

一种改进的 NEMO 协议

张 晴,禹继国

(曲阜师范大学 计算机科学学院,山东 日照 276826)

摘 要: NEMO 协议在组播过程中没有利用层次结构最底层的节点,并且在选择簇首时也没有考虑到节点异构性。针对这些问题,提出了一种改进的 NEMO 协议。改进的 NEMO 协议通过分簇技术充分利用了最底层的节点,避免了 P2P 覆盖网络中的搭便车现象,并且在选择簇首时考虑了节点的异构性。每一个簇首根据簇中剩余节点的能力选择一些节点作为辅助领导节点,当簇首失效时,就从这些辅助领导节点中选择能力最大的节点作为新的簇首,从而提高了覆盖网络的容错性。仿真表明,改进后的 NEMO 协议在平均传递延迟和节点失效时的代价方面要优于原始的 NEMO 协议。

关键词: NEMO 协议;改进的 NEMO 协议;搭便车现象;P2P 覆盖网络

中图分类号: TP393

文献标识码: A

文章编号: 1673-629X(2010)07-0098-04

An Enhanced NEMO Protocol

ZHANG Qing, YU Ji-guo

(Department of Computer, Qufu Normal University, Rizhao 276826, China)

Abstract: NEMO protocol doesn't make use of the nodes at the bottom layer of hierarchy, and doesn't consider node heterogeneity when selecting cluster leaders. For these problems, an enhanced NEMO protocol has been proposed in this paper. The enhanced NEMO protocol makes full use of nodes at the bottom layer of hierarchy by clustering technology, so it can avoid free riding in P2P overlay. In addition, it considers node heterogeneity when selecting cluster leaders. Each cluster leader selects some peers according to their abilities in its cluster to be its co-leader. When the cluster leader fails, the co-leader who has the largest ability is selected to be the new cluster leader. In this way, can improve overlay network's resilience. The enhanced NEMO protocol is better than NEMO protocol in some aspects by analysis.

Key words: NEMO protocol; enhanced NEMO protocol; free riding; P2P overlay

0 引 言

组播是一种允许一台主机一次同时发送单一数据分组到多台主机的技术。文献[1]中提出了 IP 组播,在 IP 组播体系结构中,组播功能是在路由器中实现的,网络中的路由器利用分布式算法定义了一棵数据转发树,当组播包沿着数据转发树转发时,它们在树的不同分支点处,由路由器复制。IP 组播是转发数据的最有效的方式,它能够将全网内的重复数据包降到最低。然而,IP 组播存在很多问题^[2~5]。随着 P2P 技术的发展,一些研究者开始研究 IP 组播是否是实现组播

的最有效的方式,并且提出了应用层组播。在应用层组播中,组播功能是在端系统处实现的。近年来,提出了很多应用层组播协议^[2,3,5~11],应用层组播协议的目标是建立并保持一个有效的覆盖网。但是,在应用层组播中,节点不断地加入与离开,为了提高覆盖网的容错性,已有学者提出了一些容错技术^[12~15]。

1 相关知识

NICE^[8]是一种可扩展的应用层组播协议,它将参加组播的节点分成簇,并且当组播源节点不同时,数据转发树也不同。与 Narada^[5]相比,当组播成员超过 32 个时,NICE 有更低的链路强度,更好或类似的端到端的延迟,类似的错误恢复能力,同时 NICE 有较低的控制负载。

NICE 通过将组播成员分配到不同的层次来创建层次结构。层次从最底层开始顺序编号,最底层表示为 L0。在最底层上,基于节点的端到端延迟,所有参加组播的节点都被组织成多个簇。每一个簇中节点的

收稿日期:2009-11-08;修回日期:2010-02-17

基金项目:国家自然科学基金(10471078);山东省中青年科学家奖励基金(2005BS01016);山东省科技攻关计划(2009GG10001014);山东省教育厅科研项目(J07WH05)

作者简介:张 晴(1983-),女,硕士研究生,研究方向为对等计算和计算机网络与通信;禹继国,博士,教授,硕士研究生导师,研究方向为对等计算、无线网络、优化理论与算法设计等。

物理位置都比较近,簇的大小在 d 和 $3d$ 之间, d 是一个配置常数。每一个簇中都有一个簇首, NICE 选择簇的中心节点作为簇首。每一个簇中的簇首都变成上一层的成员,然后,基于端到端的延迟再次分簇。重复这一过程,直到所有层中的节点都被分配到相应的簇中,并且选出新的簇首,然后新的簇首上升到更高一层中。当最高层只有一个节点时,就停止这一过程。因此,这个逻辑层次的每一层中都至少有一个簇。NICE 的层次结构如图 1 所示。

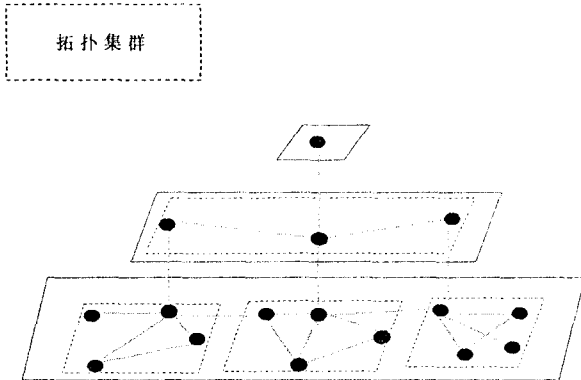


图 1 NICE 的层次结构

NEMO 协议^[14]是一种以性能为中心的应用层组播协议,它的设计思想采用了 NICE 协议^[8]的层次结构。但是, NEMO 协议引入了三个简单的技术,从而增强了其容错性。第一个是 $co-leaders$, 它降低了簇首的转发负载, 并且在簇首失效时能代替簇首, 从而避免了对单个节点的依赖, 所以在 NEMO 协议中, 节点被分为三种类型: 簇首, 辅助领导节点和普通节点; 第二个是用 NACKS (triggered negative acknowledgment) 来检测丢失的数据包并通过重传机制重传丢失的数据包, 从而提高了传输效率; 第三个是周期性随机算法 (periodic probabilistic operation)。NEMO 采用了周期性随机算法来修复网络, 在很大程度上降低了网络开销。但是, 在 NEMO 协议中, 存在着搭便车现象, 并且在选择簇首时也没有考虑节点的异构性。

2 改进的 NEMO 协议

改进的 NEMO 协议基于 NEMO 协议的层次结构, 但是与 NEMO 协议有两方面的不同: 首先, 选择簇首的方法与 NEMO 协议不同。在改进的 NEMO 协议中, 从一个簇中选择能力最强的节点作为簇首。其次, 数据转发的过程不同于 NEMO 协议。在改进的 NEMO 协议中, 当转发信息时, 簇首充分利用了最底层上每一个簇中的普通节点。

2.1 簇首的选择

改进的 NEMO 协议中, 从簇中选择能力最大的节

点作为一个簇的簇首, 这种方法利用了覆盖网络中的节点异构性。选择能力最大的节点作为簇首的原因是, 当组播信息或者一个节点加入时, 簇首可能会担负很大的责任或被多次访问。每一个簇首根据剩余节点的能力选择一些节点作为 $co-leader$ (辅助领导节点), 这些 $co-leaders$ 和簇首在簇中形成一个 $crew$ 。当簇首失效时, 就从这些辅助领导节点中选择能力最大的节点作为新的簇首。通过这种方法, 可以提高覆盖网络的容错性, 并且与原始 NEMO 协议相比, 簇首和辅助领导节点失效的可能性降低。

2.2 数据转发

在原始的 NEMO 协议中, 领导节点 (簇首和辅助领导节点) 将收到的信息转发给每一层中它所在簇中的每一个节点, 并且转发给上一层的节点。这种方法的缺点是, 当网络规模很大时, 领导节点很容易超负荷, 从而导致信息丢失, 因此, 传输效率降低并且开销增大。改进的 NEMO 协议采用了在最底层的每一个簇中分簇的技术^[16], 并且在信息转发过程中, 利用了最底层的普通节点, 使得簇首和辅助领导节点的负担大大降低, 信息转发的过程如下。

组播源节点首先将信息发给与它在同一个簇中的某个领导节点 (簇首或辅助领导节点)。假设, 在第 i 层上, 簇首 n 收到从它所在的簇中的组播源节点处发来的一条信息。根据 NEMO 协议的逻辑结构, 如果节点 n 出现在第 i 层上, 那么, 它一定是从第 0 层到第 $i-1$ 层上它所在簇的簇首。 n 根据分簇技术, 首先将信息发给第 0 层和第 i 层上与它在同一个簇中的所有节点。分簇技术的思想是: 节点 n 在相应的层中, 根据自己的能力从自己所在的簇中, 选择 C_n 个节点并且将信息转发给这 C_n 个节点, 同时, n 将簇中剩余的节点组织成 C_n 个小簇, 并且让这 C_n 个节点分别成为这 C_n 个小簇的簇首。在每一个小簇中, 这 C_n 个簇首重复分簇技术, 直到第 0 层和第 i 层上, n 所在簇中的所有节点都收到信息为止。

到此为止, 节点 n 所在的第 0 层上的全部节点和第 i 层上的全部节点都已经收到信息。对第 1 层到第 $i-1$ 层上 n 所在簇中的全部节点, 节点 n 并不亲自转发信息, 而是从第 0 层与它在同一个簇的节点中, 选出 $i-1$ 个节点并委派这些节点分别负责这 $i-1$ 个簇中的信息转发。比如说, 选择了一个节点 m , 让 m 将信息转发给第 1 层上、与节点 n 处于同一个簇中的所有节点。这时, 节点 m 利用分簇技术, 按照节点 n 在第 0 层上转发信息的过程来转发信息, 直到所有的节点收到信息为止。根据 NEMO 协议的层次结构, 越到高层, 簇中的节点数目就会越少, 所以, 这 $i-1$ 个节点按照能力从

大到小的顺序分别负责第 1 层到 $i-1$ 层上 n 所在簇中的全部节点。

到此为止,节点 n 所在的第 0 层到第 i 层的全部节点已经收到了信息。因为节点 n 在第 i 层的某个簇中是簇首,所以它一定出现在第 $i+1$ 层的某个簇中。这样又可以分两种情况来讨论:

(1) 如果在第 $i+1$ 层上的某个簇中,节点 n 不是簇首,那么它将信息转发给第 0 层上和第 i 层上它所在簇的全部节点时,它还将信息转发给第 $i+1$ 层上与它在同一个簇中的领导节点。这个领导节点在组播信息时,重复与节点 n 相同的过程。

(2) 如果在第 $i+1$ 层上的某个簇中,节点 n 是簇首,那么组播信息时,它还是委派第 0 层上它所在簇的某个节点来代替它转发信息,直到第 $i+1$ 层上它所在簇的全部节点收到信息为止。

到此为止,在这个层次结构中, n 所在簇的全部节点都已经收到了信息。

在上面转发信息的过程中,假设的是组播源节点出现在第 i 层上,并且 i 大于 0,所以当组播源节点将信息转发给第 i 层上它所在簇的领导节点(包括簇首和辅助节点)时,还应根据分簇技术,将信息转发给第 0 层上它所在簇的全部节点。但是,当组播源节点是普通节点时,也就是说,它只在第 0 层上出现,那么,它只将信息转发给它的领导节点(包括簇首和辅助节点)中的一个节点,接收到信息的领导节点采用上述的方法来转发信息。同理,当组播源节点处于层次结构的最高层时,它转发信息的过程也采用上面叙述的方法。

2.3 算法描述

算法 1 描述了在每一个簇中组播信息的过程, C_x 表示节点 x 能将信息发给多少个节点。

算法 1:

Per-Cluster(x, C_x):

If 节点 x 负责簇 C 中的信息转发。

Then 节点 x 根据自己的能力从簇 C 中选择 C_x 个节点,并且让 R_x 表示这 C_x 个节点的集合, x 将信息转发给 R_x ,同时将剩余的节点组织成个更小的簇,并且委派这个 C_x 节点分别负责这 C_x 个小簇。

For 集合 R_x 中的每一个节点 v

v 调用 Per-Cluster(x, C_x);

直到簇 C 中的所有节点都收到信息为止。

算法 2 描述了当超节点 SL 从第 i 层上的一个节点收到信息时,它转发信息的过程。

算法 2:

Super-leader(msg)

If ($0 \leq i \leq n$)

{

For 第 0 层和第 i 层上,SL 所在的每一个簇 SL 调用 Per-Cluster(x, C_x);

For ($k = i, k < n, k++$)

{

If 在第 $k+1$ 层上,SL 不是它所在簇的簇首

Then 在第 $k+1$ 层上,它只将信息转发给簇的领导节点中的一个,接收到信息的领导节点调用 Super-leader(msg)

Break for

Else SL 在第 0 层上从它所在的簇中选择一个节点 m ,并且委派 m 负责第 $k+1$ 层上节点 SL 所在的簇,节点 m 调用 Per-Cluster(x, C_x)

}

If $i > 1$

Then 对于从第 1 层到第 $i-1$ 层上 SL 所在的每一个簇,SL 从第 0 层上它所在的簇中选择 $i-1$ 个节点,并且委派这 $i-1$ 个节点来负责从第 1 层到第 $i-1$ 层上它所在的簇,这 $i-1$ 个节点在它们各自的簇中调用 Per-Cluster(x, C_x)

}

算法 3 描述了组播源节点 S 组播信息的过程。

算法 3:

Forward-data(msg)

If S 位于第 n 层

Then S 在转发信息的过程中直接调用 Super-leader(msg)

Else

If S 是第 i 层上它所在簇的领导节点($0 \leq i \leq n$)

Then S 在转发信息的过程中调用 Super-leader(msg)

Else S 在第 i 层上将信息转发给它的簇首 n ,节点 n 调用 Super-leader(msg)

3 性能分析

3.1 平均传递延迟

图 2 显示了网络规模(overlay network size)和平均传输延迟(the average latency)之间的关系。随着网络规模的增大,两者的平均传输延迟都增大,但是改进的 NEMO 协议的传输延迟比 NEMO 协议的要小。这是因为在改进的 NEMO 协议中,选择能力最大的节点作为一个簇的簇首。虽然,簇首不是簇的中心节点,并且从簇首到它孩子节点的延迟可能会增大,但是,选择能力最大的节点作为簇首能够降低从源节点到其他节点的延迟路径,所以这可以优化延迟。而且,改进的

NEMO 协议利用了最底层的普通节点,在组播的过程中,这些节点能够辅助领导节点,这样可以降低领导节点(簇首和辅助领导节点)的转发负担,从而提高了传输效率,降低了传输延迟,当网络规模很大时,效果会更明显。

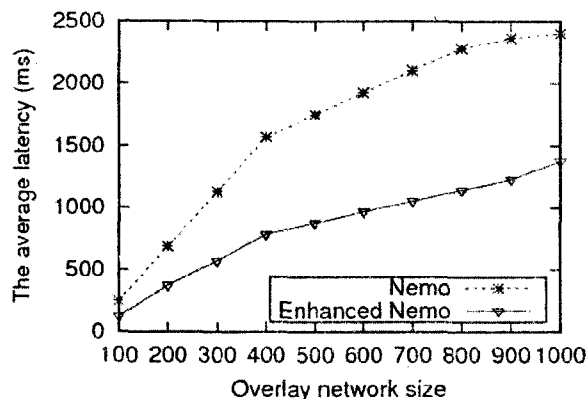


图2 网络规模与平均传输延迟

3.2 节点失效时的代价

图3显示了仿真时间(simulation time)与节点累积失效(the recovery cost of cumulative failure)时的恢复代价之间的关系。在节点失效时,改进的 NEMO 协议累积失效恢复的代价比 NEMO 的要小。这是因为在改进的 NEMO 协议中,根据节点的稳定性、容量和在线时间的长短来选择簇首,因此节点失效的可能性降低了,所以节点失效后恢复的代价降低,系统的稳定性也提高了。

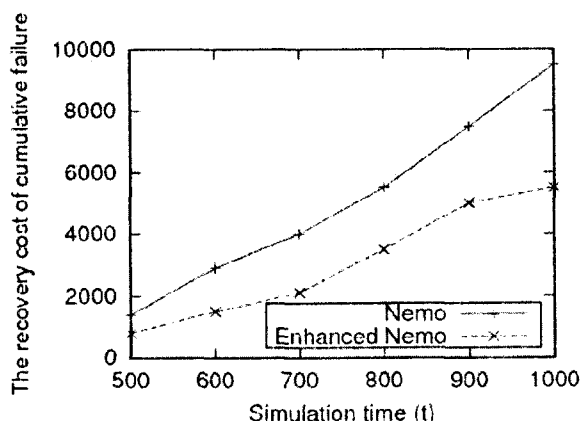


图3 仿真时间与节点累积失效恢复代价

4 结束语

文中没有提出新的应用层组播协议,而是对现有的 NEMO 协议进行了改进。在改进的 NEMO 协议中,在选择簇首时考虑到了节点的异构性,这样可以降低节点的失效代价,并且还利用了最底层的普通节点,这样使得它们对系统贡献了自己的资源,从而避免了 P2P 系统中的搭便车现象。最后,将改进的 NEMO 协

议与原始的 NEMO 协议进行了对比。仿真实验表明,改进的 NEMO 协议在某些方面比原始的 NEMO 协议更具优势。在以后的工作中,将从其他方面,对改进的 NEMO 协议进行评估,并且将它与其他应用层组播协议进行对比。

参考文献:

- [1] Deering S, Cheriton D. Multicast Routing in Datagram Inter-networks and Extended LANs[J]. In ACM Transactions on Computer Systems, 1990, 8: 85-110.
- [2] Francis P. Yoid: Extending the Internet Multicast Architecture[EB/OL]. 2000-04-02. <http://www.icir.org/yoid>.
- [3] Pendarakis D, Shi S, Verma D, et al. ALMI: An Application Level Multicast Infrastructure[C]// In Proceedings of 3rd USENIX Symposium on Internet Technologies and System. San Francisco, California, USA: [s. n.], 2001.
- [4] El-sayed A, Roca V, Mathy L. A Survey of Proposals for an Alternative Group Communication Service[J]. IEEE Network, 2003, 17(1): 46-51.
- [5] Chu Y H, Rao S G, Zhang H. A Case for End System Multicast[C]// In Proceedings of ACM SIGMETRICS. New York, NY, USA: ACM press, 2000: 1-12.
- [6] Saltzer J, Reed D, Clark D. End-to-End Arguments in System Design[J]. ACM Transactions on Computer Systems, 1984, 2(4): 195-206.
- [7] Jain S, Mahajan R, Wetherall D, et al. Scalable Self-Organizing Overlays[R]. Washington: Computer Science and Engineering University of Washington, 2002.
- [8] Banerjee S, Bhattacharjee B, Kommareddy C. Scalable Application Layer Multicast[C]// In proceedings of ACM SIGCOMM. New York, NY, USA: ACM press, 2002: 205-217.
- [9] Ratnasamy S, Handley M, Karp R, et al. Application-level Multicast using Content-Addressable Networks[C]// In Proceedings of 3rd International Workshop on Networked Group Communication. Berlin/Heidelberg: Springer, 2001: 14-29.
- [10] Castro M, Druschel P, Kermarrec A M, et al. SCRIBE: A Large-scale and decentralized application-level multicast infrastructure[J]. IEEE JSAC, 2002, 20(8): 1489-1499.
- [11] Zhuang S, Zhao B, Joseph A, et al. Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination[C]// In proc of NOSSDAV. New York, NY, USA: ACM press, 2001: 11-20.
- [12] Banerjee S, Lee S, Bhattacharjee B, et al. Resilient multicast using overlays[C]// In Proceedings of ACM SIGMETRICS. New York, NY, USA: ACM press, 2003: 102-113.
- [13] Castro M, Druschel P, Kermarrec A M, et al. Splitstream: High-bandwidth multicast in cooperative environments[C]//

互联网上的网页总数。

可以以一次实验来说明,假设用 Google 搜索词语“horse”返回 46 700 000(记为 $f(x)$) 条结果,搜索词语“rider”返回结果数为 12 200 000(记为 $f(y)$),搜索同时含“horse,rider”的网页数是 2 630 000(记为 $f(x,y)$),Google 共引用的网页数是 $N=8\,058\,044\,651$,代入上述公式(7)求得:

$$NGD(horse,rider)\approx 0.443$$

4 两类主要语义相似度计算方法的比较

下面对基于语义词典和基于语料库的词语相似度计算这两类策略的方法、前提条件、所用工具等 6 个方面进行比较,见表 2。

表 2 两类主要语义相似度计算方法比较

基于语义词典的词语相似度计算		基于语料库的词语相似度计算
方法	客观计算	经验法
前提条件	两个词汇具有一定的语义相关性,当且仅当它们在概念间的结构中有且仅有一条路径	词语的上下文可以为词语定义提供足够信息,两个词语语义相似当且仅当它们处于相似的上下文环境中
所用工具	语义词典	大规模语料库
理论依据	树论,图论	向量空间
优点比较	直观而且简单有效,可以计算出字面上不相似的词汇间的相似度	能够客观地反映词语的形态、句法、语义等特点
缺点比较	受人的主观影响比较大,有时不能反映客观现实性能	依赖于语料库的优劣,存在数据稀疏的问题,也有噪声干扰

5 结束语

鉴于语义相似度在现代科学领域中的广泛应用,在该文中,比较系统介绍了当前语义相似度计算的一些理论及方法,并简单比较了两种主要方法的特点及区别,重点描述了基于中文语义词典《知网 Hownet》的相似度计算方法;最后简单介绍了国外基于搜索引擎的相似度算法。除了完善语义词典的全面性和准确性之外,选择或找到一种相对比较简捷地准确计算出语义相似度的方法,以确定出相似度,然后将此方法应

用于信息检索等领域,改进当前仅仅依靠寻找匹配词查询所需信息的局限性。此外,在研究文本的相似性问题时,可以通过计算词与词之间、句与句的相似度得到整个文本的相似度,当相似值达到所设定的标准值时,即可认定所检验的两篇文章有抄袭其中之一的嫌疑。

关于文本相似性的研究对于当前的考试作弊雷同卷、论文抄袭等的鉴定性工作方面起到很大的改进作用,同时节省大量的人力和物力。

参考文献:

[1] Lin D. An Information Theoretic Definition of Similarity Semantic Distance in WordNet [C]//Proceedings of the Fifteenth International Conference on Machine Learning. [s.l.]: [s.n.],1998.

[2] Cilibrasi R L,Vita'nyi P M B. The Google Similarity Distance [J]. IEEE Transactions on Knowledge and Data Engineering, 2007,19(3):370-383.

[3] 董振东. 语义关系的表达和知识系统的建造[J]. 语言文字应用,1998(3):79-85.

[4] 刘群,李素建. 基于《知网》的词汇语义相似度的计算 [C]//第三届汉语词汇语义学研讨会. 台北:出版者不详,2002.

[5] 李峰,李芳. 中文词语语义相似度计算——基于《知网》2000[J]. 中文信息学报,2007(3):99-105.

[6] 李鹏,陶兰,王弼佐. 一种改进的本体语义相似度计算及其应用[J]. 计算机工程与设计,2007(1):227-229.

[7] 李熙,徐德智. 基于 WordNet 的概念语义相似度研究 [J]. 湖南科技学院学报,2008,29(12):115-116.

[8] 杨哲. 基于启发式规则的本体概念语义相似度匹配[J]. 计算机应用,2007,27(12):2919-2921.

[9] 夏天. 汉语词语语义相似度计算研究[J]. 计算机工程,2007(6):191-194.

[10] 张明宝,马静. 一种基于知网的中文词义消歧算法[J]. 计算机技术与发展,2009,19(2):9-11.

[11] Doan A,Madhavan J. Learning to Match Ontologies on the Semantic Web[J]. The VLDB Journal, 2003,12(4):116-120.

(上接第 101 页)

In proceedings of the 19th ACM SOSP. New York, NY, USA: ACM press,2003:298-313.

[14] Birrer S,Bustamante F E. Resilient peer-to-peer multicast without the cost [C]//In Proceedings of MMCN. Berkeley: University of California,2005:113-120.

[15] Birrer S,Bustamante F E. Magellan: Performance-based, co-

operative multicast [C]//In Proceedings of IWCW. Evanston, IL, USA:[s.n.],2005:133-143.

[16] Li Zhenyu,Zhu Zengyang,Xie Gaogang,et al. Fast and proximity-aware multi-source overlay multicast under heterogeneous environment [C]//In computer communications. [s.l.]:[s.n.],2008:257-267.