

MANET 中基于 DHT 的资源共享技术

侯祥松¹, 曹元大², 张煜¹

(1. 北京理工大学计算机科学技术学院, 北京 100081; 2. 北京理工大学软件学院, 北京 100081)

摘要: P2P 因良好的可扩展性、低廉的部署开销及能够聚集闲置资源等优点而被广泛应用于文件共享领域。移动自组网(MANET)是一种具有高度动态拓扑结构、节点任意移动的自组织网络, 与 P2P 有很多共同点, 近来开始出现了将两者融合的研究。该文提出一种在 MANET 环境下基于结构化 P2P 的资源共享模型, 将 MANET 中的节点按其物理位置组成多个域, 每个域中有若干节点, 域和域中的节点用 CAN 管理。模拟结果显示该模型可以明显减少路由所需的跳数。

关键词: 对等网络; 移动自组网; 资源共享

Resource Share Technology Based on DHT in Mobile Ad Hoc Networks

HOU Xiang-song¹, CAO Yuan-da², ZHANG Yu¹

(1. School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081;

2. School of Software, Beijing Institute of Technology, Beijing 100081)

【Abstract】 While largely developed independently of each other, P2P overlay networks in Internet and Mobile Ad Hoc Networks(MANET) share many key characteristics such as self organization and decentralization. This paper provides an overview of existing service discovery approaches in MANET. In order to overcome their drawbacks, it presents a structured P2P system R-CAN to reduce routing hops in MANET, and the entire system consists of a set of regions where peers are in proximity. Experimental results show that R-CAN can greatly enhance the system performance and effectively reduce the routing hops.

【Key words】 P2P network; Mobile Ad Hoc Networks(MANET); resource share

1 概述

近年来, P2P 由于具有很好的可扩展性、低廉的部署开销、能够聚集闲置资源等优点而被广泛地应用于文件共享领域。随着移动无线通信技术的发展及各种移动设备在用户中的日益普及, 移动自组网(Mobile Ad Hoc Networks, MANET)受到越来越多的关注。MANET 是一种具有高度动态拓扑结构、节点任意移动的自组织网络, 没有固定的中心设备, 所有节点都可以充当中继、路由、网关等角色, 实现各节点之间的互通互连, 并可实现和上级网络的连接。由于 P2P 和 MANET 都具有频繁变化的拓扑, 并且均是基于 hop-by-hop 链接的分布式结构, 因此两者可以融合。

文献[1]将 Pastry 应用到 DSR^[2]上, 提出了 MANET 路由协议 DPSR, 每个节点利用 SHA-1 获得唯一的 ID, 每个查询消息的目标地址也哈希到和节点相同的空间, 然后利用 Pastry 的路由算法找到对应的节点, DPSR 中每个节点只需要维护 $O(\log n)$ 个路由信息。文献[3]以 CHORD 作为基础, 将 PNS^[4]技术和 MANET 结合。实验结果显示, 稀疏网络的性能与固定网络差不多, 随网络密度的增加下降。

2 R-CAN 模型

本文提出一个基于域概念的 R-CAN, 将邻近的 Ad Hoc 节点组成一个域, 域和域中的节点均采用 CAN^[5-6]管理, 这种结构可以降低路由维护开销, 减少查找的跳数。

2.1 节点的管理

对于一个 d 维空间的 R-CAN, 所有节点的标识为一个

d 维的向量。 d 维向量被分为 2 个部分, 不失一般性, 设前 $d/2$ 维表示节点所在的域(RegionID), 后 $d/2$ 维表示节点在域中的位置(NodeID)。NodeID 使用节点自身特征生成, 是节点在网络中具有唯一性的地址标识。RegionID 可以使用域中的一个节点 ID 标识, 同样具有唯一性, 一个域表示一些物理上相邻的节点集合。当节点移动时, 它会从一个域移动到另一个域, 节点 RegionID 随之改变, 而 NodeID 一直保持不变。

通常在 2 种情况下会引发节点加入操作: (1) 一个节点第 1 次加入 MANET; (2) 节点离开了原来域的范围并进入了一个新的域范围。当节点离开网络或移动到所属域外时, 必须将离开节点管理的区域分配给域中的其他相邻节点。同一域内的节点共同负责该域所分配的空间, 在一个域内, 相当于一个小的 CAN 网络, 在域内节点的加入和离开算法与 CAN 相同。当节点 N 加入系统, N 首先广播一个 Join 消息, 所有收到消息的物理相邻节点返回自己的 RegionID, 然后节点 N 从所有结果中选择距离最近的域作为加入的域。假设 N 加入域 R , 先将自己的 RegionID 设置为 R , 然后在域 R 中找到自己 NodeID 所在的空间, 接着负责该空间的节点分裂, 同时将自己的一部分空间分配给节点 N 。完成节点加入操作后, 节点的 ID 为 (R, NodeID) 。

作者简介: 侯祥松(1977—), 男, 博士研究生, 主研方向: 分布式计算; 曹元大, 教授、博士生导师; 张煜, 博士研究生

收稿日期: 2008-07-05 **E-mail:** xiangsong.hou@gmail.com

R-CAN 中每个节点维护 2 个路由表: 存储相邻域信息的域路由表和存储域内相邻节点信息的节点路由表。节点路由表通过路由转发、域内节点间相互交换信息更新。由于域内节点间是物理相邻的, 因此维护节点路由表所需的开销很小。域路由表通过转发路由信息进行学习, 当一个节点发现其域路由表有变化时, 将新的域路由表转发到域内其他节点中。当节点离开或加入一个域时, 该节点及邻居节点需要更新其节点路由表和域路由表。节点路由表的更新过程与 CAN 的算法相同, 只是路由更新局限在同一个域内。新加入节点的域路由表先是简单地从其邻居获得, 然后在路由转发过程中逐渐学习。

2.2 域的管理

系统开始时只有一个节点, 并且该节点把 RegionID 设置为自己的 NodeID, 在其他节点加入时, 执行 JOIN 操作。当域中节点数达到阈值时, 触发域分裂算法, 当节点低于一定数量时, 触发域合并操作。域的分裂算法将域分成同样大小的 2 块, 逻辑上相邻的节点分到同一个域中, 域中的节点更新域路由表。当一个域中节点数低于某个阈值时, 该域和相邻的域合成一个大的域并更新路由表。当域进行合并和分裂操作时, 需要更新节点路由表和域路由表。域合并时相当于有新的节点加入到域中, 此时路由表的更新与节点加入时相同。域分裂时只是将分裂点的节点路由表进行更新, 而域路由表可以直接从分裂点获得。

图 1 给出了一个 2 维 R-CAN, 其中, 域分裂阈值为 4; 域合并阈值为 1。当节点 11 离开域, 域 3 中只剩下一个节点, 触发了域合并操作; 节点 11 从邻居域 2 和邻居域 4 中选择节点数少的域 2 作为其合并对象, 合并后域 2 中有 3 个节点 4, 16, 17。当节点 20 加入域 5 时, 节点数超过了域合并阈值 4, 触发了域分裂操作; 域 5 分裂成域 5 和域 8, 然后再将节点 20 加入域 5 中。

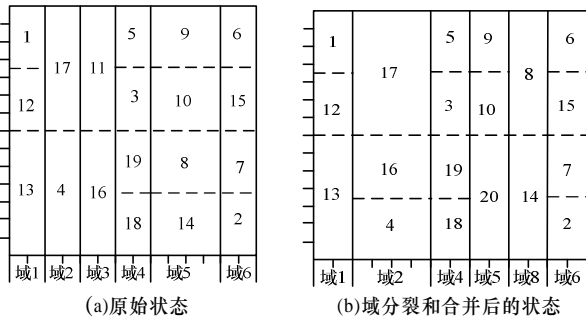


图 1 域的管理

2.3 路由

在传统的 CAN 中, 路由时节点只要朝着目标节点的方向把请求转发给自己的邻接节点, 每个接收到消息的节点先检查本地是否有需要的数据, 如果有就直接返回, 否则把请求转发给自己的邻接节点, 这样就能找到目标节点。

由于 R-CAN 将物理上相邻的节点组成一个域, 因此路由算法会有所不同。这里查询请求是由 (RegionID, NodeID) 2 部分组成的 d 维向量。在节点 N 查询一个节点 $Q(Qr, Qn)$ 时, 路由算法如下:

- (1) 节点 N 根据路由的目的域 Qr 查询本地域路由表, 将其转发至更接近目标的邻居域内的节点。
- (2) 接收到查询的节点比较目的域和本地域, 如果相同则转到步骤(3); 否则根据域路由表继续转发, 直至路由到目的

域中一个节点 T 上。

- (3) 节点 T 根据本地的节点路由表找到目标节点 D 。

如图 2 所示, 节点 7 要路由到节点 1, 首先根据域路由表经过 14, 20, 19, 16, 最后到达目标节点所在域内的节点 13; 节点 13 根据节点路由表在域 1 内经过 12 最后到达目标 1。

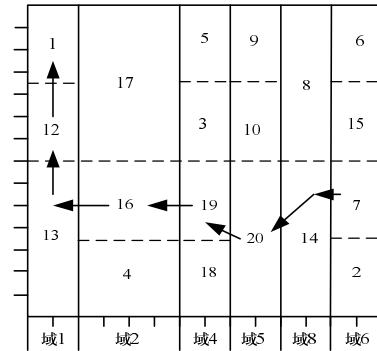


图 2 路由过程

2.4 资源发布和查找

在一个 d 维的 R-CAN 中, 当节点 N 发布资源 R 时, 将资源 R 用 2 个哈希函数 Hr 和 Hn 映射为 2 个 $d/2$ 维的向量, $Rr=Hr(R)$ 表示负责资源 R 的节点所处的域 ID, $Rn=Hn(R)$ 表示资源 R 在域 Rr 中的具体位置。在发布资源 R 时, 首先利用路由算法找到负责资源 (Rr, Rn) 的节点, 并将资源 R 的索引存储在节点中。

当一个节点 N 查找资源 Q 时, 首先计算 $Qr=Hr(Q)$, $Qn=Hn(Q)$, 然后根据路由算法找到目标节点, 根据目标节点负责的索引找到存储资源 Q 的节点地址, 并将地址返回给 N 。然后发起点 N 根据返回的地址直接从相应节点获取资源。

3 实验结果

本文建立了一个模拟的 MANET, 网络中的节点在一个 $500\text{ m} \times 500\text{ m}$ 的正方形区域中以随机路径点模型 (random waypoint model)^[2,6] 移动。所有节点在取样的 1 800 s 中停顿时间和运动时间在 60 s~120 s 间随机均匀选出。实验中的分裂阈值为 20, 合并阈值为 5。为了评价 R-CAN 的性能, 同时直接将 CAN 作为 MANET 中节点的管理方法, 比较两者在不同的实验环境下路由时 MANET 网络层所需要的跳数。

图 3 给出了运动速度对性能的影响。从中可以看出, 随着节点运动速度的增加, 路由所需的跳数也增加。因为节点运动速度的增加意味着拓扑结构的变化, 速度越快, 拓扑变化越快。如果路由表的更新速度落后于拓扑变化的速度, 路由过程中会出现不匹配现象。但由于 R-CAN 能将物理相邻的节点聚合在一个域内, 节点的小范围移动只影响节点路由表, 并不影响整个路由过程, 因此 R-CAN 可以在拓扑快速变化的无线网络中获得比 CAN 更好的性能。

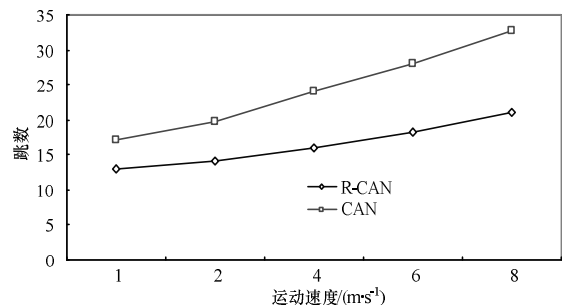


图 3 节点运动速度对跳数的影响

为了验证系统规模对跳数的影响,让节点的运动速度在 0~8 m/s 中随机选择, R-CAN 的维度为 4。图 4 给出了实验过程中节点数对路由跳数的影响,随着节点数的增加,虽然两者的跳数都相应增加,但 R-CAN 的跳数明显低于 CAN。这是因为 R-CAN 让物理相近的节点聚合在同一个域中, R-CAN 域数目的增加远小于 CAN 中节点数的增加,大大降低了路由的跳数。

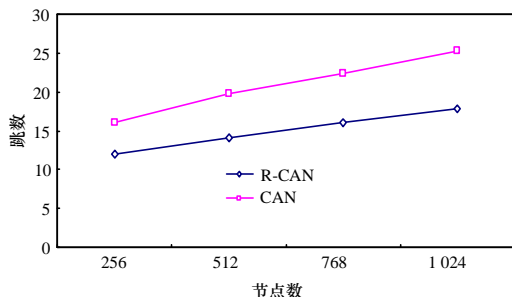


图 4 节点数对跳数的影响

为了验证 DHT 结构中的维度对跳数的影响,使系统的节点数固定为 512 个,节点的运动速度在 0~8 m/s 中随机选择。如图 5 所示,每个节点的邻居数随着维度的增加而增加,因此,找到目标节点更容易。虽然两者的跳数都随维度的增加减少,但 R-CAN 的跳数明显少于 CAN。

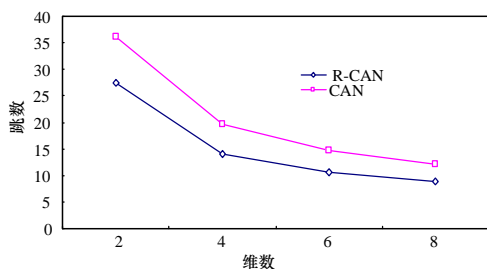


图 5 维数对跳数的影响

(上接第 108 页)

实验表明, Chord 系统中节点组织方式的结构化使查询具有一定的确定性,平均查询跳数最少。本模型的数据项备份机制使重复查询仅需一跳,较 Gnutella 系统,查询跳数也大幅减少。然而作为查询经过的逻辑路径长度,查询跳数并不像查询时延那样真实反映实际的搜索性能。在查询时延方面,本模型在三者中最小,这表明模型通过充分利用节点物理临近性,克服了 Chord 系统中严重的查询绕路问题。

最后需要说明的是,实验中的节点个数为模型中超级节点,并不包含一般客户节点。因此,在保证更优性能的前提下,模型能够容纳更多的节点,具有更好的扩展性。

5 结束语

本文提出了一种新的 P2P 搜索方法,并基于此构建了轻量级混合式搜索模型。分析和实验表明该模型很好地克服了查询绕路问题,有效地减少消息冗余,具有较高的搜索效率和较好的扩展性能。如何在模型中加入管理与监督机制是下一步的研究目标。

参考文献

[1] Shirky C. What is P2P and What Isn't[C]//Proc. of O'Reilly's Emerging Technology Conference. NY, USA: [s. n.], 2002-05.

4 结束语

本文提出了一种在移动 Ad Hoc 环境下基于结构化 P2P 的资源共享模型 R-CAN,该模型将移动 Ad Hoc 中的节点按其物理位置组成多个域,每个域中有若干节点,域和域中的节点分别用 CAN 管理,每个节点维护域路由表和节点路由表。对于资源使用 2 个不同的哈希函数获得其域 ID 和节点 ID,路由时节点先把请求转发给相应的域,然后在域内找到存储该资源的节点。模拟结果显示,该模型可以明显减少路由所需的跳数。

参考文献

[1] Charlie H Y, Das S M, Pucha H. Exploiting the Synergy Between Peer-to-Peer and Mobile Ad Hoc Networks[C]//Proceedings of the 9th Conference on Hot Topics in Operating Systems. Hawaii, USA: [s. n.], 2003.

[2] Johnson D B, Maltz D A. Dynamic Source Routing in Ad Hoc Wireless Networks[M]//Imielinski T, Korth H. Mobile Computing. New York: Kluwer Academic Publishers, 1996.

[3] Cramer C, Fuhrmann T. Proximity Neighbor Selection for a DHT in Wireless Multi-hop Networks[C]//Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing. [S. l.]: IEEE Press, 2005.

[4] Zhang Hui, Goel A, Govindan R. Incrementally Improving Lookup Latency in Distributed Hash Table Systems[C]//Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. [S. l.]: ACM Press, 2003.

[5] Ratnasamy S, Francis P, Handley M, et al. A Scalable Content-addressable Network[C]//Proc. of ACM SIGCOMM'01. [S. l.]: ACM Press, 2001.

[6] Camp T, Boleng J, Davies V. A Survey of Mobility Models for Ad Hoc Network Research[J]. Wireless Communications and Mobile Computing, 2002, 2(5): 483-502.

[2] Napster Free[Z]. (2001-10-20). <http://free.napster.com>.

[3] OSMB. Gnutella[Z]. (2003-10-20). <http://www.gnutella.com>.

[4] Downloadmp[Z]. (2008-05-20). <http://www.kazza.com>.

[5] Stoica I, Morris R, Karger D, et al. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Application[C]//Proc. of ACM SIGCOMM'01. [S. l.]: ACM Press, 2001.

[6] Kalogeraki V, Gunopulos D, Zeinalipour-yazti D. A Local Search Mechanism for Peer-to-Peer Networks[C]//Proceedings of the 11th ACM Conference on Information and Knowledge Management. [S. l.]: IEEE Computer Society, 2002.

[7] Jawhar I, Wu J. A Two-level Random Walk Search Protocol for Peer-to-Peer Networks[C]//Proceedings of the 8th World Multi-conference on Systemic Cybernetics and Informatics. [S. l.]: IEEE Computer Society, 2004.

[8] Marzolla M, Mordacchini M, Orlando S. Tree Vector Indexes: Efficient Range Queries for Dynamic Content on Peer-to-Peer Networks[C]//Proceedings of the 14th International Conference on Parallel Distributed and Network-based Processing. [S. l.]: IEEE Computer Society, 2006.