ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.4, April 2008, pp.925−945
DOI: 10.3724/SP.J.1001.2008.00925

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

基于双向令牌的可扩展及可靠的群组成员管理*

王国军+, 吴 敏, 周 薇, 贺建飚, 陈松乔

(中南大学 信息科学与工程学院,湖南 长沙 410083)

Scalable and Reliable Group Membership Management with Bi-Directional Tokens

WANG Guo-Jun+, WU Min, ZHOU Wei, HE Jian-Biao, CHEN Song-Qiao

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

+ Corresponding author: Phn: +86-731-8877711, Fax: +86-731-8877711, E-mail: csgjwang@mail.csu.edu.cn

Wang GJ, Wu M, Zhou W, He JB, Chen SQ. Scalable and reliable group membership management with bi-directional tokens. *Journal of Software*, 2008,19(4):925−945. http://www.jos.org.cn/1000-9825/19/925.htm

Abstract: Group communications have been studied in the wired Internet for many years and remain a very hot research topic, especially on extending the existing achievements into mobile and wireless network environment. This paper identifies some interesting problems in mobile group communications regarding dynamic group membership due to member joining and leaving, dynamic locations due to node mobility, and dynamic networks due to node/link failures. These problems are solved by proposing a ring-based hierarchy of proxies with bi-directional tokens. The proposed hierarchy is a combination of logical rings and logical trees, which takes advantage of the simplicity of logical rings and the scalability of logical trees. More importantly, such a combination makes the proposed protocol based on this hierarchy more reliable than the existing tree-based protocols. Theoretical analysis and simulation studies of the proposed protocol show that: (1) It scales very well when the size of a group becomes large; (2) It is strongly resilient to failures that occur in the network. It is particularly suitable for those service providers and network operators which have deployed their machines in a hierarchical setting, where each machine can be locally configured to know the information about its sibling and parent machines.

Key words: group communications; group membership management; multicast; reliability; bi-directional tokens

摘 要: 面向有线因特网的群组通信已研究多年,目前仍是研究热点之一,尤其是将现有研究成果扩展到移动与无线网络环境方面.研究了移动群组通信,该问题涉及群组成员关系动态性(成员加入及退出)、成员位置动态性(移动主机的移动性)和网络动态性(结点或链路出错).提出了适合于移动群组通信的基于双向令牌的层次环模型(也称为层次环结构)以解决该问题.该模型是逻辑环与逻辑树的结合模型,它利用了逻辑环的简单性和逻辑树的可扩展性.更为重要的是,这样的结合使得基于层次环结构的群组通信协议比现有的基于树结构的协议更可靠.理论分析和

* Supported by the National Natural Science Foundation of China under Grant Nos.60503007, 90718034 (国家自然科学基金); the National Science Fund for Distinguished Young Scholars of China under Grant No.60425310 (国家杰出青年科学基金); the Program for New Century Excellent Talents in University of Ministry of Education of China under Grant No.NCET-06-0686 (新世纪优秀人才支持计划); the Program for Changjiang Scholars and Innovative Research Team in University of China under Grant No.IRT-0661 (国家教育部长江学者与创新团队发展计划)

Received 2006-08-03; Accepted 2007-05-24

模拟研究表明:(1) 当群组规模增大时,该协议的可扩展性很好;(2) 该协议具有很高的可靠性.该协议特别适合于服务提供者和网络运营商将其计算设备分层次部署的情况,这时就要求每台计算设备都能局部化地维护其兄弟和父亲设备的信息.

关键词: 群组通信;群组成员管理;组播;可靠性;双向令牌

中图法分类号: TP393 **文献标识码**: A

近年来,随着移动计算与无线通信技术的飞速发展,由有线因特网和多种无线网络如无线局域网、蜂窝网络和卫星网络组成异构型的网络,即通常所说的移动因特网,成为了研究热点[1].但是在这种网络环境中,许多具有挑战性的问题将比在有线网络中更难以解决,如可扩展性和可靠性.其中一个典型应用是将数据从一个或多个发送者同时传送给多个接收者的群组通信应用,比如面向新闻信息、股市行情、天气预报、交通状况的数据广播应用和基于网络的多媒体流式应用.但是研究表明,在移动因特网中实现高效的群组通信非常困难[2−4].现有的群组通信系统(group communication system,简称GCS)[5,6]主要是为传统的广域网而设计的,没有考虑把移动主机(mobile host,简称MH),如笔记本电脑、个人数字助理(personal digital assistant,简称PDA)、移动电话和移动可视电话当作群组成员的情况,因此无法保证在MH存在时也能高效工作.实际上,广域网中固有的问题,如延迟大和不稳定性,仍然存在于移动因特网中.而且,在引入MH以后可能出现频繁断开连接、频繁切换、频繁发生错误等更为困难的问题.

本文基于我们在文献[7]中的前期工作,提出了适合于移动群组通信的基于双向令牌的层次环模型(也称为层次环结构).一方面,该模型是逻辑环与逻辑树的结合模型,它利用了逻辑环的简单性和逻辑树的可扩展性;更为重要的是,这样的结合使得基于层次环结构的群组成员管理协议比现有的基于树结构的协议更为可靠.当树结构中某结点出错时,树结构会被打破成多棵子树,所有子树的根结点必须找到新的父结点重新附着到树结构上.因此,树结构存在单点错误问题(single point of failure problem).当在层次环结构中出现错误结点时,该结构也会被打破成为多个子结构.但是,由于该结构中的每个代理维护两类候选邻居(候选的兄弟结点与候选的父结点),该结构能够尽快地维护它的完整性,从而可以解决树结构中的单点错误问题.

另一方面,该模型结合了应用层组播和网络层组播的解决方案,既利用了网络层组播的高效性,又利用了应用层组播的易于部署的特点.从系统实现的观点看,现有两种基本的因特网组播通信技术:基于网络层的组播(以下简称IP组播[7])和基于应用层的组播(以下简称应用层组播[8]).IP组播提出得早,组播效率高,但应用不广泛.其主要原因是:(1) 部署时需要升级所涉及的网络中所有的路由器,因而是对现有网络基础设施的一种“巨变”;(2) 只提供局部的群组成员管理[9],对某些需要全局成员信息的应用则不适合;(3) 由于组播树的维护是基于一种平面的网络结构,因而难以扩展到大规模的组播通信中.针对第 3 点,文献[10]提出了按需分支组播的方法.该方法只要求处于组播树分枝处的路由器和本地链路上有组成员的路由器保存组播树的有关信息,将基于平面的组播树维护方式转变为层次型结构的维护方式,从而提高了可扩展性.为了从IP组播的困境中走出来,近几年提出了大量的应用层组播方案.由于应用层组播只要求网络提供单播服务,其他工作如组播树的维护、组播包的复制都在称为“叠加网”中的主机而不是在路由器中完成,因此部署比较容易,但是其效率可能不如IP组播高,且难以扩展到大规模网络环境中.本文所提出的层次环模型中的结点根据实际需要可以是传统的IP组播中使用的路由器,也可以是新型的应用层组播中部署的主机.

本文第 1 节介绍相关工作.第 2 节概述所提出的成员管理协议.第 3 节和第 4 节分别描述所提出的成员管理协议中的两个子协议:成员信息传播子协议和拓扑结构维护子协议.第 5 节分析层次环模型的可扩展性及可靠性.第 6 节通过大量模拟实验给出协议的性能评价.第 7 节总结全文.

1 相关工作

群组通信系统提供面向一组进程的通信服务.群组是指该组进程,这些进程被称为该群组的成员.进程可以自愿加入或退出某群组,或者因为出错而离开某群组.群组的成员关系信息是指该群组中当前活跃的进程/成员

列表,该列表通常称为“视图”.GCS 能够有效地简化容错分布式应用的编程难度.GCS 的一个重要任务是群组成员管理服务,即当成员加入、退出、切换或出错事件发生时维护成员关系信息;GCS 的另一个重要任务是组播通信服务,即将信息从一个或多个源结点高效地传送到群组中的所有活跃进程.

群组成员管理算法主要分为 3 大类:(1) 基于广播的算法[11];(2) 基于协调者的算法[12];(3) 基于令牌的算法[13].基于令牌算法的基本思想是,把群组成员组织成一个逻辑环,然后让令牌环绕逻辑环以传播成员消息.但是,基于逻辑环的算法无法很好地扩展到大规模群组通信环境中.为了提高成员管理算法的可扩展性,研究者提出许多可扩展的成员管理模式.成员角色模式[14]区分核心成员、客户成员和汇点成员.Spread系统[15]集成了两个层次的协议:第 1 层是局域网中的Ring协议,第 2 层是广域网中的Hop协议.

一些基于多层的层次结构模式提供了更好的可扩展性方案.Transis系统[11]把广域网看成组播簇结构,每个簇表示可以通过广播或组播硬件进行通信的计算机域.通过将所有簇中挑选的代表结点形成层次结构来组织这些簇.结构中的每一层可以看作是它以下各层的抽象,并维护它所在层的群组服务.该系统使用抽象方法使得群组服务在广域网中具有好的可扩展性.但是,由于高层域的代表结点实际上是最底层域中的服务器,这加重了服务器处理不同层次群组服务的负担.特别地,如果代表结点出错,则会影响到该代表结点涉及的所有群组域.

CONGRESS系统[16]把广域网看作由域组成的层次结构.每个域由一个CONGRESS服务器提供服务,包括局部成员服务器(local membership server,简称LMS)和全局成员服务器(global membership server,简称GMS),以提供弱一致性的成员信息.LMS设置在每个主机上为其客户提供服务.GMS设置在树结构中,高层GMS实际上就是底层的LMS.该结构被称为基于代表结点的树层次结构.CONGRESS系统的新颖之处在于它的客户-服务器方法:群组成员服务由指定的成员服务器提供,但它们本身不是任何群组的成员.作为成员参与某个或某些群组的进程成为成员服务器的客户.客户发送请求给它的服务器以加入或退出特定的群组,而成员服务器发送成员视图给它的客户.这种方式使得该协议在群组个数和每个群组中的成员个数上的可扩展性都非常好.Moshe系统[17]扩展了CONGRESS系统,提供了强的成员信息语义和强的消息分发语义.CONGRESS和Moshe系统的新意在于,它们以客户-服务器的方式建立:群组成员服务由指定的成员服务器提供,但服务器本身并不是任何群组的成员.

为了提高系统的可靠性,研究者提出许多错误检测方法.考虑到许多应用有某种时间约束,文献[18]为错误检测器设计了一个新的规范:(1) 及时性,即错误检测器需要多久才能察觉到出错;(2) 准确性,即错误检测器检测结果正确的概率有多大.文献[18]提出的Freshness points方法同时满足这两个规范.其基本思想是,如果进程q需要检测进程p是否出错,则需p每隔η个时间单位发送心跳(heartbeat)消息$m_1,m_2,\ldots$给q.为了决定是否怀疑p,q采用固定的时间点序列$\tau_1,\tau_2,\ldots$,这些点被称为Freshness points,它们是由固定参数δ改变心跳消息的发送时间而获得的.更具体地说,$\tau_i=\sigma_i+\delta$.这里,σ_i是消息m_i的发送时间.对任意给定的时刻t,假设i使得$t\in[\tau_i,\tau_{i+1})$,则当且仅当q已经收到消息m_i或更往后的消息时,q信任t时刻的p.

以上工作主要面向传统的有线广域网环境.而在移动因特网中,群组成员关系不仅受到进程状态(活跃或出错)和链路状态(连接或断开)的影响,还受到MH的移动性的影响.但是,针对这种网络环境的成员管理问题的研究还很少.在移动因特网中,移动IP技术[19−21]使得MH在改变网络接入点时不必改变IP地址而保持通信的连续性.移动IP提供了面向组播移动性管理的两种基本方案:双向隧道和远程签署.双向隧道方案向外界屏蔽了MH的移动性,MH移动时,无须重构组播树,但是组播路径不是优化的;远程签署方案具有优化的组播传输路径,但是可能引起组播树的频繁重构.这两种基本方案代表了移动IP技术与IP组播技术结合时的两种极端情形,因而在一般情况下并不适用,其主要原因是没有充分考虑MH的移动性对组播通信性能的影响.

文献[22]提出两层的 Host-View 协议.Host-View 由一组移动支持站(mobile support station,简称 MSS)组成.MSS 代表群组中聚合的位置信息,即:只要该 MSS 范围内至少有一个 MH 仍是该群组中的成员,则该 MSS 代表这些成员加入该群组的 Host- View;如果所有成员都退出了该 MSS,则该 MSS 将退出 Host-View.该协议以 MSS 为基本的组播移动性管理单元,通过记录一组 MSS 而不是跟踪单个的 MH,使成员管理和组播通信变得非常简单.而且,为了把组播消息传送给由 MH 组成的群组,只需发送消息给该群组的 Host-View 中的 MSS 就可以

满足要求.由于大部分的任务将在 MSS 中完成,MH 的任务将大为减少.但是,Host-View 协议不允许群组成员动态加入和退出,也没有指定一种方法用于创建和删除群组.特别地,由于每次“显著性移动”必然导致全局更新,这不仅可能导致组播效率低,而且有可能引起长时间的服务中断.

文献[23]提出了一个 3 层的 RelM 协议以解决 Host-View 协议中的问题:最底层由 MH 组成,中间层由 MSS 组成,这些 MSS 组合成若干个 MSS 组,每个组由一个监督主机(supervisor host,简称 SH)控制,SH 组成了第 3 层.因为 SH 是有线网络的一部分,它可以处理大部分的协议细节,比如维持 MH 之间的连接以及为可靠组播通信收集确认消息.

文献[12]提出的 RMP(a reliable multicast protocol for distributed mobile systems)协议也是基于 3 层结构:一层是 MH,中间层是 MSS,第 3 层是协调者.在 RMP 中,每个 MSS 维护一个称为 local 的数据结构,用来标识它所在局部范围中 MH 的集合.RMP 采用的系统模型相当通用,它没有限制群组成员的移动模式,并且可适用于具有不完全的空间覆盖区域的无线网络环境.特别地,MH 从一个 MSS 切换到另一个 MSS 不会导致有线网络中的任何消息交换.

在以上相关工作中,基于令牌环绕逻辑环的算法虽然简单但可扩展性不好;而基于树型层次结构的算法可扩展性好但可靠性存在问题.本文的基本思路就是将两种结构结合起来,提出了层次环结构,使得基于该结构的成员管理算法既具有良好的可扩展性,又具有较高的可靠性.

2 协议概述

2.1 系统模型

研究者现已提出许多面向移动因特网的网络体系结构以解决移动因特网中的异构性问题,即不同类型的 MH通过不同类型的无线网络无缝地访问不同类型的应用.典型的体系结构包括无线叠加网络体系结构[24]、全 IP无线与移动网络体系结构[25]和总是最佳连接体系结构[26].基于这些体系结构,本文提出了基于多层代理的移动因特网体系结构,如图 1 所示.部署在有线因特网上的组播服务器称为全局发送者,而部署在无线网络中的组播服务器称为局部发送者.全局发送者为整个网络中的移动用户提供组播服务,而局部服务器为有限区域的用户提供服务.本文扩展了代理方法[24]和通信网关方法[27],在不同网络之间设置若干层次的代理,负责为服务提供商与移动用户隐藏异构性.本文区分两种类型的代理:(1) 直接代理(direct proxy,简称DP),如无线局域网的接入点、蜂窝网络的基站和卫星网络的卫星,它们直接为附着的MH提供服务;(2) 间接代理(intermediate proxy,简称IP),部署在组播发送者与DP之间,组播发送者与DP(及MH)之间的通信必须经由IP.

在图 1 中,不同的无线网络通过不同的方式连接到有线因特网上:无线接入网络(radio access network,简称 RAN)通过蜂窝网络的核心网络(core network,简称CN)连接到有线因特网[28];无线局域网既能通过网关(gateway)直接连接到有线因特网,也能先通过网关再通过蜂窝网络的CN间接连接到有线因特网[29];而卫星网络可以通过固定地面站(fixed earth station,简称FES)与有线因特网连接[25].在图 1 中,代理可以是网络中独立的主机,也可以依附在一些网络实体上,例如无线局域网的网关、蜂窝网络的无线网络控制器(radio network controller,简称RNC)、GPRS服务支持结点(serving GPRS support node,简称SGSN)、GPRS网关支持结点(gateway GPRS support node,简称GGSN)[28]、卫星网络的FES,甚至是有线因特网中的边界路由器.

基于该结构提出了基于代理结点的层次环模型.图 2 是一个 4 层的例子,自顶向下分别是间接代理层 2(intermediate proxy tier 2,简称 IPT2)、间接代理层 1(intermediate proxy tier 1,简称 IPT1)、直接代理层(direct proxy tier,简称 DPT)和移动主机层(mobile host tier,简称 MHT),上面 3 层根据位置相邻性(或其他特性)分别组织成一个或多个逻辑环,每个逻辑环中有且仅有一个领导结点.领导结点除了起到普通代理的作用,还负责与层次结构中上一层的父结点通信(如果父结点存在).领导结点是其父结点的孩子结点,父结点与该领导结点之间的关系称为父子关系.另外,每个逻辑环中邻居的关系称为前后关系.逻辑环中每个代理都存在该代理的前一个代理(简称前代理或前结点)以及该代理的后一个代理(简称后代理或后结点).如果逻辑环只包含一个结点,则领导结点、前结点、后结点都是这个结点本身.每个代理初始化时获取以下信息:(1) 若干个候选的兄弟代理,该代理

通过它们与同一层逻辑环合并;(2) 若干个候选的父代理,该代理通过它们附着到上一层逻辑环.

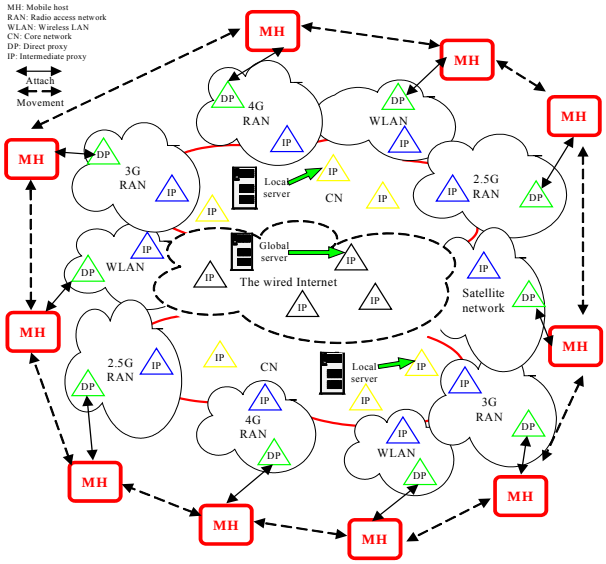


Fig.1 Multi-Tier proxy-based mobile Internet architecture

图 1 基于多层代理的移动因特网体系结构

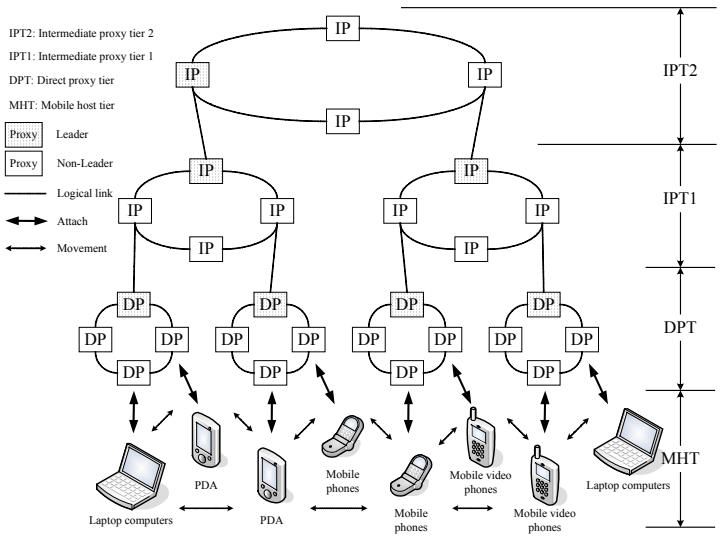


Fig.2 Ring-Based hierarchy of proxies for group communications

图 2 面向群组通信的基于代理结点的层次环模型

最后,本节简要介绍移动性检测机制.每一个 DP 周期性地在其覆盖区域中广播心跳消息.作为群组成员的 MH 一旦收到心跳消息,就返回一条包含其主机标识符的招呼(greeting)消息,对 DP 宣告其存在.DP 一旦收到招呼消息,就把 MH 登记为群组成员,即 MH 通过 DP 加入层次环结构.在此过程中,如果 DP 没有附着到该结构,则开始加入该结构.MH 加入该结构后,周期性地发送成员刷新(member-update)消息给 DP,以刷新其存在信息.如果在超时间隔过后 DP 仍没有收到刷新消息,则认为 MH 切换到另一个 DP 或者出错;如果 MH 从不同的 DP 收到若干个心跳消息,则简单地选择在它与 DP 之间距离最近的那个 DP,然后返回一条招呼消息.

2.2 局部群组的概念

本文把每个DP逻辑环范围内的群组成员看作是一个基本的局部群组,把每个IP逻辑环范围中的群组成员看作是一个扩充的局部群组.扩充的局部群组包含其子层次环结构中这些基本局部群组中存在的所有成员.

基于局部群组概念,这里以成员加入(member-join)消息为例说明成员改变消息是如何沿着层次环结构自底向上传送的.如果 MH 希望加入群组,它首先发送 Member-Join 消息给其附着的 DP,然后,该消息进入这个 DP 的消息队列中,接着同时执行以下两个操作:

- 成员信息传播操作:该操作运行在DP位于的局部群组或逻辑环中,以传播Member-Join消息.如果该消息到达局部群组或逻辑环的领导结点,则由领导结点发送到其父结点(如果父结点存在).当领导结点的父结点收到该消息时,该消息就进入父结点的消息队列中,然后,继续在父结点所在的局部群组或逻辑环中传送.该成员信息传播过程一直持续到 Member-Join 消息到达顶层的局部群组或逻辑环为止.
- 拓扑结构维护操作:如果 DP 已经在层次环结构中,则对于 Member-Join 消息不需要任何拓扑结构维护的操作;如果 DP 不在层次环结构中,则通过与其候选兄弟结点联系,尝试加入 DP 逻辑环.如果至少有一个 DP 逻辑环满足条件,如邻接性条件,则 DP 加入该 DP 逻辑环.如果联系过程失败,就建立一个 DP 逻辑环只包含这个 DP 本身,并设置自己为逻辑环的领导结点.该逻辑环将与其邻接的一个 DP 逻辑环合并,或与上一层的候选 IP 联系,然后附着到其中一个 IP.该加入过程一直持续到消息到达层次环结构的顶层局部群组或顶层逻辑环为止.

对于层次环结构中的成员消息传播而言,当把每个局部群组看作是一个结点时只使用单向传播:MH 请求 DP 更新成员信息;DP 领导结点再请求其父结点更新成员信息,等等.这样,MH 和其附着的 DP 之间就形成客户-服务器关系;DP 领导结点和其父结点也形成客户-服务器关系.客户-服务器方法可以有效地降低协议设计的复杂性,并提高协议的可扩展性,这是因为每个成员改变消息只需传送给其父结点,而无须广播到整个网络中.

2.3 邻居检测的概念

严格地说,只有已经加入群组的 MH 才是群组成员,层次环结构中所有代理都不是群组成员.由于成员信息传播操作沿着层次环结构自底向上传送成员信息,则每个局部群组将包含其子层次环结构中所有群组成员的弱一致性的成员关系信息.成员信息的准确性依赖于当层次环结构中出错时如何准确、及时地维护整个结构.

一方面,要求为系统中每个代理初始配置若干个候选邻居(候选父结点和候选兄弟结点),候选结点的配置可能随系统演化而发生变化.本文假设在任意时刻为每个代理所配置的候选结点总数都维持在一个较小的值上,如最多为 8.另一方面,层次环结构中每个代理在任意时刻至多有 4 个当前邻居(前结点、后结点、父结点和孩子结点).层次环结构中代理总数可能会很多,但是邻居个数(候选邻居和当前邻居)可以很少.

邻居检测的工作机理是:为每个代理配备一个邻居检测部件,负责监视代理的所有候选邻居和当前邻居的状态.为了检测当前邻居是否出错,可以直接使用基于心跳消息的Freshness points方法[18].然而,为了检测候选结点是否可达,本文扩展该方法以处理探测(polling)消息.如果进程q需要检测进程p是否可达,则q周期性地每隔η个时间单位发送探测消息$m_1,m_2,...$到p.当p收到探测消息后就把消息返回给q.决定是否怀疑p的余下过程类似于基于心跳消息的Freshness points方法.错误检测时,代理发送的每个心跳消息包含该代理的前代理和后代理的结点标识符的信息(若前代理和后代理存在).当目标代理收到该消息后就相应地更新信息.无论每个运行拓扑结构维护操作的代理当前是否处在层次环中,都使用基于探测消息的Freshness points方法以维护其候选邻居.如果目标代理是候选父结点,则探测消息带回目标代理是否有孩子结点的信息;如果是候选兄弟结点,则带回目标代理是否在其子层次环中"顶层"逻辑环的信息,即目标代理的领导结点当前是否有父结点,以及目标代理的前代理、后代理和领导代理的结点标识符的信息(若前代理、后代理和领导代理存在).

本文假设崩溃类型的结点错误可能发生,并称这种错误为代理错误.同时假设网络中链路错误只可能暂时地发生,而不会永久性地发生(除非链路对应的代理结点出错).崩溃结点不能从系统接收任何消息,也无法发送任何消息给系统,而暂时出错的链路将在相当短的时间内丢弃沿该链路传送的消息.本文基于邻居检测概念提

出了 Neighbor-Repair 算法(即基于当前邻居检测的层次环修复算法).如果发生网络分区,成员管理协议在每个分区中独立运行.当分区合并时,成员信息也合并在一起.基于可达性检测,本文提出了 Partition-Repair 算法(即基于候选邻居检测的网络分区的合并算法).本文将涉及网络分区与网络合并的错误称为网络错误.

虽然错误检测器很少出错,但它仍然有可能出错.下面我们来说明一旦邻居检测器出错时是否会对协议造成影响.首先,考虑进程 q 正在检测当前邻居 p.如果 p 确实出错,则无法发送和接收任何信息,因此,可以假设 p 总是被 q 正确地检测为出错.但是,即使 p 没有出错,在某些情况下也可能被错误地检测为出错.例如,从 p 发送到 q 的心跳所有消息全部丢失了.在这种情况下,为了确保协议正确运行,需要 q 丢弃从 p 给 q 的所有消息,并且 q 停止发送任何消息给 p.这样,q 认为 p 真的出错,p 也认为 q 出错,因为之后 p 无法从 q 接收任何信息.然后考虑进程 q 正在检测候选邻居 p 的情况.如果 p 是真的不可达,则可假设 q 总能正确地检测到 p 是不可达的.但是,q 可能错误地检测 p 是不可达的,而实际上是可达的.易知,这不会对本文的协议造成影响.这是因为 q 不会把其所有可达的候选邻居检测为不可达,否则就违背了 Freshness points 方法有相当高准确性的特征.

2.4 基本组播协议

使用层次环模型的组播信息传送是由组播发送者发送组播信息给“顶层”逻辑环中的代理.对于全局发送者,“顶层”逻辑环是整个结构的顶层逻辑环;而对于局部发送者,“顶层”逻辑环是其所位于的子结构的顶层逻辑环.然后,组播信息沿每个逻辑环传送再转发给所有孩子结点.最终,MH 从其附着的 DP 接收组播信息.

这里先介绍针对组播通信的组播移动性管理概念.对于组播通信中的位置管理,本文考虑把层次环结构视为整个组播群组的一个虚拟位置.换言之,该层次环结构动态地变化以反映当前群组中所有群组成员的聚合位置信息.对于组播通信中的切换管理,本文采用了预留机制.它与成员管理的预留机制有关,但不同之处在于,当 DP 接收到预留消息时,如果 DP 已经建立了自己与其领导代理的父结点 IP 之间的组播路径,则丢弃这个消息;如果没有建立组播路径,则立即建立一条通向其领导代理的父结点 IP 的预留路径.这样,如果 MH 切换到新的已经执行过预留操作的 DP,则该 MH 能够立即接收组播消息.

由于 MH 的成员关系动态性与位置动态性,层次环结构中低层次比高层次变化得更快.如果在 DP 逻辑环变化非常频繁时仍使用该结构,则组播传送就会变得很低效.因此,在组播传送协议中,本文使用一个与层次环结构稍有不同的层次结构:(1) 如果 DP 察觉到自己包含的成员个数从零变化到非零,则开始建立一条经由某些称为组播移动代理(multicast mobility agent,简称 MMA)的中间代理通向其领导代理的父结点 IP 的组播路径;(2) 如果 DP 已经从邻接 DP 收到预留消息,且不包含任何活跃成员,则启动预留一条经由某些 MMA 通向其领导代理的父结点 IP 的组播路径;(3) 如果 DP 退出该结构,则拆除通向其领导代理的父结点 IP 的组播路径.

3 基于局部群组的成员信息传播子协议

3.1 MH、代理、令牌的数据结构

MH 的数据结构.作为群组成员的每个 MH 记录以下信息:

- GID:GroupID.群组标识符,可以采用某种群组寻址模式,例如IP组播中的D类地址[7];
- DP:NodeID.附着的 DP 的结点标识符;
- GUID:GloballyUniqueID.MH的全局唯一标识符,例如移动IP网络的家乡地址[30];
- LUID:LocallyUniqueID.MH的局部唯一标识符,例如移动IP的转交地址[30];
- Status:Integer.为一整型值,典型状态如活跃、断开连接和出错.

代理的数据结构.每个代理记录以下信息:

- GID:GroupID.请见 MH 的数据结构;
- Current,Leader,Previous,Next,Parent,Child:NodeID.分别表示层次环结构中的当前结点、领导结点、前结点、后结点、父结点和孩子结点的结点标识符;
- PreviousOK,NextOK,ParentOK,ChildOK:Boolean.分别表示当前结点在层次环结构中的前结点、后结

点、父结点、孩子结点的状态.TRUE 表示没有出错,FALSE 表示出错或者不存在;

- CandidateSiblings[],CandidateParents[]:NodeID.分别表示候选兄弟结点和候选父结点的结点标识符列表;
- CandidateSiblingsOK[],CandidateParentsOK[]:Boolean.分别表示候选兄弟结点和候选父结点的状态列表.TRUE 表示可达,FALSE 表示不可达或者不存在;
- ListOfMembers[]:MemberInfo.以当前结点所属的逻辑环为根向下到其覆盖的所有 DP 及 MH 的子层次环中活跃成员的列表;
- MQ:MessageQueue.缓存成员改变/成员关系更新消息的消息队列.

令牌的数据结构.每个代理独立地收集、生成成员改变/成员关系更新消息,使用令牌沿着逻辑环以传送这些消息.令牌记录以下信息:

- GID:GroupID.请见 MH 的数据结构;
- Holder:NodeID.令牌持有者的结点标识符;
- MQ:MessageQueue.缓存成员改变/成员关系更新消息的消息队列.

3.2 成员信息传播算法

在层次环结构中,每个代理维护当前存在于其子层次环中的所有活跃群组成员的"全局"成员信息.群组成员管理包括两个主要任务:成员信息传播和拓扑结构维护.这两个任务是紧密相联的,但是,这里用在这两个任务之间通信的事件把它们分开以简化处理.

本文为成员信息传播设计两种信号消息(signaling message):Membership-Change(简称 MC)消息(即成员改变消息,该类消息与单个移动主机相关)和Membership-Update(简称MU)消息(即成员关系更新消息,该类消息包含了与多个移动主机相关的聚合成员信息).每个逻辑环中的领导结点负责根据其 ListOfMembers[]周期性地生成 MU 消息.MC 消息有以下几种:(1) Member-Join/Leave/Handoff 消息(当 MH 加入、退出或切换时生成);(2) Member-Update 消息(由每个作为成员的 MH 生成,周期性地向其附着的 DP 刷新其状态);(3) Member-Failure 消息(当 DP 周期性地检查其 ListOfMembers[]后发现某个 MH 已经闲置超过预定义的时间时发送该消息).有 3 种情况可能导致错误状态:(1) 作为群组成员的 MH 确实出错;(2) MH 发送的所有 Member-Update 消息全部丢失;(3) MH 从其附着的新 DP 发送给旧 DP 的 Member-Handoff 消息丢失.

在 DP 层中,MC 消息由每个逻辑环中的令牌沿着逻辑环传播.成员改变消息,例如 Member-Join,Member-Leave,Member-Failure,Member-Handoff,由环绕在每个逻辑环中的令牌沿着层次环结构自底向上传播.对于每个逻辑环,如果令牌成功环绕一轮后没有丢失,在使用某种重传机制的基础上,令牌的控制权可靠地转移到逻辑环中的下一个结点.令牌的控制权成功转移后,逻辑环中所有代理都已获得了相同的成员关系.算法 1 表示在每个 DP 逻辑环中运行的成员信息传播算法.

算法 1. One-Round 令牌传递的成员信息传播算法.

输入:令牌(token)所处的当前结点 CurNode 和 CurNode 所属的逻辑环;

输出:沿着逻辑环传播成员改变/成员关系更新消息.

```
1.  While TRUE Do {
2.    On Receiving a Token:
3.      If Token.Holder==CurNode.Current Then
4.        Let Token.MQ be empty;
5.      Elseif Token.MQ is not empty Then
6.        Update CurNode.ListOfMembers[] with Token.MQ;
7.      Elseif CurNode.MQ is not empty Then {
8.        Let Token.Holder be CurNode
9.        Let Token.MQ be CurNode.MQ;
```

```
10.   }
11.   Transfer the Token to the next node reliably;
12.  }
13.  //注释 1. 当令牌环绕一轮后,其 MQ 被设为空;
14.  //   然后空令牌沿着下一条逻辑链路被可靠地转移到一个 MQ 不为空的结点.
15.  //注释 2. 由于代理结点可能出错(见第 4 节),因此令牌可能会丢失.
16.  //   本文假设,如果领导结点在超时间隔以后没有收到令牌,则领导结点将重新生成一个新的令牌.
```

从某一层的领导结点到其上一层的父结点(如果父结点存在)的 MU 消息的传播,本文采用超时机制:首先每个领导结点周期性地发送其成员信息给其父结点,成员信息保存在其父结点的 MQ 中;然后,父结点使用 One-Round 令牌传递算法沿着它所在 IP 层的逻辑环传播成员信息.沿层次环结构传播成员改变/成员关系更新消息的成员信息传播算法见算法 2.

算法 2. 沿着层次环结构的成员信息传播算法.

输入:层次环结构中的每一个代理所维护的不同层次的成员信息;

输出:沿着层次环结构自底向上传播成员改变/成员关系更新信息.

```
1.   ParFor each proxy running the proposed protocol Do {
2.    On Timeout event for generating MU messages at a leader proxy:
3.     Generate and Send MU messages to the leader proxy's parent (if exists);
4.    On Receiving an MC message at a proxy: //该代理是 DP;
5.     Keep the MC message in the proxy's MQ;
6.      // MQ 中的消息按照 One-Round 令牌传递的成员信息传播算法传送;
7.    On Receiving an MU message at a proxy: //该代理是 IP;
8.     Keep the MU message in the proxy's MQ;
9.      // MQ 中的消息按照 One-Round 令牌传递的成员信息传播算法传送.
10.  }
```

层次环中每个代理独立地收集、生成、发送 MC/MU 消息,成员信息传播算法以并行及分布式的方式运行.为加快成员信息在逻辑环中的传播并提高其可靠性,可以在每个逻辑环中设置双向令牌以传送成员信息.双向令牌实际上是在逻辑环的两个相反方向上运行且互相独立的两个令牌.为简单起见,算法中没有表示出双向令牌.需要指出的是,该算法也没有表示两个令牌之间的信息交换,这是因为只是使用两个令牌在同一个逻辑环中传播成员信息,这样可以比使用单个令牌时的可靠性高些;另外,本文提出的成员管理协议只需要提供弱一致性的成员关系信息,因此不需要两个令牌中传播严格一致的成员信息.

这里指出,在成员信息传播子协议中可以使用预留机制让 DP 的若干个邻接 DP 事先加入层次环.如果 DP 察觉到其包含的活跃成员从零变到非零,则生成并发送预留消息给其所有邻接 DP.DP 一旦收到该消息,若已包含活跃成员,则简单地丢弃该消息;否则,立即加入层次环.这样,如果 MH 切换到已经执行过预留操作的 DP 中,则 MH 立即通过新的 DP 加入层次环.

这里还需指出,层次环结构在两种情况下会动态改变:一是结构中出现错误时(将在第 4 节讨论);另一种是 MH 动态加入、退出、切换以及因为群组成员出错而离开时.对于后一种情况,存在两类拓扑结构改变事件:一类是 Proxy-Join/Leave 事件,引起单独的结点加入或退出结构中同一层上的某个逻辑环;另一类是 Proxy-Attach/Detach 事件,引起相邻两层上的结点之间的父子关系的建立或结束.

成员信息传播算法中的两种情况会导致拓扑结构改变事件:

一种情况发生在 DP 收到成员信息传播算法生成的 MC 消息时:如果 MH 切换到另一个 DP(或加入一个 DP)并且它正好是附着到该 DP 上的第一个成员,则该 DP 开始加入该结构;若该 MH 是它所退出(或出错所在)的 DP 中的最后一个成员,则该 DP 可能开始从结构上退出.DP 层的 Proxy-Attach/Detach 事件可能进一步引发 DP 的

父亲 IP 相应地加入、退出、附着或离开该结构.

第 2 种情况发生在 IP 收到成员信息传播算法生成的 MU 消息时:若收到 MU 消息的 IP 所维护的活跃成员数由零变为非零,并且该 IP 不在结构中,将会导致 Proxy-Join 或 Proxy-Attach 事件;同样地,若活跃成员数由非零变为零,并且该 IP 在该结构中,则会导致 Proxy-Leave 或 Proxy-Detach 事件的发生.

为了避免 Proxy-Leave/Detach 事件引起结构的频繁重构,本文采用 Lazy-Leave/Detach 机制.只有当 DP 在整个超时间隔内都不包含活跃成员,并且它的所有邻居 DP 也不包含活跃成员时,它才离开该结构.相应地,对于 IP,只有当它在整个超时间隔内都不包含任何孩子结点时才离开该结构.

3.3 成员信息传播子协议的性能分析

当成员加入、退出、切换、出错以及层次环结构中结点出错时,层次环结构会动态变化.下面的分析中使用了以下性能度量:

Join-Delay,定义为从 MH 发出加入该群组的意愿到它接收到第 1 个组播数据包的时间间隔;

Handoff-Delay,定义为从 MH 检测出它已经进入一个新的 DP 到它从新 DP 接收到第一个组播数据包的时间间隔;

Service-Speed,定义为从 MH 发出加入该群组的意愿到它的成员信息成功地在顶层逻辑环的领导结点登记的时间间隔;

Signaling-Overhead,定义为成员信息传播与拓扑结构维护中所有代理接收到的信号数据包总数或总大小除以代理的总数,然后再除以总的执行时间,它代表协议的平均开销.

本文基于上述度量作最坏情况分析.这里,使用 Member-Join 消息在层次环结构中传送的过程为例来说明.

假设 3.1. 如果MH通过其附着的DP请求加入群组,该DP总是成功地把MH登记为群组成员,而无论该DP目前是否处于层次环结构中.假设该操作时间的界为$T_{MH\text{-}Join\text{-}Registration}$.

假设 3.2. 如果代理通过候选兄弟结点或候选父结点请求加入层次环结构,则该代理总是成功地通过其中一个候选兄弟结点与同一层的兄弟逻辑环合并,或通过其中一个候选父结点附着到上一层逻辑环.假设该操作时间的界为$T_{Proxy\text{-}Merge\text{-}Attach}$.

假设 3.3. 如果Member-Join消息已经进入代理的MQ中,该消息总是被成功地传送到该代理所在逻辑环的领导结点.假设该操作时间的界为$T_{Proxy\text{-}MS\text{-}Leader}$.

假设 3.4. 如果Member-Join消息到达领导结点的ListOfMembers[],该消息总是成功地传送到该领导结点的父结点(如果父结点存在).假设这个操作时间的界为$T_{Proxy\text{-}MS\text{-}Parent}$.

假设 3.5. 假设层次环结构中的任何MH总是成功地接收组播消息的时间的界为$T_{Multicast\text{-}Transmission}$.

这里把上述假设作为一个整体称为 Always-Successful-assumptions(简称 AS-assumptions).AS-assumptions 是合理的:(1) 为了提供给用户最佳的服务,服务提供商或网络运营商有责任在其服务范围内合理地部署所有的代理;(2) 如果代理部署不合理,则处于相应区域的用户可能得不到好的服务.在第 1 种情况下,AS-assumptions 成立;而在第 2 种情况下,AS-assumptions 有可能不成立.

有了 AS-assumptions,本文首先证明 Join-Delay 与 Service-Speed 度量的界.

定理 3.1. 假设 AS-assumptions 在高度为 h 的层次环结构中成立,在最坏情况下,Join-Delay 的界为

$$T_{MH\text{-}Join\text{-}Registration}+(h-1)\times(2\times T_{Proxy\text{-}Merge\text{-}Attach}+T_{Proxy\text{-}MS\text{-}Leader}+T_{Proxy\text{-}MS\text{-}Parent})-T_{Proxy\text{-}Merge\text{-}Attach}-T_{Proxy\text{-}MS\text{-}Parent}+T_{Multicast\text{-}Transmission} \quad (1)$$

证明:假设逻辑环的领导结点只通过其候选父结点附着到层次环结构中,不会通过其候选兄弟结点与其他逻辑环合并.在最坏情况下:(1) 当MH发送Member-Join消息给其附着的DP以请求加入群组时,首先花费$T_{MH\text{-}Join\text{-}Registration}$时间把它登记为其附着的DP中的群组成员;(2) 如果该DP恰好不在层次环中,它就请求通过其候选兄弟结点之一与逻辑环合并,这将花费$T_{Proxy\text{-}Merge\text{-}Attach}$时间;(3) Member-Join消息沿逻辑环传送到领导结点,将花费$T_{Proxy\text{-}MS\text{-}Leader}$时间;(4) 如果该领导结点恰好不在层次环中,则请求通过其候选父结点之一附着上一层逻辑环,这将花费$T_{Proxy\text{-}Merge\text{-}Attach}$时间;(5) Member-Join消息从领导结点传送到其父结点,这将花费$T_{Proxy\text{-}MS\text{-}Parent}$时

间; (6) 对于上面h–2 层的IP层,重复以上步骤(2)~步骤(5);(7) 至此,建立起沿层次环自底向上成员信息传播的一条完整“路径”;(8) 组播数据包沿反向“路径”自顶向下传播,最终被MH接收到,这将花费$T_{Multicast\text{-}Transmission}$时间. □

定理 3.2. 假设 AS-assumptions 在高度为 h 的层次环结构中成立,在最坏情况下,Service-Speed 的界为

$$T_{MH\text{-}Join\text{-}Registration}+(h-1)\times(2\times T_{Proxy\text{-}Merge\text{-}Attach}+T_{Proxy\text{-}MS\text{-}Leader}+T_{Proxy\text{-}MS\text{-}Parent})-T_{Proxy\text{-}Merge\text{-}Attach}-T_{Proxy\text{-}MS\text{-}Parent} \quad (2)$$

证明:证明过程与定理 3.1 的证明过程相似.唯一的区别是这里去掉了传送组播数据包的时间.

从上述两个定理来看,Join-Delay 的界还大于 Service-Speed 的界.实际上,这只是最坏情况才可能如此.在平均情况下,Join-Delay 很小,因为 MH 附着的 DP 可能已经位于层次环中.然后论证 Signaling-Overhead 度量.每个代理的 MQ 中所有成员消息都会作为一个聚合消息沿每个逻辑环传送,这样信号开销会大为减少.另外,成员信息传播从逻辑环的领导结点到领导结点的父结点基于超时机制,这也减少了信号开销.因此,可以认为层次环结构中每个代理的信号开销的界为常数.对 Handoff-Delay 度量可以作相似的分析,这里从略.因为切换处理过程通常采用某种预留机制,当 MH 切换到一个新的 DP 中时,大部分时间都能立即接收到组播消息.因此,在平均情况下,Handoff-Delay 度量都比 Join-Delay 要小,有关这一点将在第 6 节的模拟分析中得以验证. □

4 基于邻居检测的拓扑结构维护子协议

本节研究层次环中的两种错误对协议的影响:(1) 代理错误,可导致层次环结构中父子关系、前后关系结束;(2) 网络错误,可造成在网络分区期间系统中形成若干个互不相交的子层次环结构.本节首先描述处理代理错误的 Neighbor-Repair 算法,然后描述处理网络错误的 Partition-Repair 算法.

4.1 Neighbor-Repair算法

处理代理错误的 Neighbor-Repair 算法如下.对于层次环结构中的每一个代理:

情况 1. 如果代理是领导结点,且其错误检测部件(failure detection component,简称 FDC)报告其父代理出错,则将该代理的 ParentOK 设置为 FALSE,指示父子关系结束.然后,它与其中一个可达候选父结点建立新的父子关系,称为 ATTACH 过程,或通过其中一个可达候选兄弟结点与逻辑环合并,称为 MERGE 过程;

情况 2. 如果代理的 FDC 报告其孩子代理出错,则将该代理的 ChildOK 设置为 FALSE,指示父子关系结束;

情况 3. 如果代理的 FDC 报告其前代理出错,则将 PreviousOK 设置为 FALSE.为去掉逻辑环中的错误结点,代理将发送 Previous-Repair 消息给前代理的前代理,建立它们之间的兄弟关系.

如果代理错误不常发生,与 Previous-Repair 消息相关的 Neighbor-Repair 算法会很快终止,本文称这部分算法为 Fast-Neighbor-Repair 过程.如果出错的代理恰好是领导结点,在 Fast-Neighbor-Repair 算法终止后,该逻辑环将没有领导结点.为了解决该问题,发送 Previous-Repair 消息的代理将自己设为新的领导代理,然后发送 Leader-Change 消息通知逻辑环中所有代理更新其领导代理.如果代理错误频繁发生,则如果前代理的前代理也出错,则 Previous-Repair 消息将永久丢失.在这种情况下,在超时间隔过后,Slow-Neighbor-Repair 过程就会启动.为了从逻辑环中去掉两个以上连续出错的代理,则发送 Dest-Find 消息沿逻辑环中查找目标代理(称为 DESTINATION),然后建立它们之间的兄弟关系.DESTINATION 是无法沿逻辑环可靠地转发 Dest-Find 消息的那个代理.因为 Dest-Find 消息沿逻辑环传送给 DESTINATION,与 Fast-Neighbor-Repair 过程相比较,这部分算法需要花更长的时间才能终止,本文称其为 Slow-Neighbor-Repair 过程.如果领导结点恰好是出错的结点之一,则在 Slow- Neighbor-Repair 算法终止之后,该逻辑环将没有领导结点.为了解决这个问题,需要 Dest-Find 消息收集它到 DESTINATION 的路径上的领导结点信息.如果领导结点不在路径上,DESTINATION 就被设为新的领导结点,再发送一条 Leader-Change 消息以通知新的领导结点信息.

情况 4. 如果代理的 FDC 报告其后代理出错,则将该代理的 NextOK 设为 FALSE.首先执行 Fast-Neighbor-Repair 过程;如果在超时间隔过后它还没有终止,则执行类似的 Slow-Neighbor-Repair 过程.

以上是 Neighbor-Repair 算法的基本设计.但在基本设计中,由于情况 3 和情况 4 的对称操作会导致以下问题:(1) 重复操作.因为出错代理的两个邻居可能同时检测到其错误状态,它们可能分别独立地启动执行

Neighbor-Repair 算法.该算法仍然可以正常运行,但是同样的操作会执行两次;(2) 领导信息不一致.由于情况 3 和情况 4 的对称操作,在发送 Leader-Change 消息时,逻辑环中各个代理上的领导结点信息可能会不一致;(3) 逻辑环不一致.考虑代理实际上没有出错但被错误地怀疑为出错的情况,这里需要被错误怀疑的代理将正在检测的代理也看作出错.但是,这样会导致逻辑环不一致.例如,有 5 个代理的逻辑环具有前后关系:$A\leftrightarrow B\leftrightarrow C\leftrightarrow D\leftrightarrow E\leftrightarrow A$;然后假设 C 错误地怀疑 D 出错,之后 D 也会怀疑 C 出错;然后 C 发送 Next-Repair 消息给 E,D 也发送 Previous-Repair 消息给 B;再假设 C 和 E 的兄弟关系在 B 和 D 的兄弟关系之前已建立,即去掉 D 首先形成了一致的逻辑环 $A\leftrightarrow B\leftrightarrow C\leftrightarrow E\leftrightarrow A$,再去掉 C 将会形成不一致的逻辑环:包括后一条链路的逻辑环 $A\rightarrow B\rightarrow \boldsymbol{D}\rightarrow E\rightarrow A$ 和包括前一条链路的逻辑环 $A\leftarrow B\leftarrow \boldsymbol{C}\leftarrow E\leftarrow A$.为避免出现这些问题,只需简单地去掉情况 3 或情况 4 即可.

4.2 Partition-Repair算法

如果网络出错,层次环结构被划分成若干个子层次环结构(理想情况下,每个网络分区对应一个子层次环结构),当网络分区合并时,若干个子层次环结构又合并在一起.在出现分区期间,层次环结构中的一个逻辑环可能破裂成两个或更多个部分.逻辑环的每个部分通过运行 Neighbor-Repair 算法将修复成一个新的逻辑环.本文在网络出现分区的情况下也使用第 4.1 小节的 ATTACH 和 MERGE 过程.只有处于“顶层”逻辑环的领导结点可以发送请求以启动这两个过程.首先定义一个领导结点的可达候选邻居结点队列(a queue of reachable candidate neighbors,简称 QRCN)作为可达候选邻居结点的集合.QRCN 是在使用检测可达性时保留在返回的探测消息中的信息而动态形成的.若 QRCN 不为空,则领导结点从其中选择一个候选邻居,该候选邻居必须不在领导结点位于的子层次环结构中,领导结点再用该候选邻居启动 ATTACH 或 MERGE 过程来处理网络分区的合并.

处理 ATTACH 和 MERGE 过程的 Partition-Repair 算法如下.对于网络分区子层次环结构中“顶层”逻辑环的领导结点来说:

情况 1. ATTACH 过程用来在两个代理之间建立新的父子关系.为了确保层次环结构的一致性维护,本文采用了两阶段提交协议,把每次维护看作一个事务.每次 ATTACH 事务只涉及两个代理:LEADER 及其某个候选父结点代理(称为 PARENT).在第 1 阶段,LEADER 发送 Leader-Attach 消息给 PARENT,PARENT 肯定地或否定地应答.在第 2 阶段,LEADER 确认或回滚,通知 PARENT 相应地做确认或回滚.

情况 2. MERGE 过程用来把两个逻辑环合并成一个逻辑环.每个 MERGE 事务涉及 4 个代理,即 LEADER、LEADER 的后结点(称为 NEXT)、CANDIDATE 和 CANDIDATE 的后结点(称为 CANDIDATE-NEXT).在第 1 阶段,LEADER 发送一个 Leader-Merge 消息给其他 3 个代理,然后,这 3 个代理做出肯定或否定的回复.在第 2 阶段,如果 LEADER 收到的 3 个回复都是肯定的,LEADER 就确认并通知另外 3 个代理确认;否则,LEADER 回滚并通知另外 3 个也回滚.当这两个逻辑环成功地合并成一个时,则 LEADER 发出一个从探测消息得到的领导结点信息的 Leader-Change 消息,通知 LEADER 原来逻辑环中所有代理更改新的领导结点信息.在 MERGE 过程中,如果正在合并的两个逻辑环恰好是它们各自子层次环结构中的“顶层”逻辑环,则两个 LEADER 中结点标识符较大的结点才允许启动其事务.

4.3 拓扑结构维护子协议的性能分析

当考虑到网络出现分区时,下面所提到的特征和界将适用于每个分区.

特征 4.1. 及时性.该特征度量拓扑结构维护子协议对错误事件响应的时间.

(1) 拓扑结构维护子协议基于 Freshness points 方法来检测错误并获得可达性信息,该方法满足及时性.

(2) 所有拓扑结构维护过程都是基于局部信息的,因为它们只需要知道当前邻居和候选邻居的信息.

(3) 所有拓扑结构维护过程以并行与分布式方式运行并将在确定时间内终止.对于 Fast-Neighbor-Repair 和 Slow-Neighbor-Repair 过程,如果前者在超时间隔后没有终止时,后者就会启动然后在确定时间内终止.尤其对于 ATTACH 和 MERGE 过程,当一个过程不能正常终止,它就从其可达候选邻居集合中选择另一个候选邻居启动另一个过程.本文假设选择过程在所有可达候选邻居中运行过一轮后,至少有一个过程会正常终止.

特征 4.2. 准确性.这个特征度量错误发生时拓扑结构维护子协议维护层次环结构的准确程度.

(1) 基于 Freshness points 方法可以高准确性地检测出代理错误和网络分区与网络合并.

(2) 如果 Freshness points 方法错误地怀疑没有出错的代理,就从逻辑环中去掉正在检测的代理或被错误怀疑的代理.这样,去掉的代理将变成一个独立的代理,然后自己单独加入层次环结构中.

基于及时性和准确性,可以认为执行每次拓扑结构维护过程的时间的界为某个常数,而且层次环结构的维护是非常准确的.另外,可以认为成员信息传播子协议中类似的性能度量的界也存在.

5 层次环结构的可扩展性和可靠性的分析

5.1 可扩展性的比较分析

在文献[16]中,成员服务器被组织成一个基于代表结点的树结构,称为 CONGRESS 层次结构,其中包含局部成员服务器和全局成员服务器.本节针对层次环结构与 CONGRESS 层次结构作可扩展性比较.树结构/层次环结构中的 LMS/DP 个数 n 作为等效的可扩展性参数.为简单起见,下面分别计算沿 CONGRESS 层次结构和层次环结构将成员改变消息分别传送给所有 LMS 和所有 DP 时所经过的逻辑链路的总数 *HopCount*,它近似于 CONGRESS 层次结构或层次环结构中边的总数.

首先考虑CONGRESS层次结构.在高度为h≥4、每个GMS的分支数r≥2 的层次结构中,LMS的个数为$n=r^{h-2}$,则*HopCount*为

$$HopCount_{Tree\text{-}based}(n,h,r) \overset{def}{=} \sum_{i=0}^{h-3} r^{i+1} \tag{3}$$

然后计算高度为 h≥3、每个逻辑环包含 r≥2 个结点的层次环结构中的 *HopCount*.底层逻辑环中的 DP 总数为$n=r^{h-1}$,逻辑环的总数为 $tn=\sum_{i=0}^{h-2} r^i$,则*HopCount*为

$$HopCount_{Ring\text{-}based}(n,h,r) \overset{def}{=} ((r+1)\times tn-1) \tag{4}$$

正规化 *HopCount*,即把它除以 n,它代表一条成员改变消息引起的成员信息传播的“平均”消息个数.分别把 CONGRESS 层次结构和层次环结构的正规化 *HopCount* 记为 $HC^N_{Tree\text{-}based}$ 和 $HC^N_{Ring\text{-}based}$:

$$HC^N_{Tree\text{-}based} \overset{def}{=} HopCount^N_{Tree\text{-}based}(n,h,r) \overset{def}{=} \frac{\sum_{i=0}^{h-3} r^{i+1}}{n} \tag{5}$$

$$HC^N_{Ring\text{-}based} \overset{def}{=} HopCount^N_{Ring\text{-}based}(n,h,r) \overset{def}{=} \frac{(r+1)\times tn-1}{n} = \frac{(r+1)\times\sum_{i=0}^{h-2} r^i-1}{n} \tag{6}$$

根据式(5)和式(6),我们给出了表 1 的数值分析结果.可见,在给定相同个数的 LMS/DP 的情况下,层次环结构的可扩展性与树结构的可扩展性一样好.

Table 1 Comparison on scalability between the tree-based and ring-based hierarchies

表 1 树结构与层次环结构的可扩展性对比

n	h	r	$HC^N_{Tree\text{-}based}$	n	h	r	$HC^N_{Ring\text{-}based}$
16	4	4	1.250 0	16	3	4	1.500 0
64	5	4	1.312 5	64	4	4	1.625 0
256	6	4	1.328 1	256	5	4	1.656 3
36	4	6	1.166 8	36	3	6	1.333 3
216	5	6	1.194 4	216	4	6	1.388 9
1 296	6	6	1.199 1	1 296	5	6	1.398 2
64	4	8	1.125 0	64	3	8	1.250 0
512	5	8	1.140 6	512	4	8	1.281 3
4 096	6	8	1.142 6	4 096	5	8	1.285 2
100	4	10	1.100 0	100	3	10	1.200 0
1 000	5	10	1.110 0	1 000	4	10	1.220 0
10 000	6	10	1.111 0	10 000	5	10	1.222 0

5.2 可靠性的比较分析

首先说明:如果把层次环结构中每个逻辑环看作是一个结点,层次环结构与树结构是同构的,则层次环结构的可靠性比树结构更高.这是因为,当只考虑出现一个结点出错的情形时,显然,由于树结构中一个错误会把树结构破裂成若干个部分,而层次环结构中一个错误会使得一个逻辑环破裂,该结构可能不会破裂成若干个部分(除非该错误结点具有孩子结点).实际上,当考虑同时出现多个错误时,也会存在类似状况.

然后说明:如果两个结构是同构的,树结构比基于代表结点的树结构具有更高的可靠性.这是因为,在基于代表结点的树结构中,一个代表结点出错实际上是若干个逻辑结点出错,而在树结构中一个结点出错只是它本身出错.因此,树结构比基于代表结点的树结构更可靠.

下面只需说明层次环结构比树结构更可靠.树结构中如果一个结点出错,则其孩子结点只有一种选择,即去寻找新的父结点并附着其上.而层次环结构中如果一个代理出错,则其孩子结点有两种选择:通过候选兄弟结点之一与同一层逻辑环合并,或者通过候选父结点之一附着到上一层逻辑环.与此同时,通过运行 Neighbor-Repair 算法把出错代理从逻辑环中去掉.因此在出错的情况下,层次环结构比树结构可以更可靠地维护.

以上只是非形式的比较分析.下面从数学上给出严格的比较分析.首先定义 Function-Well(运行良好)概念.树结构中的一个出错结点会影响到其所有孩子结点,因此,这里定义树结构的 Function-Well 如下:如果允许在除最低两层以外的层次结构中至多有 k 个结点同时出错,则称该树结构是 Function-Well 的.为了计算树结构的 Function-Well(简称 fw)概率,假设层次结构是满的:它包含最大层数(记为 h),层次结构中每个代理有最多的孩子结点数(记为 r).直观来看,在同样的网络环境中,小层次结构比大层次结构更可能 Function-Well.因此,这里使用满层次结构作最坏情况分析.该层次结构中倒数第 2 层包含$n=r^{h-2}$个结点,从第 1 层~倒数第 3 层包含 $tn=\sum_{i=0}^{h-3} r^i$ 个结点,在该层次结构中至多 k 个结点无法 Function-Well.假设 f 是层次结构中具有均匀且独立分布的结点出错概率(node failure probability),则树结构的 Function-Well 概率为

$$Prob_{Tree\text{-}based}^{fw}(n,h,r,f,k)\overset{def}{=}\sum_{i=0}^{k}\binom{tn}{i}(1-f)^{tn-i}\times f^i \tag{7}$$

再计算逻辑环的 Function-Well 概率.如果在逻辑环中不能出现两个连续的代理同时出错,则只需要 Fast-Neighbor-Repair 过程修复逻辑环.在这种情况下,称逻辑环 Function-Well.如果逻辑环无法 Function-Well,即逻辑环中至少有两个连续的代理同时出错,则需要 Slow-Neighbor-Repair 过程修复逻辑环.

然后讨论计算整个层次环的 Function-Well 概率所使用的参数:n 表示 DP 的个数,h 表示层次环的高度,r 表示逻辑环中的结点数,f 表示层次环结构中具有均匀且独立分布的结点出错概率,k 表示不能 Function-Well 的逻辑环的最大个数.如果层次环结构中至多 k 个逻辑环不能 Function-Well,则认为该层次环结构是 Function-Well 的.定义每个逻辑环的 Function-Well 概率 t 为

$$t=Prob_{ring}^{fw}\overset{def}{=}\sum_{i=0}^{\left\lfloor\frac{r}{2}\right\rfloor}B(r,i)(1-f)^{r-i}\times f^i \tag{8}$$

在式(8)中,$B(r,i)$代表错误发生的个数,即在逻辑环的 r 个结点中恰好有 i 个结点出错,而且不会发生互为邻居的两个结点出错.对于 $0\le i\le 3$ 与 $r\ge 5$,很容易推断出 $B(r,i)$如下:

$$B(r,0)=\binom{r}{0}=1 \tag{9}$$

$$B(r,1)=\binom{r}{1}=r \tag{10}$$

$$B(r,2)=\binom{r}{2}-\binom{r}{1}=\frac{r(r-3)}{2} \tag{11}$$

$$B(r,3)=\binom{r}{3}-\binom{r}{1}-\binom{r}{1}\binom{r-4}{1}=\frac{r(r^2-9r+20)}{6} \tag{12}$$

而当 $i\geq 4$ 时难以得到相应的公式,但可以由计算机程序计算在 $0\leq i\leq\left\lfloor\frac{r}{2}\right\rfloor$ 时 $B(r,i)$的值.表 2 是其数值结果.

接着,在假设满层次环结构时作最坏情况分析.该层次环包含最大层数,每个逻辑环具有最多结点个数.这样,层次环中包含 $tn=\sum_{i=0}^{h-2}r^i$ 个逻辑环,至多 k 个逻辑环不能 Function-Well.层次环的 Function-Well 概率表示为

$$Prob^{fw}_{Ring\text{-}based}(n,h,r,f,k)\overset{def}{=}\sum_{i=0}^{k}\binom{tn}{i}t^{tn-i}(1-t)^i \tag{13}$$

从式(7)和式(13)得出的数值分析结果见表 3.结论如下:(1) 在网络规模和结点出错概率相同的情况下,层次环结构总比树结构更为可靠.特别地,随着 LMS/DP 个数的增多和结点出错概率的增大,树结构的可靠性大为降低,而层次环的可靠性只会适度地降低.例如,在最大层次环结构中,当结点出错概率从 0.1%增大到 5.0%时,树结构的 Function-Well 概率从 99.980%降到了 7.999%,层次环结构从 100.000%只降到 51.216%;(2) 当结点出错概率设定为 0.1%,并且 DP 个数多达 1 000 个时,层次环结构只需运行 Fast-Neighbor-Repair 过程修复该结构,此时,该结构的 Function-Well 概率高达 99.889%;如果允许 Slow-Neighbor-Repair 过程最多执行 2 次,则概率可高达 100.0%;(3) 在允许 Slow-Neighbor-Repair 过程至多执行 2 次的 Function-Well 层次环的定义以及 99.980%的高概率的情况下,拥有 DP 个数多达 1 000 个的群组可确保在结点出错概率设定为 0.1%时,该结构仍能 Function-Well;(4) 当结点出错概率增大到 5.0%时,小规模的层次环结构仍然能以高概率 Function-Well.例如,对于包含 64 个 DP 的小规模层次环而言,其 Function-Well 的概率是 99.900%,但是对于包含 1 000 个 DP 的大规模层次环只有 51.216%的低概率 Function-Well.

Table 2 $B(r,i)$ values for computing the function-well probability of a logical ring

表 2 计算逻辑环的 Function-Well 概率的 $B(r,i)$值

r	I	$B(r,i)$	r	i	$B(r,i)$
4	0	1	8	2	20
4	1	4	8	3	16
4	2	2	8	4	2
6	0	1	10	0	1
6	1	6	10	1	10
6	2	9	10	2	35
6	3	2	10	3	50
8	0	1	10	4	25
8	1	8	10	5	2

Table 3 Comparison on reliability between the tree-based (h=5) and ring-based (h=4) hierarchies

表 3 树结构(h=5)与层次环结构(h=4)的可靠性比较

n	r	f(%)	k	$Prob^{fw}_{Tree\text{-}based}$ (%)	$Prob^{fw}_{Ring\text{-}based}$ (%)	n	r	f(%)	k	$Prob^{fw}_{Tree\text{-}based}$ (%)	$Prob^{fw}_{Ring\text{-}based}$ (%)
64	4	0.1	0	97.921	99.992	512	8	0.1	0	92.957	99.942
64	4	0.1	1	99.979	100.000	512	8	0.1	1	99.749	100.000
64	4	0.1	2	100.000	100.000	512	8	0.1	2	99.994	100.000
64	4	1.0	0	80.973	99.172	512	8	1.0	0	48.014	94.381
64	4	1.0	1	98.149	99.997	512	8	1.0	1	83.419	99.841
64	4	1.0	2	99.884	100.000	512	8	1.0	2	96.293	99.997
64	4	5.0	0	34.056	81.825	512	8	5.0	0	2.365	24.776
64	4	5.0	1	71.697	98.317	512	8	5.0	1	11.451	59.678
64	4	5.0	2	91.508	99.900	512	8	5.0	2	28.668	83.925
216	6	0.1	0	95.789	99.974	1000	10	0.1	0	89.489	99.889
216	6	0.1	1	99.912	100.000	1000	10	0.1	1	99.432	100.000
216	6	0.1	2	99.999	100.000	1000	10	0.1	2	99.980	100.000
216	6	1.0	0	64.910	97.478	1000	10	1.0	0	32.772	89.591
216	6	1.0	1	93.104	99.969	1000	10	1.0	1	69.517	99.443
216	6	1.0	2	99.084	100.000	1000	10	1.0	2	89.931	99.980
216	6	5.0	0	11.018	53.987	1000	10	5.0	0	0.337	7.0719
216	6	5.0	1	35.955	87.506	1000	10	5.0	1	2.304	26.031
216	6	5.0	2	63.516	97.669	1000	10	5.0	2	7.999	51.216

6 性能评价

6.1 模拟方案

本文的模拟实验中使用NS-2 模拟工具[31]模拟 4 层的层次环结构.DP被配置成$m\times n$的网格.IPT1 中有$\frac{m\times n}{4}$个IP,IPT2 中有$\frac{m\times n}{4}$个IP和$\frac{3m\times n}{2}$个其他代理,其中一个代理还作为组播源.初始时,在稀疏模式(sparse mode,简称 SM)模拟过程中,每个 DP 都有一个 MH 附着;在稠密模式(dense mode,简称 DM)模拟过程中,每个 DP 有 4 个 MH 附着.在任何时候,SM 模拟中由 8 个 DP 构成的集合中大约有 1 个群组成员,DM 模拟的每个 DP 中大约有 1 个群组成员.每个代理初始配置 4 个候选兄弟结点和 4 个候选父结点.图 3 是一个 4×4 的 DP 配置的例子,每个 DP 逻辑环最初由 4 个 DP 构成.本文模拟了 8×8,12×12,16×16,20×20 的 DP 配置,每个 DP 逻辑环最初由 4 个 DP 构成,每个 IP 逻辑环最初分别由 4,6,8,10 个 IP 构成.SM 模拟中的最小拓扑结构由 64 个 MH、64 个 DP、IPT1 中的 16 个 IP、IPT2 中的 16 个 IP 和 96 个其他代理构成,其结点总数是 256 个.如果每个 DP 的覆盖区域为 670m×670m,则其总覆盖区域为 5360m×5360m.DM 模拟中的最大拓扑结构由 1 600 个 MH、400 个 DP、IPT1 中的 100 个 IP、IPT2 中的 100 个 IP 和 600 个其他代理构成,其结点总数是 2 800 个.如果每个 DP 的覆盖区域为 670m×670m,则其总覆盖区域为 13400m×13400m.

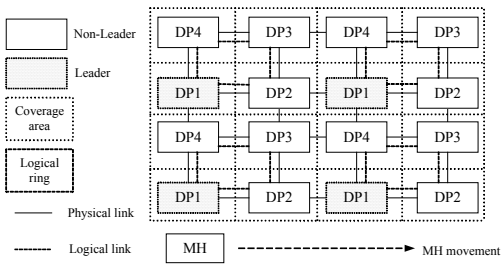


Fig.3 Example 4×4 DP configuration

图 3 4×4 DP 配置的例子

在所有方案中,模拟总时间为 600s.有线链路带宽为 10Mbps,链路延迟为 10ms,消息丢失率为 1.0%;DP和MH之间的无线链路的等效带宽为 2 Mbps,链路延迟为 20ms,消息丢失率为 2.0%.信号消息重传的超时间隔设定为 100ms,最大重传次数为 3.在作移动性检测时,每个DP的超时间隔为 1s,MH向其附着的DP刷新其状态的超时间隔设为 1s.每个逻辑环中领导结点的成员更新间隔为 1s.采用最大速率为 15m/s、暂停时间为 5s的CMU移动性模型[31]和每个代理从层次结构真正退出、离开的Lazy-Leave/Detach超时间隔设为 3s.每个领导结点检测成员信息传播使用的令牌是否丢失的超时间隔设为 3s.

假设每个MC消息大小为 10 字节,DP、IPT1 层的IP和IPT2 层的IP生成的MU消息大小分别为 20,30,40 字节.DPT,IPT1 和IPT2 中传播成员信息的每个令牌大小分别为 20,35,70 字节.采用消息大小为 512 字节以及包速率为每50ms发送一个数据包的定常比特率(constant bit-rate,简称CBR)的通信流进行通信.为了模拟动态的网络环境,通过仿真每一个代理所有直接的链路同时断裂以模拟该代理出错,但不模拟任何MH成员出错.本文使用NS-2 中的确定性模型[31],有以下 4 个参数:Start-Time表示代理开始出错的时间,Up-Interval和Down-Interval表示代理在某段时间内是好的(即没有出错)和是坏的(即出错),Ratio是模拟中可能出错的代理数与代理总数的比例.对于所有情况,Start-Time设为从 0.0s~100.0s的一个随机值.使用{Up-Time,Down-Time, Ratio}的三元组表示结点出错概率,在模拟中设定{95.0s,5.0s,0.2},{95.0s,5.0s,1.0},{90.0s,10.0s,1.0}分别表示 1.0%,5.0%和 10.0%的结点出错概率.

在代理检测其当前邻居是否出错的 Freshness points 方法中,η和σ参数设为 50ms 和 200ms,在检测候选邻居可达性时,η和σ参数设为 50ms 和 250ms.Fast-Neighbor-Repair 过程启动 Slow-Neighbor-Repair 过程的超时间隔

设为 1s.在每个“顶层”逻辑环中,领导结点发送请求启动 ATTACH 或者 MERGE 过程的超时间隔设为 100ms.

为了模拟成员关系动态性,本文设计了使用两个参数的 Join/Leave 模式:Minimal/Maximal Interval 定义为同一个 MH 的任意两个连续的 Member-Join/Member-Leave 事件间的最小/最大时间间隔,分别设为 50s 及 70s. MH 触发其 Member-Join 事件的开始时间定义为从 1.3s~20.0s 的一个随机变量.为了动态地控制群组成员与所有 MH 的成员比例为某个预期值,每次触发 Member-Join 事件就采用一个随机变量以确定 MH 是否真的加入该群组.例如,为把 SM 模拟中的成员比例控制为 12.5%,即大约 8 个 DP 的集合中只有 1 个成员的情况,如果与 MH 的 Member-Join 事件相关的随机变量小于 12.5%,那么 MH 加入该群组;否则就忽略 Member-Join 事件.

6.2 模拟结果

本文根据不同的网络规模和不同结点出错概率进行了大量的针对 SM 和 DM 模式的模拟.把 10 次独立的模拟程序得出的平均值作为模拟结果.图 4 和图 5 分别表示可扩展性和可靠性.

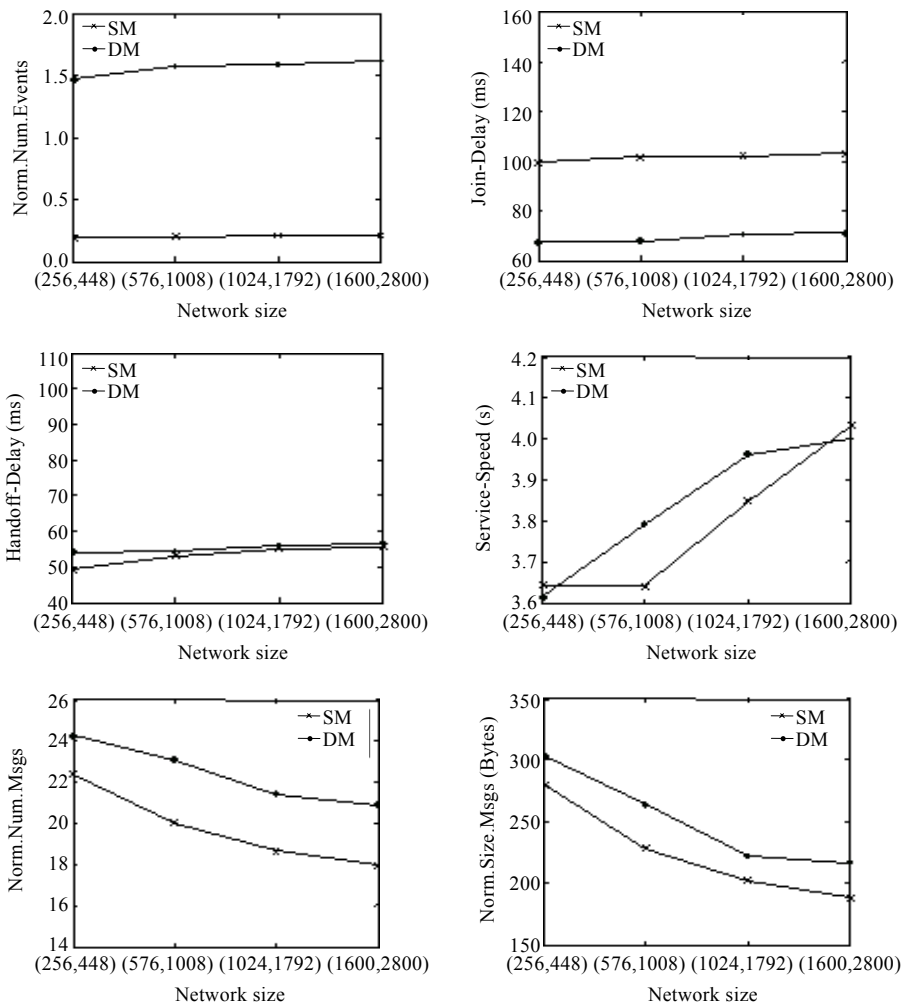


Fig.4 Simulation results for scalability property

图 4 可扩展性的模拟结果

在图 4 中,*X* 轴表示网络规模,*Y* 轴表示所评估的性能度量.网络规模以成对的数值出现,每对数值分别表示 SM 和 DM 中所模拟的结点总数.每个子图中都有对应 SM 和 DM 的两条曲线.这里没有给出移动性检测中的心

跳/探测消息及错误检测中的心跳/探测消息的个数/大小.由于所有拓扑结构中可靠性具有类似的趋势,这里只给出最大拓扑结构时的结果(如图 5 所示,*X* 轴表示结点出错概率,*Y* 轴表示所评估的性能度量).图中还同时表示正规化的 Member-Join/Leave/Handoff 事件的个数(简称 Norm.Num.Events).该度量定义为,在模拟过程中,这些事件的总数除以 DP 总数,再除以总的模拟时间,最后乘以 60.因此,它表示每个 DP 在每分钟内处理这些事件的平均个数.在两个图中,Norm.Num.Events 度量在所有网络规模或者所有结点出错概率的情况下都基本不变.例如,图 4 中 SM 的 4 种网络规模中该度量分别为 0.19,0.20,0.20 和 0.21,而 DM 的 4 种网络规模中该度量分别为 1.47,1.58,1.59 和 1.62.

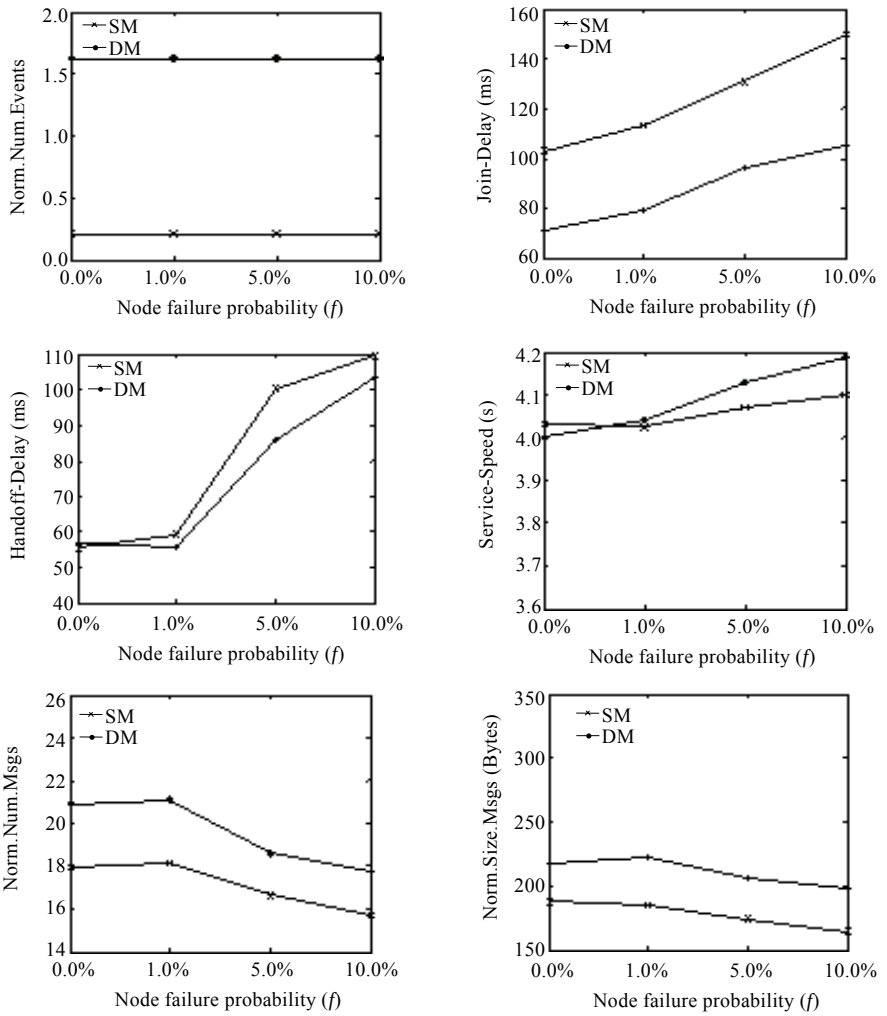


Fig.5 Simulation results for reliability property

图 5 可靠性的模拟结果

从模拟结果可以看出:

(1) 图 4 表示群组成员管理协议的可扩展性非常好.当网络规模增大且成员密度固定时,该协议的性能即 Join-Delay,Handoff-Delay 和 Service-Speed,保持很高且在很小范围内发生变化,然而性能开销即 Norm.Num.Msgs 或 Norm.Size.Msgs 很小且总保持在同一水平.例如,SM 中 4 种网络规模的 Join-Delay 度量分别为 99.42ms,101.39ms,101.83ms,103.18ms,最大变化量仅为 103.18−99.42=3.76ms;DM 中 4 种网络规模的 Join-Delay 度量分

别为 67.11ms,67.90ms,70.24ms,70.70ms,最大变化量仅为 70.70−67.11=3.59ms.另外,SM 中 Norm.Num.Msgs 和 Norm.Size.Msgs 的最大值分别为每个代理每秒 22.38 个信号消息和每个代理每秒 280.43 字节;相应地,DM 中的最大值分别为 24.21 个信号消息和 303.19 字节.Signaling-Overhead 度量的两条曲线中,随着网络规模的扩大,Signaling-Overhead 具有减小的趋势.因为每个逻辑环中用于传播成员改变/成员关系更新消息的令牌消息是所有信号消息中最重要的组成部分之一,可以计算出不同网络拓扑结构中令牌总数与有线结点总数的比例.由于逻辑环的总数也就是令牌的总数会随着层次环结构的变化而变化,可以利用模拟开始时的初始值来估计该值.因此,4 种模拟的网络拓扑结构中该比例分别为 2×(1+4+4×4)/(3×8×8),2×(1+6+6×6)/(3×12×12),2×(1+8+8×8)/(3×16×16),即 21.88%,19.90%,19.02%和 18.50%.可以看到,当网络规模增大时,比例变小.

(2) 图 5 表示,当网络规模和成员密度都固定不变时,层次环中的结点出错概率对该协议的性能有一定的影响.随着结点出错概率从 0.0%增大到 10.0%,性能会适度地降低.例如,SM 中 4 种结点出错概率的 Handoff- Delay 度量分别为 55.72ms,59.38ms,100.19ms,109.50ms,最大错误情况和无错误情况之间的最大变化量仅为 109.50−55.72=53.78ms;DM 中 4 种结点出错概率的 Handoff-Delay 度量分别为 56.51ms,55.66ms,85.77ms,103.41ms,最大错误情况和无错误情况之间的最大变化量仅为 103.41−56.51=46.90ms.

(3) 在图 4 和图 5 中,可以观察到每个性能度量的一些趋势.Join-Delay 度量受到群组成员密度的影响.如果成员分布稠密,例如 DM,则对于 MH 来说很可能它一加入 DP 就能立即收到消息;相反,如果成员分布稀疏,例如 SM,则当 MH 加入 DP 时,由于该 DP 可能不得不开始加入层次环而使 MH 必须等待一段时间才能接收消息.这就是 Join-Delay 子图中两条曲线之间有明显间隙的原因.Handoff-Delay 度量对群组成员密度则不敏感,主要是因为该协议使用一种预留机制,该机制弱化了 SM 和 DM 间的差异.Service-Speed 度量主要受到网络规模的影响,随着网络规模的增大它只是略微地增大.然而,它对结点出错概率和成员密度不敏感.这是因为模拟中把层次结构的高度固定为 4 层,生成 Member-Update 消息和 Membership-Update 消息的超时间隔固定为 1s.而在实际环境中,这些参数根据不同应用的需要可能发生变化.因此,这里只是给出了用于评估该协议的一个相对的 Service-Speed 度量.Signaling-Overhead 度量主要受到成员密度的影响,而对网络规模和结点出错概率不敏感.主要原因是 SM 中群组大小和代理个数比相同网络规模的 DM 中群组大小和代理个数都小得多.因此,SM 中的信号消息个数与大小自然比 DM 中的要少,从而可以看到,Signaling-Overhead 子图中的两条曲线之间有很明显的间隙.

7 结束语

本文的主要贡献是:(1) 提出了一个适合于移动群组通信的新颖的层次环结构,它是逻辑环和逻辑树的结合模型,利用了逻辑环的简单性与逻辑树的可扩展性;(2) 基于该模型提出了一种基于局部群组的成员信息传播算法,对于移动因特网环境中提供可扩展的、可靠的群组通信服务非常重要;(3) 提出了一种能够满足及时性和准确性、基于邻居检测的拓扑结构维护算法,它是成员信息传播算法高效、可靠、一致运行的基础.本文还论证了该协议具有与基于树的协议一样高的效率,因为只有自底向上的逻辑环序列才需要参与成员信息传播,而不是层次环结构中所有逻辑环.作为将来的工作,我们将扩展该协议,处理具有恶意入侵行为的 Byzantine 类型的错误.与本文基于崩溃类型这种错误类型相比,基于 Byzantine 错误类型的研究将更具挑战性.

References:

[1] Fasbender A, Reichert F, Geulen E, Hjelm J, Wierlemann T. Any network, any terminal, anywhere. IEEE Personal Communications, 1999,6(2):22−30.

[2] Varshney U. Multicast support in mobile commerce applications. IEEE Computer, 2002,35(2):115−117.

[3] Dutta A, Chennikara JM, Chen W, Altintas O, Schulzrinne H. Multicasting streaming media to mobile users. IEEE Communications Magazine, 2003,41(10):81−89.

[4] Dutta A, Schulzrinne H. MarconiNet: Overlay mobile content distribution network. IEEE Communications Magazine, 2004,42(2): 64−75.

[5] Shi ML, Xiang Y. Group communication. Journal of China Institute of Communications, 1998,19(1):45−53 (in Chinese with English abstract).

[6] Pan JP, Gu GQ. Model of group communications and projection for transport protocols. Journal of Software, 1998,9(8):574−579 (in Chinese with English abstract).

[7] Wang G, Cao J, Chan KCC. A fault tolerant group communication protocol in large scale and highly dynamic mobile next-generation networks. IEEE Trans. on Computers, 2007,56(1):80−94.

[8] Deering S, Cheriton DR. Multicast routing in datagram internetworks and extended LANs. ACM Trans. on Computer Systems, 1990,8(2):85−110.

[9] Castro M, Druschel P, Kermarrec AM, Rowstron AIT. Scribe: A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in Communications, 2002,20(8):1489−1499.

[10] Jin ZQ, Xiang XJ, Chen PP. On-Demand branching multicast. Journal of Software, 2003,14(3):553−561 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/553.htm

[11] Dolev D, Malki D. The transis approach to high availability cluster communication. Communications of the ACM, 1996,39(4): 64−70.

[12] Anastasi G, Bartoli A, Spadoni F. A reliable multicast protocol for distributed mobile systems: Design and evaluation. IEEE Trans. on Parallel and Distributed Systems, 2001,12(10):1009−1022.

[13] Rajagopalan B, McKinley PK. A token-based protocol for reliable, ordered multicast communication. In: Proc. of the 8th Symp. on Reliable Distributed Systems. Seattle: IEEE, 1989. 84−93. http://ieeexplore.ieee.org/

[14] Babaoglu O, Schiper A. On group communication in large-scale distributed systems. In: Proc. of the 6th Workshop on ACM SIGOPS European Workshop: Matching Operating Systems to Application Needs. New York: ACM Press, 1994. 17−22. http://portal.acm.org/

[15] Amir Y, Stanton J. The spread wide area group communication system. Technical Report, CNDS 98-4, Baltimore: The Johns Hopkins University, 1998.

[16] Anker T, Chockler GV, Dolev D, Keidar I. Scalable group membership services for novel applications. In: Mavronicolas M, Merritt M, Shavit N, eds. Proc. of the Networks in Distributed Computing. Providence: American Mathematical Society, 1998. 23−42.

[17] Keidar I, Sussman J, Marzullo K, Dolev D. Moshe: A group membership service for WANs. ACM Trans. on Computer Systems, 2002,20(3):191−238.

[18] Chen W, Toueg S, Kawazoe Aguilera M. On the quality of service of failure detectors. IEEE Trans. on Computers, 2002,51(5): 561−580.

[19] Sun LM, Liao Y, Wu ZM. A hierarchy reliable mobile multicast algorithm based on mixed acknowledgement mechanism. Journal of Software, 2002,15(6):908−914 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/15/908.pdf

[20] Wu Q, Wu JP, Xu K, Liu Y. A survey of the research on IP multicast in mobile Internet. Journal of Software, 2003,14(7): 1324−1337 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/1324.htm

[21] Wang SL, Hou YB, Huang JH, Huang ZQ. Dynamic range-based mobile multicast protocol. Chinese Journal of Computers, 2005, 28(12):2096−2102 (in Chinese with English abstract).

[22] Acharya A, Badrinath BR. A framework for delivering multicast messages in networks with mobile hosts. ACM/Kluwer Mobile Networks and Applications, 1996,1(2):199−219.

[23] Brown K, Singh S. RelM: Reliable multicast for mobile networks. Computer Communications, 1998,21(16):1379−1400.

[24] Brewer EA, Katz RH, Chawathe Y, Gribble SD, Hodes T, Nguyen G, Stemm M, Henderson T, Amir E, Balakrishnan H, Fox A, Padmanabhan VN, Seshan S. A network architecture for heterogeneous mobile computing. IEEE Personal Communications, 1998, 5(5):8−24.

[25] Zahariadis TB, Vaxevanakis KG, Tsantilas CP, Zervos NA, Nikolaou NA. Global roaming in next-generation networks. IEEE Communications Magazine, 2002,40(2):145−151.

[26] Gustafsson E, Jonsson A. Always best connected. IEEE Wireless Communications, 2003,10(1):49−55.

[27] Kellerer W, Vogel HJ. A communication gateway for infrastructure-independent 4G wireless access. IEEE Communications Magazine, 2002,40(3):126−131.

[28] Tamura T, Takahashi T, Morita T, Ohtaki K, Takeda H. IMT-2000 core network node systems. IEEE Wireless Communications, 2003,10(1):15−21.

[29] Buddhikot MM, Chandranmenon G, Han S, Lee YW, Miller S, Salgarelli L. Design and implementation of a WLAN/CDMA2000 interworking architecture. IEEE Communications Magazine, 2003,41(11):90−100.

[30] Perkins C. IP mobility support. IETF RFC 2002, 1996.

[31] NS-2. http://www.isi.edu/nsnam/ns/

附中文参考文献:

[5] 史美林,向勇.群组通信研究.通信学报,1998,19(1):45−53.

[6] 潘建平,顾冠群.群组通信模型及运输协议映射.软件学报,1998,9(8):574−579.

[10] 金志权,项晓晶,陈佩佩.按需分枝组播.软件学报,2003,14(3):553−561. http://www.jos.org.cn/1000-9825/14/553.htm

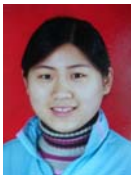
[19] 孙利民,廖勇,吴志美.基于混合应答机制的层次型可靠移动组播算法.软件学报,2002,15(6):908−914. http://www.jos.org.cn/1000-9825/15/908.pdf

[20] 吴茜,吴建平,徐恪,刘莹.移动 Internet 中的 IP 组播研究综述.软件学报,2003,14(7):1324−1337. http://www.jos.org.cn/1000-9825/14/1324.htm

[21] 王胜灵,侯义斌,黄建辉,黄樟钦.基于动态范围的移动组播协议.计算机学报,2005,28(12):2096−2102.

王国军(1970－),男,湖南长沙人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为移动计算,可信计算,软件工程.

吴敏(1963－),男,博士,教授,博士生导师,主要研究领域为过程控制,鲁棒控制,智能系统.



周薇(1980－),女,博士生,主要研究领域为网络与信息安全,群组通信.

贺建飚(1964－),男,博士生,副教授,主要研究领域为计算机网络,智能识别与智能信息处理,嵌入式系统,移动机器人.

陈松乔(1940－),男,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程及应用,算法及网络优化.