

ADLE: 一种 Ad hoc 网络分布式领导者选取算法

王 征^{1,2}, 刘心松², 李美安²

(1. 西南财经大学经济信息工程学院, 四川成都 610074; 2. 电子科技大学 8010 研究室, 四川成都 610054)

摘 要: Ad hoc 网络的动态拓扑结构和节点自组织给分布式算法的实现带来了诸多困难. Ad hoc 网络分布式领导者选取算法: ADL. 该算法基于广播机制, 采用逻辑时戳保证消息的时序性, 避免了节点饿死. 同时, 它通过缩小算法执行范围降低了消息复杂度, 而且不需要节点了解系统中所有节点的信息, 因而能够适应 Ad hoc 网络的动态拓扑结构和节点频繁出入. 分析与仿真结果表明该算法具有较低的消息复杂度、小响应延迟和公平性.

关键词: 分布式; 领导者选取算法; Ad hoc; 逻辑时戳

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2007) 11-2232-04

ADLE: A Distributed Leader Election Algorithm of Ad hoc Networks

WANG Zheng^{1,2}, LIU Xin-song², LI Mei-an²

(1. CCSE, Southwestern University of Finance and Economics, Chengdu, Sichuan 610074, China;

2. 8010 R&D CCSE, University of Electronic Science and Technology, Chengdu, Sichuan 610054, China)

Abstract: Ad hoc networks possess dynamic topologies and self-organized nodes, which hinders the implementation of distributed mutual exclusion algorithms. Aiming at the comparatively lagged level of the distributed mutual exclusion algorithms for Ad hoc networks, an adaptive algorithm for Ad hoc networks was presented as ADLE algorithm. Based on broadcast communication, the novel algorithm guaranteed the time sequence and prevents nodes from starvation by the logical timestamps. Furthermore, it decreased the message complexity and the delay complexity through restricting the algorithm executing ranges. And the nodes did not initially need knowledge of all the others in ADLE algorithm, which adapted it to the dynamic topology structures and the frequent node arrival/departure in Ad hoc networks. Analysis and simulation results show that it has lower message complexity, shorter response delay and better fairness than the traditional algorithms.

Key words: distributed; leader election algorithm; Ad hoc; logical timestamp

1 引言

分布式领导者选取是 Ad hoc 网络中的一个重要问题: Ad hoc 分布式系统中的进程往往需要并发的访问一个共享资源; 如果该资源只能被一个进程使用, 则各个进程中访问该资源的代码段被称为临界区(CS)^[1]. 研究 Ad hoc 中的分布式互斥算法有相当实际的意义, 例如在自愈雷场系统^[2~6]进一步研究中, 未来的智能地雷将采用主动触发/攻击模式: 当目标进入雷场时, 智能地雷不再被动等待触发, 而主动探测/攻击目标. 此时, 如若不采用分布式互斥算法, 将会导致多个地雷主动攻击探测到的同一个目标. 本文将详细讨论此类系统中的 Ad hoc 分布式领导者选取算法.

分布式互斥算法是领导者选取的一种解决方案^[4~6]. 本文在详尽分析 Ad hoc 的分布式互斥问题的基

础上, 提出了一种基于 Ad hoc 网络的分布式领导者选取算法 ADLE.

2 传统算法与 Ad hoc 分布式互斥系统模型

传统分布式互斥算法通常不是针对 Ad hoc 网络提出的, 因而并未考虑 Ad hoc 网络的特性. 本节将研究传统算法及其模型的缺陷, 提出用于 Ad hoc 分布式互斥的系统模型.

2.1 传统算法

分布式互斥算法可分为非令牌算法和令牌算法两种^[3]. 非令牌算法中, Lamport^[2]的全局逻辑时戳算法被广泛的应用在 Ad hoc 网络中. Maekawa^[3]提出了局部的基于请求集(Quorum)的算法, 每个节点需要维护固定长度 $\log(N)$ 的请求/应答队列, 该请求集生成计算量很大; 同时, 一旦 Ad hoc 系统中的节点出入网络, 原有请

求集就会被破坏,所有节点需要生成新的请求集,这对于计算能力有限,节点频繁出入的 Ad hoc 网络是不能容忍的.因此上述算法并不适合 Ad hoc 网络.

基于令牌的分布式互斥算法根据令牌的不同获取方法,令牌算法可以分为令牌查询(token-asking)和逻辑环(logical-ring)两种;前者中的进程没有固定的组织结构,这种算法的代表有 Chandy 算法, Ricart & Agrawala 算法,此类算法中,节点仍然需要维护大小为 N 的请求队列;后者在逻辑上将进程组织成为环型结构,令牌持续在这个虚拟环中传送,获得令牌的节点可以进入临界区.

目前,传统的 Lamport 算法仍然是 Ad hoc 分布式互斥算法的主流;针对 Ad hoc 网络提出的分布式互斥算法只有不多的几种,具有代表性的算法有: Jiang^[4]、Walter^[6] 算法等是非令牌算法的改进; Chen^[5] 采用令牌查询与逻辑环结合;但是这些算法仍然需要每个节点不断更新/获取全局信息,消耗了 Ad hoc 网络有限的带宽,因此迫切的需要一种能够适应 Ad hoc 网络的分布式互斥算法解决上述问题.

2.2 系统模型

本文 Ad hoc 分布式互斥的系统模型定义如下:

节点/进程定义:系统中有 N 的进程(假设每个节点仅有一个进程;因此,本文中的进程和节点的含义等价),进程标识为 $\{P_i | N-1 \geq i \geq 0\}$;单个进程使用一次临界资源的时间为 E .

本文中请求 R_i 的优先级定义为 $Pr(R_i)$, Ad hoc 分布式互斥模型中的定义如下:

定义 1 并发(Concurrent), R_i 和 R_j 是并发的,当且仅当 R_i 在 R_j 的生存周期内产生或者 R_j 在 R_i 的生存周期内产生,这个定义是对称的.

定义 2 并发集 ConcurrentSet($Cset$): $Cset_i = \{R_j | R_j$ is concurrent with $R_i\} \cup \{R_i\}$

定义 3 并发集首, ConcurrentHeader($Cheader_i$): R_i : $Pr(R_i) > Pr(R_j) \forall R_j \in Cset_k, i \neq j$

定义 4 并发集尾, ConcurrentTailer($Ctailer_i$): R_i : $Pr(R_i) < Pr(R_j) \forall R_j \in Cset_k, i \neq j$

定义 5 前驱 $Pred-(R_i, Cset_k) = R_j$ iff $R_j \in Cset_k \wedge Pr(R_i) < Pr(R_j) \wedge \neg (\exists R_k \in S | (Pr(R_i) < Pr(R_k) < Pr(R_j)))$

定义 6 后继 $Succ-(R_i, Cset_k) = R_j$ iff $R_j \in Cset_k \wedge Pr(R_i) > Pr(R_j) \wedge \neg (\exists R_k \in S | (Pr(R_i) > Pr(R_k) > Pr(R_j)))$

另外,从文献[5]引入两个全局视图 GV (Global View) 和 $Dist$ 度量两个请求 R_i 和 R_j 之间的请求及其数量.

定义 7 $GV(R_i, R_j) = \{R_k | Pr(R_i) \geq Pr(R_k) \geq Pr$

$(R_j)\}$

定义 8 $Dist(R_i, R_j) = |GV(R_i, R_k)| - 1$

最后,为了度量单个消息在 Ad hoc 网络中存储/转发带来的负载,引入跳数 $Hop(P_i, P_j)$ 这一概念.

定义 9 跳数 $Hop(P_i, P_j)$ 表示 P_i 进程以点到点方式传输一个消息到 P_j 进程时,该消息被转发的次数.值得注意的是, Hop 不是一个对称的概念:由于 Ad hoc 的拓扑结构是动态的,因此 $Hop(P_i, P_j)$ 不一定等于 $Hop(P_j, P_i)$.

定义 10 杀伤区域 $FireRang(X, Y)$ 表示雷场中特定的杀伤地域.

3 ADLE 算法

本节对 Ad hoc 分布式互斥问题进行分析,提出 ADLE 分布式互斥算法的主要思想,并且给出具体的算法.

3.1 ADLE 的主要思想

ADLE 算法基于广播机制,并采用消息复用机制构建动态队列,减少了构建全局队列或仲裁集所带来的高额系统开销:当某节点发现有目标进入特定杀伤区时,则广播请求消息(该消息包含了时戳以及杀伤区标志);其余节点在转发该消息时,需要判断目的节点是否能够杀伤该区域内的目标,如果不行,则停止向其转发消息(DARPA 的实例中,由于点到点通信距离大大超过杀伤范围的半径,达到 10 比 1;因此拥有同一杀伤区的任意节点可进行直连通信);由于该杀伤区可能被多个节点监测,如果这些节点未接到请求该杀伤区的消息,它们均将作上述操作;发送并接到了请求消息的节点,将请求不断插入请求消息队列,并向队列优先级最高的节点发送应答,直到接受到应答或者延迟超限;接受到应答的节点,不断进行应答计数,当应答数超过请求队列一半时,该节点进入临界区,向目标发起攻击,详细描述见下文.

3.2 算法描述

ADLE 算法采用的数据结构和消息结构类似 Lamport 算法,具体格式参见文献[2];针对自愈雷场的实际情况,引入 Timer 时间触发器以处理网络分裂等情况.

ADLE 算法仍然采用采用消息/事件驱动方式描述.该算法中的过程名按照“On_EventName”形式定义, EventName 代表被触发的事件名; P_i 节点中描述事件的伪代码如下:

Procedure On_ReqCS; // P_i 请求进入临界区

Send Request(i) to the nodes monitoring the FireRang [x, y];

//发送 Request 消息给监测 FireRang [x, y] 杀伤区域的节点(包括自身);在此及下文中均隐去 Lamport 时

戳的处理与比较;同时请求消息携带 $\text{FireRang}[x, y]$ 标志信息,以区分不同的互斥对象

Initialize Timer; //初始化时间触发器,以处理网络分裂等情况

End;

Procedure On_RecRequest; // P_i 接到 P_j 请求

If Self can't destroy $\text{FireRang}[x, y]$ then Lose Request (j);

//如果 P_i 不能处理 $\text{FireRang}[x, y]$,则抛弃该请求

Insert Request(j) into RQ_i ; //将请求插入队列

If Request(j) is the Cheaderi of $Cseti$ then Send Reply (j, i) to P_j ;

//如果该请求优先级最高则发送应答

End;

Procedure On_RecReply; // P_i 接到 P_j 应答

Compare Reply(i, j) with $RQ_i[j]$;

If all $RQ_i[j]$ have Reply(i, j) then Enter the CS of $\text{FireRang}[x, y]$;

//如果对于该队列中所有请求,节点都接收到了给自己的应答,则进入临界区

End;

Procedure On_Timer; // P_i 计时器事件

If $\text{FireRang}[x, y]$ is living then Enter the CS;

//如果限定时间内,目标没有被消灭,即可能发生了网络分裂等情况,需要作特殊处理;则 P_i 进入临界区,攻击该杀伤区中的目标

End;

注意,本算法没有使用 Release 消息,其原因是自愈雷场中的分布式互斥对象具有一次性特征;通常,配有大威力战斗部的节点能够在一次性毁伤目标,因此没有必要使用 Release 消息释放该对象;当新的目标进入该杀伤区时,将被视为新的分布式互斥对象。

由于确定监测同一目标的节点集后,ADLE 的操作过程与 Lamport 算法类似,因此该算法也具有 Lamport 算法的非死锁等优良特性,算法的证明也与之类似,限于篇幅,请读者参见文献[2]。

4 算法分析、实例与仿真

4.1 算法分析与实例

ADLE 算法的最大特点在于:

(1)请求队列的动态生成:动态队列是自愈雷场应用环境的要求;较之传统分布式算法,ADLE 算法中的请求队列无需通过节点监测生成,因而在不执行互斥操作时并不占用额外的系统资源;ADLE 算法运行时,监测同一临界对象的节点集在一定时间内保持稳定;该节点集中的节点同时发送请求;节点通过判断杀伤

区坐标来确定是否参与互斥操作,动态队列的使用控制了算法运行范围,降低了消息复杂度,缩短了响应延迟。

(2)互斥操作局部化:传统算法的互斥操作在全局范围内进行,而 ADLE 算法根据自愈雷场自身的特点,将互斥操作的范围控制在能够攻击同一目标的节点集,从而不必进行全局操作,节省了系统开销;同时,通过地理坐标杀伤区化的方法使得系统可以区分不同的互斥对象,同时处理多个进入雷场的目标,提高了系统的并发度。

ADLE 算法的实例如下:

在图 1 的雷场中,一个目标进入了 P_1 、 P_2 和 P_3 的公用杀伤区内,则此三个节点执行 ADLE 算法,进行互斥操作; P_1 、 P_2 和 P_3 相互协商,最终将由 P_2 发动攻击, P_1 和 P_3 发送应答给 P_2 。另外,系统中其他节点也接收到 P_2 等节点发送的请求消息,如 P_6 节点,当它从请求消息中取出 $\text{FireRang}[x, y]$ 杀伤区域字段进行判断后,发现自己无法攻击该目标,则抛弃该消息,并不进行处理或者转发;由此可以看出互斥操作局部化的方法,使得 ADLE 算法的执行范围大大缩小,节省了系统开销。

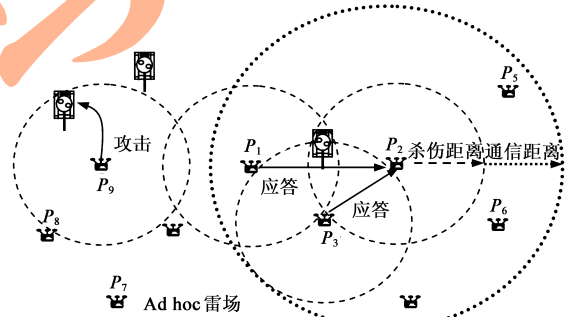


图 1 ADLE 算法实例

4.2 仿真结果

仿真试验设定单个节点通信半径 300m,杀伤范围 100m,节点随机散布;单个节点移动速度最大为 $3m/T$; Ad hoc 雷场网络的为 $800 \times 800m^2$ (节点间一跳的通信时间和攻击单个目标的平均使用时间均为 T). 尽管目标可以随时出现在雷场各处,平均出现时间间隔为 $120T$,

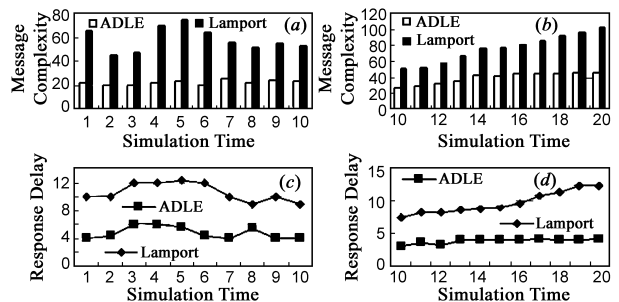


图 2 ADLE 基本性能仿真结果

目标平均存在时间为 $600T$; 仿真结果如图 2、3。

针对 ADLE 的平均消息数仿真分析主要进行了两组, 第一组在动态拓扑的情况下, 统计 20 个节点中进行互斥的需要的平均消息数, 如图 2(a); 第二组在节点数量变化, 固定拓扑结构的情况下, 统计互斥需要的平均消息数, 如图 2(b)。

针对 ADLE 的平均延迟数仿真分析主要进行了两组, 第一组在节点移动的情况下, 统计 20 个节点中进行互斥的需要的平均延迟数, 如图 2(c); 第二组在节点数量变化(从 10 逐次到 20), 固定拓扑结构的情况下, 统计互斥需要的平均延迟数, 如图 2(d)。从图 2 中可以看出, ADLE 算法的改进措施将分布式互斥操作控制在较小的范围内, 从而使该算法具有较短的响应延迟以及很低的消息复杂度。

图 3 列出了该雷场在没有领导者选取算法 (None LE) 和运行 Lamport 算法以及 ADLE 算法时, 突破该雷场所需的目标数量(突破定义为雷场分裂无法重组或者新目标进入雷场无法攻击); 与前两者相比, ADLE 算法凭借局部互斥操作等方法, 取得了较高的毁伤率。

5 结论

本文提出了一种新的分布式领导者选取算法 ADLE。该算法通过缩小执行范围提高了性能; 同时该算法较之传统算法, 不需要单个节点拥有其它节点的信息, 能够适应 Ad hoc 网络节点频繁出入等特性。理论分析表明该算法具有较好的性能。

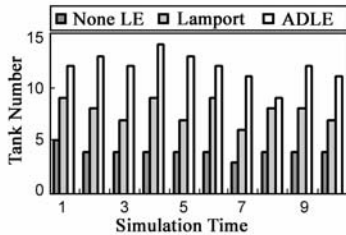
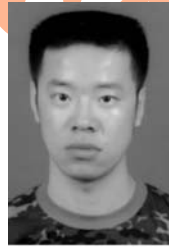


图 3 Ad hoc 雷场毁伤效果仿真结果

参考文献:

- [1] Yen Li-Hsing, Chi Kuang-Hwei. Maintaining a ring structure for mobile ad hoc computing[J]. Journal of Parallel and Distributed Computing, 2004, 64(12): 1371 - 1379.
- [2] L Lamport. Time, clocks and the ordering of events in distributed systems[J]. Comm ACM, 1978, 21(7): 558 - 565.
- [3] M Maekawa, A logN algorithm for mutual exclusion in decentralized systems[J]. ACM Trans Computer Systems, 1985, 3(2): 145 - 159.
- [4] Jehn-Ruey Jiang. A distributed h-out of-k mutual exclusion algorithm for ad hoc mobile networks[A]. Proceeding of the international Parallel and Distributed Processing Symposium (IPDPS'02)[C]. Durham, NC USA, 2002.
- [5] Chen Yu, Welch, Jennifer L. Self-stabilizing dynamic mutual exclusion for mobile ad hoc networks[J]. Journal of Parallel and Distributed Computing, 2005, 65(9): 1072 - 1089.
- [6] J Walter. A mutual exclusion algorithm for ad hoc mobile networks [J]. Wireless Networks: 2001, 7(9): 595 - 600.

作者简介:



王 征 男, 1979 年生, 2007 年毕业于电子科技大学, 博士, 现为西南财经大学经济信息工程学院教师. 主要研究方向为分布式算法等.
E-mail: wangzheng151400@163.com

刘心松 男, 1940 年生, 教授, 博士生导师. 主要研究方向为分布式并行等.

李美安 男, 1973 年生, 博士生. 主要研究方向为宽带网络.