hash 环中进行,如果找到目的节点和相应的 Key 值,则返回响应消息给源节点,查询过程结束,否则进入(2).

(2) 该节点将把查询扩展到其归属 HA_i 所属的第(i+1)级聚集 $CL_{(i+1)}$内进行,如果找到目的节点和相应的 Key 值,则返回响应消息给源节点,查询过程结束,否则进入(3).

(3) 如果尚未到达全聚集 FC(即 i<(H−1)),则将 i 加 1 后返回(2),查询过程结束,否则返回查询失败.

2 HASN_Scale 路由算法

在基于 P2P 计算模式的自组织网络中,每个节点都具有客户机、服务器和路由器 3 大功能;系统中维护着大量的节点,节点可以动态地加入和离开系统,系统中任意节点之间可以进行消息通信.HASN_Scale 路由算法是 HASN 模型中运行于网络查询层 NDL 之上的路由算法.

HASN_Scale 采用了基于 DHT 的杂凑式路由算法,通过对 Chord 路由算法进行改进,为 HASN 模型提供一个具有高度可扩展性的节点命名、路由发现、路由更新和通信机制的路由算法.系统通过为每一个节点指定唯一的标识,并通过 HASN_Scale 路由表来确定本节点与其他节点之间的邻接关系,节点利用这种邻接关系向其他节点传递消息,通过多个节点的协同传递后到达目标节点,从而实现任意节点之间的通信.下面具体介绍 HASN_Scale 路由算法的基本设计:节点命名、节点的路由信息、路由发现算法以及路由更新算法.

2.1 节点的命名

HASN_Scale 采用类似于 Chord 模型中的 Consistent Hashing 算法[16,17]将网络中每个节点的标志(如 IP 地址、域名等)映射成一个长度为 M 位(bit)的二进制序列 NID,并将其唯一分配给该节点;对系统中需要存储数据的标识(文件名、ObjectId 等)进行哈希运算,将其映射成一个长度为 M 位(bit)的二进制序列 KID(key Id),其中 M 值大小需要满足 $N \le 2^M$ (假设网络节点总数为 N).整个 HASN_Scale 系统采取整体命名方式,利用各个 NID 来指定某个节点在多级 Hash 环域中的位置.当节点第 1 次加入 HASN_Scale 系统时,系统将随机地为其分配一个 NID,NID 的取值范围为 0~2^M−1.每个 NID 对应一个用 0,1 字符串来表示的二进制序列,各个节点的 NID 所对应的数值是互不相同的,对于一个总长度为 M 的 NID,它所对应的数值为

$$\text{Valueof}(NID) = \sum_{r=1}^{M} NID_i \times 2^r \tag{1}$$

HASN_Scale 中引入了全环域(global ring zone,简称 GRZ)和私有环域(individual ring zone,简称 IRZ)的概念.全环域(GRZ)是$[0, 2^M - 1)$之间的所有 0,1 字符串所对应的数值空间(M 通常为 128).系统中的每一个节点动态地将 GRZ 划分为一个个独立的私有子空间,并将该子空间作为自己的私有环域(IRZ),IRZ 可以表示为(NID_Predecessor,NID),其中 NID 是系统为该节点分配的 ID 值,NID_Predecessor 是该节点在 Hash 环上前继节点的 NID,节点的命名如图 3 所示.

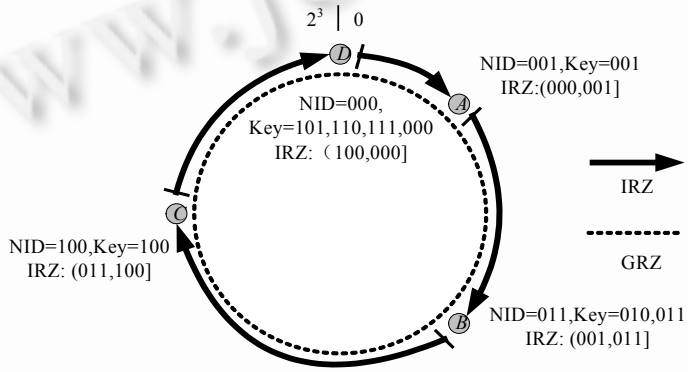


Fig.3 Nodes' naming mechanism

图 3 节点的命名

在 HASN_Scale 中引入私有环域的目的是,采取大小可以动态变化的路由表与采取固定大小的路由表相比更加灵活一些,当节点加入时所采用的随机函数所产生的随机数分布足够均匀时,节点可以根据它的 IRZ 大小来估计系统中节点的总数目,理论上其 IRZ 大小与系统中总的节点数目呈线性反比关系.另外,在查询目标节点时,查询过程直接通过比较 KID 值和目标节点 IRZ 的范围就可以有效地判断目标节点是否到达,从而简化了查询过程.

对于一个具有 N 个节点的系统,其各个节点的 IRZ 之间满足下面的关系:

$$\left.\begin{aligned} IRZ_1 \cup IRZ_2 \cup IRZ_3 \ldots \cup IRZ_N = GRZ \\ IRZ_i \cap IRZ_j = \varphi,\ i \ne j, 1 \le i, j \le N \end{aligned}\right\} \tag{2}$$

2.2 节点路由信息

HASN_Scale 路由模型中的每个节点将保存 H 个分级路由表(hierarchical routing table,简称 HRT),其中 H 是 HASN 网络拓扑层高度.假设第 i 级聚集中有 n 个节点,则每个子节点所维护的相应第 i 级 HRT 大小是 $O(\log n)$ 量级的.分级路由表(HRT)在 Chord 模型的 Finger 表结构基础上进行了改进,由于采用节点命名机制和模型基础结构的不同,HRT 路由表结构也与 Chord 大不相同.下面首先给出一个 M 位命名空间下的节点分级路由表相关概念:

- 当前聚集级数(current cluster level,简称 CCL):分级路由表下节点所属 NTL 中的聚集级数,其中 $1 \le k \le H$;

- 前继节点(predecessor):本节点在 Hash 环上沿逆时针方向遇到的第 1 个节点;
- 后继节点(successor):本节点在 Hash 环上沿顺时针方向遇到的第 1 个节点;
- 第 K 项路由环域起点(*HRTEntry*[*K*].*start*):第 K 个路由表项所覆盖环域空间起始位置,其中

$$HRTEntry[K].start = (n + 2^{k-1}) \bmod 2^M, 1 \le K \le M \tag{3}$$

- 环域空间范围(next hop ring interval,简称 NRI):路由表项 K 所覆盖的环域空间范围大小,其中

$$NRI[k] = [HRTEntry[K].start, HRTEntry[K+1].start), 1 \le K \le M \tag{4}$$

- 下一跳节点私有环域空间(next hop IRZ,简称 NIRZ):在环域空间范围 NRI 内沿顺时针方向第 1 个活动节点 G 的私有环域空间,即 G 满足

$$HRTEntry[K].start \in IRZ(G), 1 \le K \le M \tag{5}$$

分级路由表(HRT)的基本内容有:当前拓扑层聚集级数(CCL),第 K 项路由环域起点(*HRTEntry*[*K*].*start*)、下一跳节点私有环域空间(NIRZ),其中,K 为路由表序号.路由表的结构如下:

当前拓扑层聚集级数 CCL	第 K 项路由环域起点(*HRTEntry*[*K*].*start*)	下一跳节点私有环域空间(NIRZ)
⟨CCL *i*⟩	⟨HRTEntry *X*⟩	⟨NIRZ *Y*⟩

表 1 和表 2 给出了图 1 中节点 B 的分级路由表,假设该网络具有两级聚集结构,即 H=1(在此假设 CL_0 层包括 A,B 和 D 节点,而 CL_1 级则为全聚集 FC,包括 A,B,C 和 D).

Table 1 Node B's HRT on CL_0

表 1 节点 B 在 CL_0 级路由表(HRT)结构

K	CCL	*HRTEntry*[*K*].*start*	NIRZ
1	0	100	*D*=(011,000)
2	0	101	*D*=(011,000)
3	0	111	*D*=(011,000)

Table 2 Node B's HRT on CL_1

表 2 节点 B 在 CL_1 级路由表(HRT)结构

K	CCL	*HRTEntry*[*K*].*start*	NIRZ
1	1	100	*C*=(011,100]
2	1	101	*D*=(100,000]
3	1	111	*D*=(100,000)

2.3 HASN_Scale路由发现算法

根据 HASN_Scale 路由模型的节点命名机制,网络中所有节点通过其私有环域空间 IRZ 的连接构成全环域 GRZ.采用这种简单的节点邻接关系,每个节点仅需保存其后继节点信息,节点之间的通信可以通过向后继节点进行消息前递的方式来实现.这种消息前递方法虽然简单,但效率却非常低,因为网络中的两个节点为了找到对方需要经过 $O(N)$跳(N 为网络中节点个数),最坏情况下,源节点可能需要将消息绕环传递一圈才能到达目的节点.正是基于这个原因,为了加速查询过程和提高查询效率,HASN_Scale 路由模型为每个节点引入了分级路由表 HRT,并在此基础上提出了 HASN_Scale 路由发现协议.HASN_Scale 路由发现协议也是在 Chord 路由发现协议上的一个改进,其与 Chord 的根本区别在于,HASN_Scale 路由发现协议中引入了多层拓扑层聚集级数(cluster level)的概念.HASN_Scale 路由发现算法见表 3.

我们将 HASN_Scale 路由发现算法描述如下:

(1) 当某节点 A 需要发起一次路由查询(或接收到经转发的路由消息)时,它将首先检查(或提取消息中的)查询键值 D,并将键值 D 与本节点的私有环域空间(IRZ)相比较,如果 $D \in$IRZ,则路由发现过程终止,A 构造并返回应答消息;如果 $D \in$IRZ,则进入(2).

(2) 节点 A 将按照分级路由表(HRT)聚集级数大小,逐级查询键值 D,在每一级 HRT 中,A 将查询键值 D 依次与分级路由表项进行比较,寻找下一跳节点私有环域空间(NIRZ)包含 D 的路由表项,并将分组转发至相应下一跳节点,消息的每次传递都将在路由表项中选择 NIRZ 与 D 值最接近的路由表项所对应的下一跳节点,并将消息发送给该节点,该节点接收到消息后同样根据上述规则继续进行前递;如果在当前级别 HRT 的路由查询失

败,则进入高一级的 HRT,依此类推.

(3) 如果路由发现过程已到达最高级 HRT(即 CCL=H−1),并且仍然未查询到包含查询键值 D 的目标节点,则路由发现过程结束,并返回查询失败消息.

综上所述,网络中每个节点将根据其组群划分情况构建多个分级路由表,第 i 个分级路由表对应第 i 级聚集群体 CL_i.在路由发现过程中,节点将按级别由低至高的顺序依次在各个聚集中进行搜索,直至全聚集 FC(FC 包括网络中的所有节点).如果在 FC 层仍未找到目的节点,则返回查询失败.

Table 3 Route discovering algorithm of HASN_Scale

表 3 HASN_Scale 路由发现算法

```
h:current network topology layer (NTL) cluster level,1≤h≤H;
HRTEntry[i]:the i_th entry in current hierarchical routing table (HRT),1≤i≤M;
D:the value of Key in query message;

if  (D∈valueof(node)){
    //search in local space
          return node_id;}
      else {
          //search in HRT
          while (h<H){
        for k=1 upto M{
            if D∈[HRTEntry[i].start,HRTEntry[i+1].start){
                    forward to NIRZ[i];
                    wait_for_reply;
                    if (timeout){
                        h=h+1;}
                     else{
                        break;}          }}}
    }
```

2.4 节点的动态加入和退出

作为支持自组织网络特性的路由算法,HASN_Scale 必须能够支持节点随时动态进入和退出系统.HASN_Scale 算法采用了上文所述的节点命名机制为系统中每个节点分配一个唯一标识 NID,并划分其私有环域空间(IRZ).当一个新节点要加入系统时,系统将通过分割 Hash 环上某个成员节点的 IRZ 来为该节点分配一个新的 IRZ;当一个节点要退出系统时,系统将把该节点的 IRZ 与系统某个成员节点合并来回收该环域空间.

一个节点加入系统的过程描述如下:

(1) 新节点 M 首先按照上节介绍的 NTL 组网协议选择其归属 HA;

(2) M 选择系统中的一个成员节点 H 作为其引导节点(bootstrap node),在 HASN_Scale 中,默认情况下,M 将选取其归属 HA 作为引导节点.然后,节点 M 将通过引导节点 H 查找到其在 Hash 环上的后继节点 S(S=Successor(M)),该后继节点 S 则根据 M 的 NID 大小将自己的 IRZ((NID(S.Predecessor),NID(S)])划分为两个部分,自己保留(NID(M),NID(S)]所在的那一部分,并将另一部分(NID(S.Predecessor),NID(M)]作为新加入节点 M 的 IRZ.

(3) M 通过其后继节点 S 获得其前继节点 P(P=Predecessor(S))的信息.然后 M 分别向 P 和 S 发送一个加入请求消息,S 和 P 接收到消息后,将分别更新自己的前继和后继节点,其中,S 将自己的前继更新为 M,而 P 则把自己的后继更新为 M.至此,Hash 环的完整性得到了保证.

(4) M 通过其后继节点 S 依次在 Hash 环中查找自己的每个分级路由表表项(HRTEntry)对应的下一跳节点,实现对分级路由表(HRT)的初始化.

(5) M 的后继节点 S 将根据重新划分的 IRZ 将应属于新节点的键值 Key 转移到新节点 M.

(6) HASN_Scale 启动路由更新(route update)过程(后文将详细描述),更新系统中所有其他节点的路由表(HRT),使其能够适应网络结构的变化,并使整个网络重新收敛.

为了支持 HASN 模型的自组织特性,HASN_Scale 系统还必须支持节点的动态离开.一个节点正常退出 HASN_Scale 的具体过程描述如下:

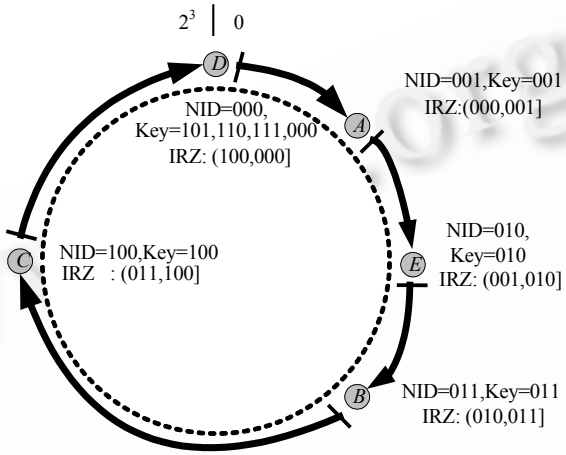
(1) 节点 *Q* 首先向其后继节点 *S* 和前继节点 *P* 发送一个离开请求消息,*S* 和 *P* 接收到消息后,将分别更新自己的前继和后继节点,其中,*S* 将自己的前继更新为 *Q* 的前继(节点 *P*),而 *P* 则把自己的后继更新为 *Q* 的后继(节点 *S*).至此,Hash 环的完整性得到了保证.

(2) 节点 *S* 将更新自己的 IRZ,即对节点 *Q* 的 IRZ 进行合并,合并前,*Q* 的 IRZ 为(NID(*P*),NID(*Q*)],*S* 的 IRZ 为(NID(*Q*),NID(*S*)),合并完毕后,*S* 的 IRZ 将变为(NID(*P*),NID(*S*)).

(3) 节点 *S* 将根据重新划分的 IRZ 将属于节点 *Q* 的键值 Key 全部转移到 *S*.

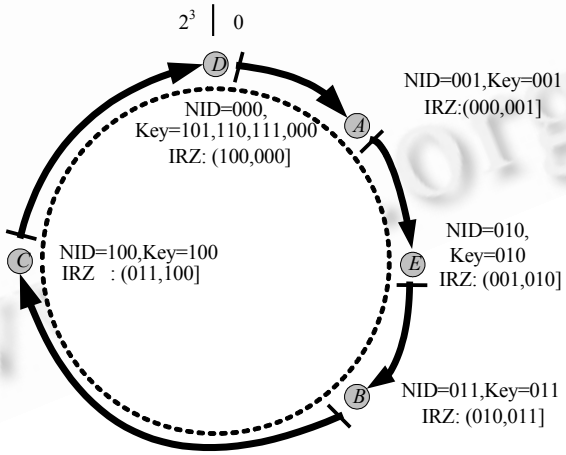
(4) HASN_Scale 启动路由更新过程,更新系统中所有其他节点的分级路由表(HRT),使其能够适应网络结构的变化,并使整个网络重新收敛.

图 4(a)和图 4(b)分别显示了图 1 中 Hash 环在节点 *E* 加入系统后和节点 *B* 退出系统后的状态.



(a) System status after *E* joins

(a) 节点 *E* 加入系统后的状态



(b) System status after *B* leaves

(b) 节点 *B* 离开系统后的状态

Fig.4 Nodes' join and departure

图 4 节点的加入和退出

2.5 路由更新机制(route update)

对于基于自组织特性的 HASN_Scale 而言,路由更新机制是路由算法启动的一种自动恢复机制,其主要功

能是在网络中由于某些节点加入或退出等原因导致网络拓扑发生变化时,更新系统中所有关联节点的分级路由表(HRT),使其能够适应网络结构的变化,并使整个网络迅速重新收敛.目前通常采用的是事件触发式、周期式以及两者相结合的路由更新方式.对于大规模网络而言,周期性路由更新机制需要花费较高代价(如带宽、电源、CPU 处理能力等),以使得路由表能跟得上当前网络拓扑结构的变化,但动态变化的拓扑结构又可能使高代价获得的路由表内容变成无效信息,尤其是在网络拓扑高速变化时,整个系统将始终处于不收敛状态.虽然这种方式直接保证了维护开销正比于路由表大小,但是只有在路由表不大时才可行,一旦路由表达到一定规模,其开销将无法承受(例如,对于含 10 000 个表项的路由表,若每 50s 探测一次,则每秒需发送 200 个消息).因此,考虑到自组织网络的高度动态特性,HASN_Scale 采用事件触发式路由更新机制.

就路由更新触发源而言,在节点加入和节点正常退出系统(如用户正常关闭 P2P 程序)时,该节点会通过应用层程序向网络中相关节点发送退出消息,该退出消息就是触发源;而当节点在没有任何先兆情况下异常退出(如节点掉线或死机等)时,系统一般通过"发现-广播"机制告知其他结点,采用和 one-hop overlay[18]类似的方法,这时,该广播消息就充当触发源.针对节点无先兆的异常退出,HASN_Scale 规定系统中的每个节点必须定期向其前继节点发送 KeepAlive 消息,以探测该节点是否存活,一旦发现前继节点异常退出,则由该异常退出节点的后继节点启动路由更新过程.

新节点加入系统的路由更新过程描述如下:

(1) 路由更新过程由事件触发,由新加入节点 M 发起,更新过程将采取回溯(retrieval)的方式,并按照分级路由表表项(HRTEntry)的次序 i 依次发现和更新关联节点,回溯过程沿逆时针方向进行,每次回溯的间隔为 2^{i-1},i=1,2,...,M.

(2) 对于新节点加入触发的路由更新过程,回溯到第 i 个节点时,将查找该节点的第 i 个路由表项对应的下一跳节点 NID 值,如果当前 NID 值大于新加入节点的 NID,则用 NID(M)替换当前表项值,进入 C,否则继续回溯第 i+1 跳节点.

(3) 回溯当前更新节点的前继节点,并查找该节点的第 i 个表项对应的下一跳节点 NID 值,如果当前 NID 值大于新加入节点的 NID,则用 NID(M)替换当前表项值,继续回溯上一个前继节点,否则回溯第 i+1 跳节点.

(4) 当 i=M 时,路由更新过程结束.

节点退出系统的路由更新过程与节点加入系统的路由更新过程相似,有两点不同:

- 节点退出系统的路由更新过程是由退出节点 Q 或退出节点的后继节点 S(S=Successor(Q))发起.
- 节点退出触发的路由更新过程也采用回溯的方法更新其他节点的路由表,但在回溯过程中,仅查找回溯节点路由表项对应的下一跳 NID 值是否为退出节点的 NID,即 NID(Q),若是,则用退出节点的后继替换当前表项值.

3 HASN 路由模型优化策略

3.1 数据项备份机制(data item replication)

HASN 模型提出组群划分和层次化结构的目的是为了充分考虑节点网络邻近性特征(network proximity character),希望通过组群划分的方法将物理距离较近的节点归为一组,以确保网络中邻近节点间的路由过程都在组群内部完成,从而避免了 Chord 等网络存在的绕路(detouring)问题,降低了系统路由时间开销,并减少了发送消息数量.

但是,由于 HASN 中节点采用的是整体命名方式,因此可能会导致在各个子聚集中查询命中率较低,而必须将查询扩展到更高一级的聚集中.假设存在一个 3 级分级网络 W,CL_1 是某大学校园局域网(LAN),CL_2 是该大学所处城域网(MAN),CL_3 是城市所处更高级广域网络(WAN),虽然在 CL_1 级校园网内部发生消息通信的比例非常大,但仍然有相当比例的消息路由必须通过 CL_2 级网络甚至 CL_3 级网络.统计数据表明,特定子网下,数据查询过程的重复率非常高,这是因为网络中存在一些访问率高的热门节点和热点信息[19].就 HASN 路由模型本身而言,如果 CL_1 级聚集下节点每次查询同一个键值都需要进入到 CL_2 级甚至 CL_3 级 Hash 环域,则必将造成较大的路

由时间开销.

针对这个问题,为了降低节点重复跨级路由过程造成的时间开销,我们提出了在底层聚集(即 CL_1 级环域)内进行数据项备份的机制,其基本思想就是按照重复率大小,有选择地备份一部分跨越 CL_1 级路由过程返回的数据项值(data item),并按照 HASN 定义的键值存放规则,将其存放在 CL_1 聚集下相应的归属节点上.这样,在下一次查找相同键值时,路由过程就可以在 CL_1 级聚集内完成,而不需要进入到更高一级的网络查询层,从而极大地提高了路由效率.需要说明的是,虽然通过数据项备份机制来提高路由效率是建立在增加节点存储开销的基础上,但事实上,HASN 在 CL_1 集节点上备份的数据项并不是实际数据对象,而仅仅是对应某数据对象存储地址的一个数据项(data item),因此不会对节点增加过多的存储开销.

3.2 环路保持机制(hash ring preserving)

对于基于 DHT 的分布式路由协议,如何在网络拓扑结构发生变化时保持 Hash 环路完整性是一个很关键的问题.Hash 环路完整性是确保整个网络路由发现和路由维护过程正常进行的必要条件,环路的不完整将会导致网络处于不收敛状态.虽然 HASN 提供了针对网络拓扑变化的路由更新机制,但在动态自组织网络中,节点的频繁加入或退出仍有可能导致在系统路由更新尚未完成就发生了键值查询操作,而 Hash 环路的不完整将造成查询中断或失败.

针对这个问题,HASN 引入了环路保持机制,其基本思想就是增加前继和后继节点备份表,将 Hash 环上最邻近的 R 项 NID 大于本节点的节点作为后继节点备份,而 R 项 NID 小于本节点的节点作为前继节点备份,并设定 $R=O(\log N)$.一旦本节点的后继节点崩溃,就立即使用后继节点备份表中保存的第 2 后继(第 2 个 NID 大于本节点的节点)作为自己临时的后继节点,以保持 Hash 环路的完整.同样,当本节点前继节点突然崩溃,则使用邻接表中保存的第 2 前继(第 2 个 NID 小于本节点的节点)作为自己临时的前继节点,以保持 Hash 环路的完整.因此,在 HASN 模型中,仅当节点备份表中对应的 R 项备份前继和后继节点同时崩溃时,Hash 环路才可能出现中断的情况,而这种概率是非常小的,假设每个备份节点崩溃的概率为 1/2,则 $P(R$ 个备份节点同时崩溃$)=O(1/N)$.

4 HASN 路由模型的性能分析

4.1 系统环境

在 Intel P4 1.6G 处理器微机和 Linux Redhat 8.0 环境下,我们使用 NS-2 网络仿真器实现了对实际网络状况的模拟,并通过 Otcl 和 C++程序设计分别实现了 Chord 和 HASN 两种路由算法.在 Internet 网络拓扑生成方面,我们使用 GT-ITM 拓扑发生器生成网络随机拓扑图,并采用了较能代表当前 Internet 结构的穿通-末端 TS(transit-stub)模型(网络拓扑图如图 5 所示)[20].同时,我们通过调整仿真参数 congestion 和 Traffic 值,对网络中存在的动态干扰流进行了模拟.

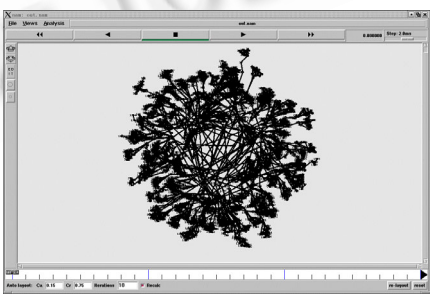


Fig.5 Network topology based on transit-stub model

图 5 基于 TS(transit-stub)模型的网络拓扑结构图

4.2 基本性能分析

路由协议性能主要由存储开销、时间开销、路由维护分组开销 3 方面来衡量,在本节中,我们将分别从以上 3 个方面来分析 HASN 路由模型的性能,并通过与 Chord 路由模型进行比较,讨论 HASN 在路由协议性能方面的改善,在此基础上,将进一步通过仿真实验进行验证.下面首先给出一些相关定义:

- N 代表全聚集(full cluster,简称 FC)下节点的个数;
- N_k 代表第 k 级聚集(cluster level k,CL_k)下节点的个数;
- RTT 代表全聚集下节点间的平均往返时间(average round trip time);
- RTT_k 代表第 k 级聚集下节点间的平均往返时间;
- TC 代表在全聚集上完成一次路由查询所需的时间开销;
- TC_k 代表在第 k 级聚集上完成一次路由查询所需的时间开销.

4.2.1 路由协议时间开销

首先,我们将讨论 HASN 模型中分级结构的引入对路由效率和时间开销带来的改善,这也是 HASN 模型设计的主要动机之一.

与 Chord 一样,HASN 模型在任意一个聚集层完成一次路由查询经过的平均消息传输时延(average message transfer delay,简称 AMTD)与网络节点总数 N 的关系为

$$AMTD \infty O(\log N) \tag{6}$$

其中 N 是该聚集层节点的总数.实际上,路由效率还与一些不确定因素有关,如查询请求分布率(uniform request distribution,简称 URD)以及系统对查询结果精确度(result precision,简称 RP)的不同要求.一般情况下,可以在 $\log N$ 前增加一个附加常数 α 来简化这些不确定因素,通常,$\alpha=1/2$[10].

综上所述,在全聚集层(FC)进行一次路由查询所预期的时间开销(time consumption,简称 TC)如式(7)所示,这也就是 Chord 路由模型中一次路由查询所预期的时间开销:

$$TC = RTT \times \alpha \times \log N \tag{7}$$

与之相比,在具有 H 级聚集层结构的 HASN 路由模型下完成一次路由查询所预期的时间开销 TC_H 为

$$TC_H = RTT_k \times \alpha \times \log N_k + RTT_{k+1} \times \alpha \times \log(N_{k+1}/N_k) + ... + RTT \times \alpha \times \log(N/N_{H-1}) \tag{8}$$

由式(7)、式(8)可得,

$$TC/TC_H = \frac{RTT \times \log N}{RTT_0 \times \log N_0 + RTT_1 \times (\log N_1 - \log N_0) + ... + RTT \times (\log N - \log N_{H-1})} = \beta \tag{9}$$

我们将 β 称作加速比(speedup),它代表 HASN 路由模型与原 Chord 路由模型在路由协议时间开销和路由协议本身效率上的改进.

对于一个二级 HASN 路由模型,即 H=1,由式(9)可得:

$$\beta = \frac{RTT \times \log N}{RTT_0 \times \log N_0 + RTT \times (\log N - \log N_0)}.$$

当 RTT_0 足够小时,上式可以简化为

$$\beta = \frac{\log N}{\log N - \log N_0}.$$

假设 N 和 N_0 呈指数关系,即 $N_0=N^x$,则

$$\beta = \frac{1}{1-x}.$$

当 $N_0=N^{1/2}$ 时,β=2;而当 $N_0=N^{1/4}$ 时,β=4/3.由此可见,基于分级聚集层结构中节点数量的不同比例关系,HASN 模型的路由协议时间开销有不同比例的缩短.值得一提的是,随着 HASN 系统进入稳定状态和数据项备份机制的不断完善,HASN 路由模型的路由协议时间开销将进一步缩短,这一点在路由模型仿真实验中将得到进一步证明.

4.2.2 路由协议存储开销

与传统的路由方式相比,在 Chord 和 HASN 路由模型中节点都无须保存系统中每个其他节点的路由信息,

而仅需保存 $O(\log N)$个表项的路由信息,因此,存储开销得到了很大的改善,提高了整个系统的可扩展性.

与 Chord 相比,HASN 路由模型通过引入私有环域 IRZ,并采取大小可以动态变化的路由表,增加了系统的灵活性,还通过缩小路由表项(取消路由环域终点值等路由表项),进一步降低了路由表的存储开销.

但由于在 HASN 路由模型中引入了多级聚集结构,因此会在一定程度上增加路由协议的存储开销.在 HASN 路由模型,每增加一级聚集将会为每个节点增加一个具有 $\log N_C$ 个路由表项的分级路由表,其中 N_C 是该级聚集下的网络节点个数.鉴于分级聚集结构中节点数量远远小于网络全聚集 FC 下的节点数,因此,增加一级聚集给系统增加的存储开销是相对有限的,它随着分级聚集中的节点数量 N_k 的增加而增加.对于一个均匀分割的具有二级结构的 HASN 系统而言(即 $N=N_C^2$),增加一级聚集给系统增加的存储开销不会超过全聚集存储开销的 50%($\log N_C/\log N$).同时,随着 HASN 模型分级层数和聚集体数量的增加,每一个聚集体中的节点数量将进一步减少,因此 HASN 路由存储开销将进一步降低.而由上一节的分析可知,分级聚集中的节点数量 N_C 越大,系统的加速比β就越大,因此,HASN 路由模型时间开销的降低是以存储开销为代价的,但增加的存储开销也是在系统允许范围之内的.

4.2.3 路由维护分组开销

在路由控制分组开销上,HASN 路由模型和 Chord 基本上是一致的,但 HASN 取消了 Chord 中全网周期性路由更新方式,其路由更新完全采取事件触发模式,一个节点加入和退出系统会触发网络的路由更新过程,整个过程发送的消息数不超过 $O(\log^2 N)$[10].因此,HASN 路由模型更能够实时保证网络的收敛性,并消除了不必要的路由维护开销.

但由于在 HASN 路由模型中引入了多级聚集结构,因此会在一定程度上增加路由协议维护开销.对于一个均匀分割的具有二级结构的 HASN 系统而言(即 $N=N_C^2$),增加一级聚集给系统增加的路由维护开销不会超过全聚集路由维护开销的 25%($\log^2 N_C/\log^2 N$).同时,随着 HASN 模型分级层数和聚集体数量的增加,每一个聚集体中的节点数量将进一步减少,因此,HASN 路由维护开销也将进一步降低.所以,HASN 路由模型增加的路由维护开销也是在系统允许范围之内的,这一点在仿真实验中可以得到验证.

4.3 实验结果

4.3.1 平均消息传输时延

平均消息传递时延(AMTD)是评测 HASN 路由模型性能的重要指标,图 6 的模拟结果分别记录了优化前后的 HASN 以及 Chord 模型中 AMTD 和节点数量 N 之间的关系.由图可见,尽管两种模型平均消息传递时延和节点个数 N 均呈 $O(\log N)$关系,但由于 HASN 路由模型充分利用了节点网络邻近性特征(network proximity character),其在性能上相对 Chord 有了很大改善,特别是应用了优化策略后的 HASN 模型.

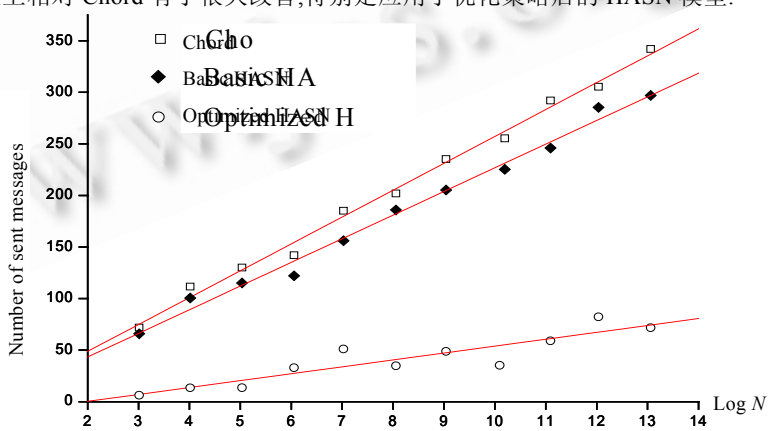


Fig.6 Relationship of AMTD and log$N$

图 6 平均消息传输时延与 log$N$ 关系

4.3.2 节点加入和退出系统平均发送消息数

节点加入和退出系统平均发送消息数是评测 HASN 路由模型自组织性能的重要指标,图 7 和图 8 的模拟结果分别记录了 HASN 和 Chord 模型中节点加入和退出系统发送消息数量与 $\log^2 N$ 之间的关系.由图可见,随着网络节点数量的增加,HASN 模型的分级层数和聚集体数量也随之增加,两种模型下节点加入和退出系统平均发送消息数与网络节点总个数 N 均呈 $O(\log^2 N)$关系,且性能趋于一致.这一实验结果有效地验证了 HASN 模型分级结构增加的路由维护开销是在系统可接受范围之内的.

4.3.3 本地化路由(path locality)

HASN 路由模型的另一个优点就是充分利用了网络局域性特征(network locality character),通过引入数据项备份机制,实现了数据项的局域化存放(localized placement).在进一步提高网络查询效率的基础上,HASN 还提高了网络的安全性,并且特别适合校园和企业内部网络,因为系统保证了本地两台主机之间的消息不会被路由到本地子网络之外,从而真正实现了本地化路由.图 9 的模拟结果分别记录了 HASN 和 Chord 路由模型下消息路由经过物理网络跳数(physical network hops)与本地查询比例之间的关系.由图可见,由于 Chord 路由模型本身在设计上并没有考虑到网络局域性特征这一因素,因此其查询消息经过的物理网络跳数值并不随着本地资源查询比例的增长而变化.与之相比,HASN 路由模型充分考虑了网络局域性特征,因此其查询消息经过的物理网络跳数值随着本地消息查询比例的增长而迅速降低.

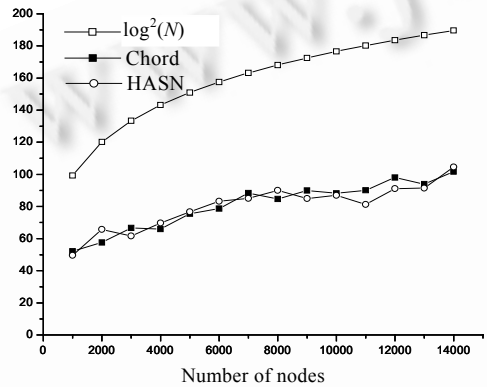


Fig.7 Messages sent vs. $\log^2 N$ when node joins

图 7 新节点加入系统平均发送消息数与 $\log^2 N$ 关系

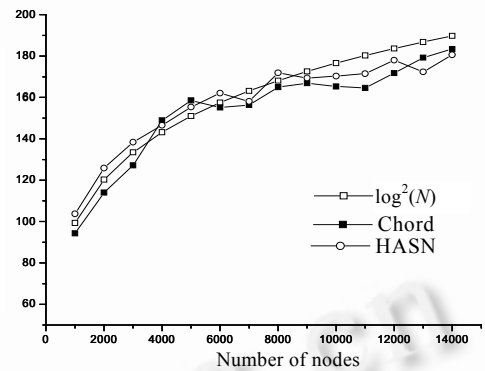


Fig.8 Messages sent vs. $\log^2 N$ when node leaves

图 8 节点退出系统平均发送消息数与 $\log^2 N$ 关系

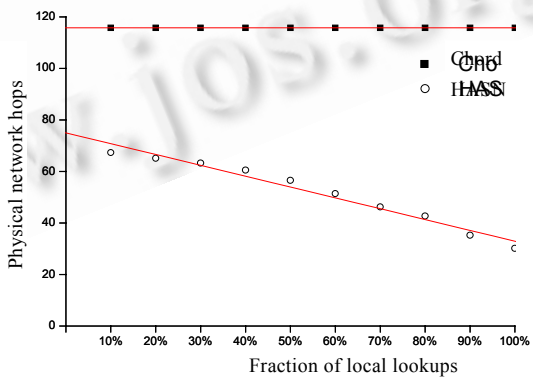


Fig.9 Physical network hops vs. local lookups

图 9 消息路由跳数与本地资源查询比例关系

5 相关的研究工作

P2P 系统以飞快的速度发展成为 Internet 中最重要的应用系统之一,其体系结构也由中央集权式的 Napster

向非结构化覆盖层网络(unstructured overlay network,简称 UON)Gnutella,Freenet 逐步演变.但是,由于 UON 的随意性,使得数据查询必须依靠广播或随机多步(multiple random walks)搜索来完成[21],耗费大量的网络开销,降低了系统的可扩展性.于是,近年来提出的 SON 以及基于 SON 的 DHT 算法成为研究领域的热点,主要算法包括 Tapestry,CAN,Chord 和 Pastry.

同时,一些新的 SON 算法也相继提出:Kademlia[22]使用 nodeId 进行 Xor 运算的结果作为 nodeId 之间的距离,从而在 Pastry 的基础上抛开了叶子节点集合(leaf set),使得协议更加简洁;Weatherspoon 和 Kubiatowicz[23]讨论了 Tapestry 中指针的维护方法,并指出离得越远的指针探测的频率应该越小;Mahajan 等人[24]提出了随着系统结点变化而动态调整指针探测频率的方法;eCAN[25]在 CAN 的基础上进一步引入加速指针.

Brocade[26]也提出了一个类似于 HASN 的基于 Internet 物理拓扑的分级覆盖层网络模型,它采用半集中式(semi-centralized)结构,通过在覆盖层网络引入强节点,将物理距离较近的对等点聚集成组.一个强节点起的作用类似于一个网络区域的界标(landmark),每个组群的强节点都是在 Brocade 组网之前预设的,并且通常由子网的网关路由器担当.与 Brocade 不同,HASN 模型是一个具有完全自组织机制的系统.HASN 中的每台对等机都以匿名方式登录网络,并根据其加入先后和物理位置自动成为分级体系中的 HA 或者普通子对等机,整个过程完全以自组织形式完成,系统不需要任何服务器的支持.

6 总结及下一步研究方向

在自组织网络技术领域,无线移动自组织网络(mobile ad hoc network,简称 MANET)一直是专家学者们研究的热点[27,28],但对基于 Internet 的有线自组织网络技术的研究却非常少.HASN 是一个基于 Internet 拓扑架构和 P2P 计算模式的自组织网络路由模型,它通过使用基于 DHT 的杂凑式路由算法 HASN_Scale,继承了结构化 P2P 覆盖层网络路由协议的所有优点,保证了网络的可扩展性;通过引入基于网络实际拓扑的网络分层和组群划分机制,充分利用节点的网络邻近性和网络局域性,进一步提高了路由性能.基于 P2P 计算模式的自组织网络路由模型的专利申请已被国家知识产权局受理.

下一步的研究将着重于把 P2P 计算模式的应用推广到 MANET 领域,使运行在逻辑命名空间(Namespace)的 P2P 覆盖层网络协议功能与运行在物理命名空间的 MANET 路由协议无缝地结合起来,提出一种基于 P2P 计算模式的新型多跳(multi-hop)网络路由协议.

致谢 在此,向加利福尼亚大学的 Jakob 博士和斯坦福大学的 Srikath 博士表示致敬和谢意,在本文撰写过程中,我们通过 E-mail 进行了多次非常有益的讨论和交流,他们为本文的工作提出了很多宝贵意见.

References:

[1] Yemini Y, Trito S. Nestor: Technologies and protocols for self-managed and self-organizing networks. 1998. http://www.cs.columbia.edu/dcc/nestor

[2] Pottie GJ, Clare LP. Wireless integrated network sensors: Towards low cost and robust self-organizing security networks. In: Proc. of the SPIE Int'l Symp. on Enabling Technologies for Law Enforcement and Security. 1998. 106−112.

[3] Li ZP, Huang JH, Huang DY, Zhuang L. Introduction to Peer-to-Peer networking technology and development. Telecommunications Science, 2003,19(3):1−4 (in Chinese with English abstract).

[4] Fox G. Peer-to-Peer networks. Web Computing, 2001,3(3):75−77.

[5] Saroiu S, Gummadi KP, Dunn RJ, Gribble SD, Levy HM. An analysis of Internet content delivery systems. In: Proc. of the 5th Symp. on Operating Systems Design and Implementation (OSDI 2002). 2002. 86−90.

[6] C-NET NEWS. Napster among fastest-growing Net technologies. 2000. http://news.com.com/2100-1023-246648.html

[7] Clarke I, Sandberg O, Wiley B, Hong TW. Freenet: A distributed anonymous information storage and retrieval system. In: Workshop on Design Issues in Anonymity and Unobservability. 2000. 25−31.

[8] Gnutella. 2003. http://www.gnutella.com/

[9] Ratnasamy S, Francis P, Handley M, Karp R, Shenker S. A scalable content-addressable network. In: Annual Conf. of the Special Interest Group on Data Communication (SIGCOMM 2001). 2001. 168−175.

[10] Stoica I, Morris R, Karger D, Kaashoek MF, Balakrishnan H. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Annual Conf. of the Special Interest Group on Data Communication (SIGCOMM 2001). 2001. 124−137.

[11] Rowstron A, Druschel P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In: Int'l Conf. on Distributed Systems Platforms (Middleware 2001). 2001. 135−141.

[12] Zhao B, Kubiatowicz J, Joseph A. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, Computer Science Division, U. C. Berkeley, 2001. 106−115.

[13] Jain S, Mahajan R, Wetherall D. A study of the performance potential of DHT-based overlays. In: Proc. of the 4th USENIX Symp. on Internet Technologies and Systems (USITS 2003). 2003. 256−261.

[14] Dabek F, Kaashoek MF, Karger D, Morris R, Stoica I. Wide-Area cooperative storage with CFS. In: Proc. of the 18th ACM Symp. on Operating Systems Principles (SOSP 2001). Chateau Lake Louise, 2001. 344−352.

[15] Li ZP, Zhao XB, Huang JH. Construction of user registration and user evaluating system in peer-to-peer network. Telecommunications Science, 2004, 20(5):14−18 (in Chinese with English abstract).

[16] Glewin D. Consistent hashing and random trees: Algorithms for caching in distributed networks [MS. Thesis]. Department of EECS, MIT, 1998. http://thesis.mit.edu/

[17] Karger D, Lehman E, Leighton F, Levine M, Lewin D, Panigrahy R. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In: Proc. of the 29th Annual ACM Symp. on Theory of Computing. El Paso, 1997. 654−663.

[18] Gupta A, Liskov B, Rodrigues R. One hop lookups for peer-to-peer overlays. In: Proc. of the 9th Workshop on Hot Topics in Operating Systems (HOTOS IX). 2003. 452−458.

[19] Ratnasamy S, Shenker S, Stoica I. Routing algorithms for dhts: Some open questions. In: Proc. of the IPTPS02. Cambridge, 2002. http://www.cs.rice.edu/Conferences/IPTPS02/

[20] Zegura E, Calvert KL, Bhattacharjee S. How to model an internetwork. In: Sohraby K, ed. Proc. of the IEEE Infocom'96. San Francisco: IEEE Computer Society Press, 1996. 594−602.

[21] Lv Q, Cao P, Cohen E, Li K, Shenker S. Search and replication in unstructured peer-to-peer networks. In: Proc. of the 16th ACM Int'l Conf. on Supercomputing (ICS 2002). 2002. 254−261.

[22] Maymounkov P, Mazieres D. Kademlia: A peer-to-peer information system based on the XOR metric. In: Proc. of the 1st Int'l Workshop on Peer-to-Peer Systems (IPTPS 2002). 2002. 153−161.

[23] Weatherspoon H, Kubiatowicz J. Efficient heartbeats and repair of softstate in decentralized object location and routing systems. In: Proc. of the ACM SIGOPS European Workshop 2002. 2002. 231−238.

[24] Mahajan R, Castro M, Rowston A. Controlling the cost of reliability in peer-to-peer overlays. In: Proc. of the 2nd Int'l Workshop on Peer-to-Peer Systems (IPTPS 2003). 2003. 368−374.

[25] Xu Z, Zhang Z. Building low-maintenance expressways for P2P systems. Technical Report HPL-2002-41, Palo Alto: Hewlett-Packard Lab., 2002.

[26] Zhao BY, Duan Y, Huang L, Joseph A, Kubiatowicz J. Brocade: Landmark routing on overlay networks. In: Proc. of the 1st Int'l Workshop on Peer-to-Peer Systems (IPTPS 2002). 2002. 564−570.

[27] Perkins C, Belding-Royer EM, Das SR. Ad hoc on-demand distance vector (AODV) routing. Network Working Group RFC 3561, 2003.

[28] Èapkun S, Buttyán L, Hubaux JP. Self-Organized public-key management for ad hoc networks. IEEE Trans. on Mobile Computing, 2003,2(1):203−213.

附中文参考文献:

[3] 李祖鹏,黄建华,黄道颖,庄雷.P2P 网络技术的发展与展望.电信科学,2003,19(3):1−4.

[15] 李祖鹏,赵修斌,黄建华.P2P 网络用户注册和评估系统的建立.电信科学,2004,20(5):14−18.