

## 基于自适应周期的流言机制快速构建自组 Overlay 拓扑<sup>\*</sup>

孙晓<sup>+</sup>, 王晖, 汪浩, 姜志宏, 陶钧

(国防科学技术大学 信息系统与管理学院, 湖南 长沙 410073)

### Enabling Quick Overlay Topology Construction Using Adaptive Cycle Gossip

SUN Xiao<sup>+</sup>, WANG Hui, WANG Hao, JIANG Zhi-Hong, TAO Jun

(College of Information Systems and Management, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: xsunxiao@hotmail.com

**Sun X, Wang H, Wang H, Jiang ZH, Tao J. Enabling quick overlay topology construction using adaptive cycle gossip. *Journal of Software*, 2008,19(9):2422-2431. <http://www.jos.org.cn/1000-9825/19/2422.htm>**

**Abstract:** Based on previous observation, a dynamic adaptive cycle into gossip-based topology management is introduced to replace the traditional fixed cycle and a quick topology convergence method based on adaptive cycle is also proposed. In this method, those nodes featured with local topology stability send fewer gossip packets, while nodes in frequently changing environment send more packets. This dynamic adaptive method improves data exchange efficiency, saves network resources, enables faster local data exchange, and consequently speeds up the overall topology convergence. In detail, by using logistic curve as the basic control function of the adaptive cycle, rules for accidental events is defined accordingly. Simulation is presented to show the validity of this approach and that it is especially suitable for dynamic network environment.

**Key words:** overlay; gossip; self-organization; converge; adaptive

**摘要:** 分析了 overlay 拓扑管理中流言机制的一般过程,发现固定周期的流言报文中存在的数据交换盲目性的弊端.为此,引入动态的自适应周期来代替固定周期,使得局部拓扑稳定的节点较少发出流言报文,而局部拓扑动荡的节点较多发出流言报文.这种方法提高了数据交换效率,节省了网络资源,允许在局部加快数据交换,从而提高拓扑的整体收敛速度.通过仿真实例对该方法的有效性进行了验证,并证实该方法在动态网络环境中尤其适用.

**关键词:** overlay;流言;自组织;收敛速度;自适应

**中图法分类号:** TP393      **文献标识码:** A

当代互联网络规模巨大、组成复杂、拓扑结构动态性加强,造成整个网络的可靠性降低,不可预见的网络故障随时都有可能发生,已经接近传统技术所能有效管理的极限.反观生物、社会等自然网络系统,具有更大的复杂性,在面对自然灾害和突发事件时,却表现出很强的适应性.这些适应性的本质就是网络节点的自组织、自修复能力.借鉴复杂适应系统(complex adaptive system,简称 CAS)理论,在当前的网络应用系统中采用自组织原理构建 overlay 网络拓扑,能够更好地应付动态变化的网络环境.那么,以这种自组织 overlay 拓扑结构为基础的网络服务,势必具有更强的可用性<sup>[1]</sup>.

<sup>\*</sup> Supported by the Natural Science Foundation of Hu'nan Province of China under Grant No.07JJ4018 (湖南省自然科学基金)

Received 2006-07-01; Accepted 2007-06-18

Overlay 网络是构建在已有网络之上的一个由节点和逻辑链路构成的虚拟拓扑结构,它基于底层网络基本的传输功能而建立,通过一个虚拟的网络拓扑将网络重新组织,其目的是提供单纯依靠网络底层所不能完成的功能<sup>[2]</sup>。Overlay 网络的拓扑结构以及构造模式是 overlay 网络的核心,其组织形式通常都是为达到某种特定网络服务的性能而建立的,并极大地影响网络服务的性能。例如,为实现应用层组播而构造的树形结构、Chord 中为方便关键字查询构造的基于 DHT 的环形结构<sup>[3]</sup>以及 Pastry<sup>[4]</sup>等模型中使用的更复杂的类超立方体结构。

实际应用中的自组织 overlay 拓扑(尤其是在动态网络环境中)应该具有组织快速、灵活的特点,当网络状况发生变化时,能够及时对物理网络的变化做出反应,构建合理的拓扑结构,以增强网络服务的可用性。从适应性的角度而言,拓扑变换需要的时间越短,效率越高,overlay 拓扑的适应性就越强。

## 1 相关研究及本文工作

流言(gossip)机制是在网络中进行信息传播的一种方法,也称为流行病算法(epidemic algorithm)。最早,流言机制起源于 20 世纪 70 年代对传染病在人群网络中传播的时间、范围等特性的研究,其研究的最初目的是为了研究疾病传播的特性,为传染病的防治起到一些启发作用。在一系列的研究过程中发现,传染病在特定条件下,可以很快传遍整个网络。因此,近年来,越来越多的研究将这种流行病的传播机制用于网络中有效信息的传输和收集<sup>[5]</sup>。

文献[6,7]使用了 gossip 机制进行概率可靠广播,文献[8]使用 gossip 机制构造了一个 P2P 成员管理模型。此类系统中,在 gossip 机制的帮助下,可以实现迅速而有效的信息分发。其他方面的应用,如使用 gossip 进行故障探测<sup>[9]</sup>、网络数据的聚合、资源的发现和监控<sup>[10]</sup>、数据复制管理<sup>[11]</sup>等。相对于基于 DHT 的结构化 P2P 网络管理模式,基于 gossip 机制的非结构化模式更能适应动态变化的网络状况。在网络状况发生剧烈变化时,结构化网络频繁进行关键字迁移和拓扑重构的开销很大,而基于 gossip 机制的网络只需要更新局部的邻居列表,就可以适应新的网络状况。

所谓的非结构化网络并非是没有固定结构的,gossip 机制的自由转发功能,也可以作为自组织出具有某种特性的网络拓扑的有效手段。在 BISON 项目组进行的研究中,基于 gossip 机制制定了一种 overlay 拓扑管理协议 T-man<sup>[12]</sup>,该协议可以构造出具有不同特性的 overlay 拓扑,并具有很好的可伸缩性和健壮性,能够在动态网络环境中取得很好的性能。BISON 中使用该协议在此基础上进行了分布式系统中一系列拓扑组织和管理工作,文献[12]讨论了构造树型、环型网络的基本方法,文献[13]构造了能够及时反映网络状况的 Chord 环结构。

Gossip 机制本质上是一种受控制的宏泛,其目的是快速、有效地将信息在网络中进行分发。因此,对于 gossip 机制本身的优化,主要集中在控制流言的规模、传播路径,提高流言报文的效率上。文献[14]通过采用一种适应性策略,使得节点倾向于与度数高的节点相连,并对交换的报文进行流量控制,以改善查询性能。文献[15]利用 small-world 特性,对非结构化 Freenet 网络的路由表进行改进,提高其路由性能。文献[16]引入一部分相对能力较强的节点作为 ultra-peer(超级节点)承载更多的计算工作,以加快路由和查询的速度。文献[17]以缩短报文的 TTL(time-to-live)来限制对同一 ultra-peer 的重复查询,提高查询效率,并减少无谓的网络资源浪费。文献[18]解决了非结构化网络中的 overlay 拓扑和底层物理网络的匹配问题,减少 gossip 报文在部分节点之间的重复发送。

本文的主要工作是针对 gossip 机制作为拓扑管理手段时,如何掌握 gossip 报文发送的时机,提高 gossip 报文的效率。使用 gossip 进行的拓扑管理是利用 gossip 报文交换网络状态信息,构建一个具有某种特性的网络拓扑。此时,一个很关键的性能指标就是拓扑收敛所需要的时间,也就是自组织拓扑的收敛速度。收敛速度表示网络拓扑自组织到一个相对优化状态所需要的时间的快慢。收敛速度越快,网络拓扑能够更快地对环境的变化做出合理的反应,网络的适应性越强。而在协议一定的情况下,单位时间内网络上交换的流言报文数量越多,交换的信息量越大,拓扑收敛的速度可能越快。因此,协议交换的数据量和拓扑的收敛速度是两个相互矛盾的指标。但是,单位时间内交换的报文数量受网络资源和节点能力的限制,不能无限增加。

由于自组织机制没有全局统一的管理,采用的是自由的随机数据交换,这种方式保证了协议具有良好的适应性,但是也带来了数据交换的盲目性。网络中交换的 gossip 报文对拓扑收敛产生的作用是不一样的,并不是所

有被交换的 gossip 报文都是有意义的.局部拓扑不稳定的节点,需要交换更多的报文,获得更多、更新的网络状态信息;而局部拓扑相对稳定的节点,对信息的需求较少.因此,有可能基于节点的差异性对 gossip 报文进行优化.

目前现有的研究很少考虑到这种节点的差异性.本文通过分析自组织拓扑收敛过程中节点的这种差异性,采用根据节点活动时间产生的自适应报文交换周期代替固定周期的方法,尽可能地减少数据交换中无用信息的处理,从而可能在局部加快数据交换速度,进而加快整个自组织 overlay 拓扑的收敛速度.

本文第 2 节介绍基于 gossip 机制自组织地进行 overlay 拓扑构建的基本原理,分析节点差异性可能造成的无用数据交换.第 3 节提出基于自适应周期进行数据交换的方法.第 4 节进行算法的仿真验证并分析实验结果.本文最后做出总结并对后续相关研究工作进行展望.

## 2 问题分析

### 2.1 基于Gossip机制的拓扑管理

基于 gossip 机制构建 overlay 拓扑就是对于一个给定的随机连接的 overlay 网络,通过以一定的自组织规则改变网络节点之间的连接关系,构造一个具有一定结构特性的拓扑结构.以 BISON 的 T-man 协议中给出的 gossip 控制协议为例,描述其信息交换的一般方法:每个节点上拥有一个 cache 保存其邻居节点的视图(view),整个网络依靠节点彼此的邻居视图维持着整个 overlay 网络的拓扑结构,节点之间每隔  $T$  时间,通过随机选择邻居节点两两交换 cache 中的信息,并以一定的取舍规则更新自己的 cache,形成新的邻居视图,以保证其拓扑结构进行演化<sup>[12]</sup>.该交换算法的描述见表 1,节点上都有主动和被动两个处理线程,节点以周期  $T$  主动发起数据交换,这两个线程都可以导致节点的邻居视图发生变化.

Table 1 The skeleton of Gossip-based protocol

表 1 Gossip 协议框架

Active thread	Passive thread
<pre>do forever   wait(T) // wait T   Peer=myview.slectPeer() // select a neighbor   Send (Peer,view) // send the view in local cache   PeerView=receive() //receive remote view   update(view,PeerView,R-Policy)   //update local view</pre>	<pre>do forever   PeerView=receive() //receive remote view   Send (Peer,view) //send local view   update(view,PeerView,R-Policy)   //update local view</pre>

另外,就 T-man 协议而言,该拓扑控制协议主要包括了以下几个基本参数<sup>[12]</sup>:

- (1) 网络节点的规模  $N$ ,好的自组织协议对网络规模的大小并不敏感.
- (2) 每个节点上的 cache 大小  $c$ ,即每个节点保存的邻居节点信息的数量.Cache 中保存邻居节点的视图.
- (3) 邻居关系优劣程度的评判函数  $R$ ,以此评判函数将邻居节点进行排列并进行取舍. $R$  函数的不同,可以自组织出不同类型的网络.文献[12]中介绍了自组织出环形、树形等结构的  $R$  函数取值方法.

### 2.2 问题分析

图 1 显示在 gossip 报文控制下,overlay 拓扑收敛的仿真曲线.当某节点经过一段时间的数据交换后,节点邻接关系相对稳定,其邻居视图在网络状况不变时,基本不再发生变化,就认为该节点处于“稳定”的邻接状态.通过统计一个静态网络从初始化阶段开始不同时间点上网络中“稳定”节点的百分比,绘制了图 1 中的曲线.由此可以看出,网络拓扑的收敛是一个渐进的过程,节点间存在差异性,尤其是在有些场合,这种差异非常明显.

而一般的协议没有考虑节点的差异性,从而造成了数据交换效率的低下.那些达到稳定状态的节点,仍然在以不变的  $T$  为周期发起数据交换.在环境变化不大时,这些数据交换并不能造成网络拓扑的演变,这些数据交换是基本“无用”的.另一方面,新加入的或者是处于动态环境中的节点,也以不变的  $T$  为周期发起数据交换.对于这类节点,拓扑的优化是其主要任务,迫切需要提高数据交换的频率.

这里就出现了一个突出的问题,主动方发起的数据交换缺乏必要的控制机制.到达稳定状态的节点所发出的“无用”数据包,造成了一定网络带宽、节点计算能力的浪费.对于这类节点,可以增加  $T$ ,以减小报文交换的频率,减少无谓的资源消耗.稳定节点较少主动发出数据交换,还可以减少该类节点局部拓扑的剧烈变化,避免影响其正常的网络服务.另一方面,如果网络状况发生剧烈动荡,大量的节点加入或退出,通过被动的数据交换,该类节点也可以及时获取必要的网络状态信息.

而未达到稳定状态的节点,通常是新加入的节点或者原有局部网络环境发生了剧烈变化,收敛到稳定结构是其首要任务.其网络服务处于系统启动或重启阶段,正常的网络服务尚未启动或暂停,节点处于相对空闲阶段,因此可以全力进行拓扑控制报文的传输和处理,其状态允许在此类节点上以较小的  $T$  进行工作.

这就表明,网络自组织中的这种自由数据交换有着进一步优化的余地,对节点发起的数据交换的时间间隔  $T$  有必要增加合理的控制机制,以优化网络拓扑收敛质量.

### 3 算法阐述

#### 3.1 自适应周期模型

本文的主要工作就是如何为节点动态确定合理的  $T$ ,以改善 overlay 拓扑的收敛性能.由于自组织分布式网络没有中心节点来进行集中的调度,因此,采用的控制方法只能从节点自身依靠有限的局部信息来发起.本文提出了以下自适应周期模型,如图 2 所示.

在通常情况下,新加入的节点需要以较快的频率交换数据,随着节点活动时间的增加,其数据交换周期在增长函数的控制下自然增加.另一方面,一些规则被用来处理突发性事件,如当网络发生大规模故障时, $T$  在突发事件的影响下,发生跳变.

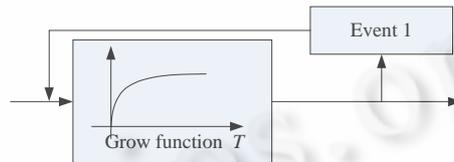


Fig.2 Basic control method of adaptive cycle model

图 2 自适应周期模型基本控制方法

#### 3.2 增长函数

节点使用增长函数确定的动态  $T$ ,保证  $T$  在正常情况下随节点活动时间的增加自然增长.在收敛曲线中可以看出,任何节点在经历一段时间(几个周期)后趋于稳定状态,因此考虑建立  $T$  和节点活动时间  $t$  的函数  $T(t)$ ,以此函数为不同的节点分配合理的  $T$ .考虑  $T$  应该有的变化趋势为:初期  $T$  的增长率较小,一段时间后增长率大幅度增加,增加到一定程度后增长减慢,到达某个限定值之后基本不再增长.令  $r$  为  $T$  的增长率,随着  $t$  的变化而变化.令  $r$  的初始值为  $r_0$ ,当  $T$  增长到一定值  $K$  停止变化,即  $r$  变为 0,则

$$r = r(t) = r_0 \left( 1 - \frac{T(t)}{K} \right) \tag{1}$$

则  $T$  和  $t$  之间的关系可以表示为

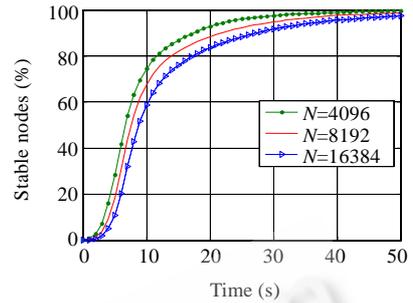


Fig.1 The convergent curve of topology

图 1 拓扑收敛曲线

$$\begin{cases} \frac{d(T)}{dt} = r_0 \left(1 - \frac{T}{K}\right) T \\ T(0) = T_0 \end{cases} \quad (2)$$

由公式(2)可以看出,所需要的  $T(t)$  方程是一条 Logistic 生长曲线,其求解后的表达方式为

$$T(t) = \frac{K}{1 + be^{-rt}}, \quad K, b, r \text{ 为常数} \quad (3)$$

生长曲线常用于对生物繁殖和生长过程的研究,是描述生物生命周期相关特性的常用方法,其函数曲线如后文图 3 所示.本文尝试以此  $T(t)$  作为控制方法中的自然增长函数产生动态  $T$ ,代替一般方法中的固定周期  $T$  来进行信息的交换,提高数据交换的效率.

### 3.3 相关应用规则和突发事件

正常情况下,节点活动时间自然增长.随着节点活动时间的增加,overlay 拓扑趋于稳定,节点以较大的  $T$  进行数据交换.当节点活动时间增加到一个较大的值时, $T$  基本不再变化.但是,单一的增长函数不足以完成应付复杂情况的功能,需要对于特别的突发事件添加专门的处理规则.在本文的自适应控制方法中,定义的突发事件规则如下:

**节点活动时间归零:**在某次数据交换后,节点的邻接关系发生较大变化,认为节点周围的环境发生剧烈变动,该部分拓扑需要重组.此时,将节点活动时间归零,该节点的活动情况类似于新加入的节点,从而以较小的  $T$  主动进行数据交换.

### 3.4 增长函数取值分析

在 gossip 机制自由交换数据的影响下,在系统中活动时间较长的节点经过多次数据交换,得到较好的邻接关系,其优化的空间不大;活动时间较短的节点,其数据交换次数较少,其邻接关系较差,需要进一步进行优化.使用自适应周期的本质,是以节点的活动时间控制其数据交换的频率,使活动时间越长的节点,主动进行数据交换的频率越低.在突发事件来临时,通过突发事件的应对规则,周围环境发生较大变动的节点通过活动时间的归零,也能获得较高的数据交换频率.

通过这种方法,系统中的节点尽可能地避免处理无用的交换数据,尽可能地使每次数据交换进行可对 overlay 拓扑的优化发生影响.在使用相同网络资源的条件下提高数据交换的有效率,提高系统自组织的效率,加快 overlay 拓扑的收敛速度.

本文考虑了两种不同取值的增长函数应用模型,如图 3 所示.首先,图中 slow cycle 增长函数的初始取值和固定周期相同,其最大周期值相当于固定周期的几倍.对于采用这种类型增长函数的应用模型,其数据交换的频率总小于或等于固定周期,它可以大幅度降低交换的数据量,但是并不会过多地降低收敛速度,所以其主要作用是节省网络资源.如图 3 所示,固定周期取值为 2s,该增长函数的最大取值为 10s,约在 60s 达到最大.在前 60s 的时间内,自适应周期的平均值约 3 倍于固定周期.假定每次交换的数据量相等,考虑前 60s 交换的总数据量,则采用自适应周期方法交换的数据只有固定周期方法的 30%.

更重要的是第 2 种应用模型,如图 3 所示的 quick cycle 曲线,其初始值略小于固定周期,并在较短的时间内超过固定周期.这种模型的网络节点在其启动初期很短的一段时间内会以较高的频率主动交换数据,然后交换周期超过固定周期,交换频率变慢.在启动初期或执行了归零规则的节点上,拓扑的优化重组是其主要工作.它们以大量资源用于频繁数据交换,“努力”地尽快融合到整个 overlay 网络中.同时,网络中活动时间较长的节点并不会受到太大的影响.这种模型中,自适应周期方法初期处理的数据量要超过固定周期,但是从总体来看,交换的数据总量仍小于固定周期方法,具体数据见第 4 节.这种模型对于动态网络环境尤其适用,这种应用策略能使系统具有一定的自适应能力.

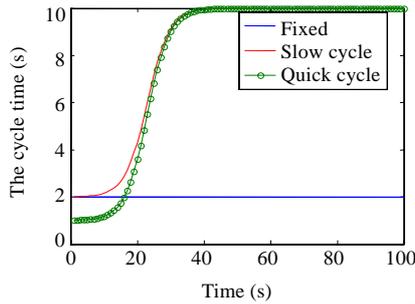


Fig.3 Two different grow functions

图3 两种增长函数取值模型

### 4 仿真

本节以仿真方法,验证算法的有效性.仿真采用 BISON 项目组提供的 PeerSim 仿真工具,并自行编写了几个协议模块,配合相关的数据分析进行仿真结果验证.

#### 4.1 仿真模型

在本文的仿真中,主要针对节点数量较多的大规模动态网络.仿真模拟了一个基于自适应周期模型的 overlay 网络拓扑的演化行为,网络节点之间的数据交换依赖于下层网络所提供的基本路由和信息传输能力.每个节点拥有其他节点的一些信息,作为邻居关系保存在自己的 cache 中.网络节点间的距离使用 Vivaldi 网络坐标系统<sup>[19]</sup>来模拟.该坐标系统通过 overlay 网络中的节点和多维空间中点的对应关系建立一个精确的几何模型,网络节点间的传输延迟通过对应的坐标空间中点的相互距离来进行模拟.

Overlay 拓扑被初始化为一个随机连接的网络,节点间通过基于 gossip 机制的数据交换寻找由 R 函数评判的较好邻居节点,观察 overlay 拓扑的收敛状况.本文仿真中,R 函数取值为节点间的最小传播时延,当节点寻找到与其距离最近的几个节点作为邻居节点时,认为节点达到“稳定”状态,节点局部的 overlay 拓扑连接达到最优.

#### 4.2 静态场景

首先仿真静态网络场景.overlay 拓扑被初始化为随机连接的网络,仿真节点数量  $N=2^{12}$ ,每个节点 cache 大小  $c=20$ .采用图 3 所示的两种应用模型和固定周期模型比较.固定周期取值 2s(图 3),仿真前 100s 的 overlay 拓扑演化状况,仿真结果如图 4 所示.

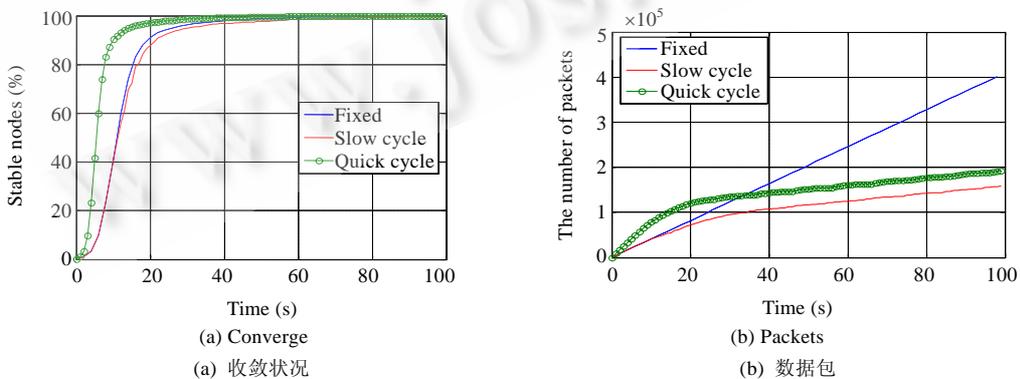


Fig.4 Static scenario

图4 静态场景

图 4(a)显示的是 3 种模型的收敛状况.根据仿真结果可以看出,在采用 slow cycle 函数的模型中,overlay 拓

扑的收敛状况和固定周期模型相比都不相上下.而对照图 4(b)的交换数据量分析,slow cycle 模型交换的数据量总小于固定周期模型,40s 已经降到固定周期模型的 50%以下.而且随着时间的推移,其交换数据的增长率也远小于固定周期模式.这种模型对收敛状况影响不大,但是节省资源的效果非常明显.

使用 quick cycle 模型,在网络启动初期交换的数据量较多,约在 35s,其交换的数据量开始小于固定周期模型,继而维持在较低的增长率水平.更为重要的是,从图 4(a)的收敛曲线来看,quick cycle 模型使系统的收敛性能更加优化,其收敛到 90% 以上的时间节省了 50% 左右.

### 4.3 动态剧变场景

自适应周期模型的应用,更重要的意义在于动态网络环境中.本节仿真了一个网络状况发生重大变化的动态场景,仿真节点数量  $N=2^{12}$ ,每个节点 cache 大小  $c=20$ .与前一场景不同的是,在 20s,网络状况发生重大变化,加入  $2^{10}$  个新随机节点,新加入节点数量相当于  $N$  的 25%.仍然采用图 3 表示的两种自适应周期模型和固定周期模型进行比较,仿真结果如图 5 所示.

对于 slow cycle 模型,在该动态场景中,其收敛状况和固定周期相差不大,主要优点还是交换的数据量少,节省了网络资源.而 quick cycle 模型在动态场景的有效性尤其明显.在发生网络剧变之前,拓扑已经收敛到接近 100%,且整个过程中的收敛百分比都高于固定周期.同时参照图 5(b),在约 50s 时,quick cycle 模型和固定周期模型交换了大致相同的数据量,而此时,quick cycle 的收敛百分比要高 10% 左右.在 50s 以后,quick cycle 交换的数据量小于固定周期,并且维持在较小的增长水平上.总体而言,相对于固定周期模型,quick cycle 模型交换了较少的数据,提高了收敛速度.

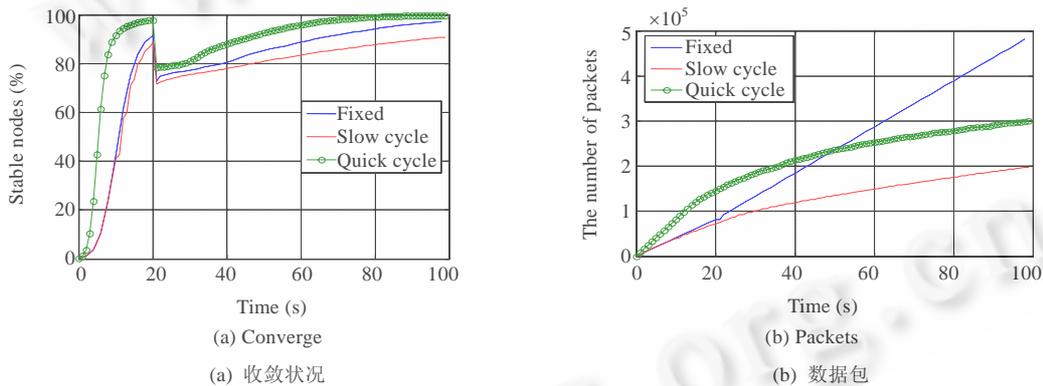


Fig.5 Dynamic shakeup scenario

图 5 动态场景

总结这两个场景的仿真结果,slow cycle 模型可以大幅度节省网络资源,quick cycle 模型可以加快收敛速度.唯一的不足似乎是 quick cycle 模型初期对资源的需求量较大,但这对于不稳定的节点来说是可以接受的.同时,这也是由于仿真中对于所有节点采用了相同的增长函数所致.通过对不同的节点根据其状况灵活采用不同参数取值的生长函数,可以进一步提高数据交换的效率.

### 4.4 波动场景中的混合策略模型

为了进一步验证自适应周期模型的有效性,我们仿真了一个存在网络 churn(波动)的动态场景.所谓 churn,就是 overlay 网络中有节点加入和退出引起的网络状况变化.根据文献[20]中对 Gnutella trace 的统计,网络中持续有节点加入和退出,也就是说,实际环境中网络的 churn 总是存在的.在 1 小时的时间里,大约 50% 的节点离开系统被其他节点所代替.我们以加入或减少一定比例的随机节点作为网络 churn 模型.根据 Gnutella trace,在 1s 的时间里发生 churn 的节点比例小于 0.02%.在我们的仿真中,采用 2% 的 churn rate(对于  $N=2^{12}$  的网络基数,约每秒 80 个节点),相比于实际网络状况,这样的 churn rate 是相当高的.

该仿真中,网络场景  $N=2^{12},c=20$ ,自第 20s 开始加入 2%的节点 churn 并一直持续.为了说明增长函数灵活取值的好处,在此仿真实例中,我们采用了前述两种增长函数的混合应用.初始的网络节点采用 slow cycle 模型,后面动态加入的节点和执行了活动时间归零规则的节点采用 quick cycle 模型.其仿真结果如图 6 所示.

如图 6(a)所示,混合策略的自适应周期模型的收敛速度快于固定周期模型.在 40s 之前,两种模型的收敛质量不相上下;在 50s 以后,自适应周期模型的收敛度持续高于固定周期模式的 5%~10%.从整体来看,自适应周期模型在收敛速度上要更快.

如图 6(b)所示,从两种模型消耗的网络资源来看,初期两者交换的数据量较接近;但是从整体上看,自适应周期模型交换的数据量一直少于固定周期模型.虽然由于网络环境动态变化带来大量的新加入节点,这些节点以较快频率交换数据,消耗资源较多,同时也可能引起部分周围环境发生重大变化的原有节点缩短周期,但是从总体上说,网络中资源消耗的总量并没有增多.前 100s,自适应周期模型交换的数据量相对于固定周期约减少了 30%.综合图 6(a)、图 6(b),自适应周期模型同时实现了提高收敛速度和降低资源消耗.

我们对该仿真实例进行了更深入的分析,图 6(c)统计了整个过程中稳定节点和不稳定节点的平均周期,可以看出,大部分时间内稳定节点的周期远高于不稳定节点,40s 以后,约是不稳定节点周期的 4~5 倍.也就是说,不稳定节点进行数据交换的平均频率的确要快很多,完全符合该自适应周期模型的预期目的.而图 6(d)、图 6(e)则取样分析了两类节点周期的具体分布情况,取样点为第 40s.可以看出,不稳定节点的周期大部分集中在短周期部分,稳定节点的周期大部分集中在长周期部分.可见,自适应周期模型可以根据节点活动时间,对两类节点做出相对明显的区分.

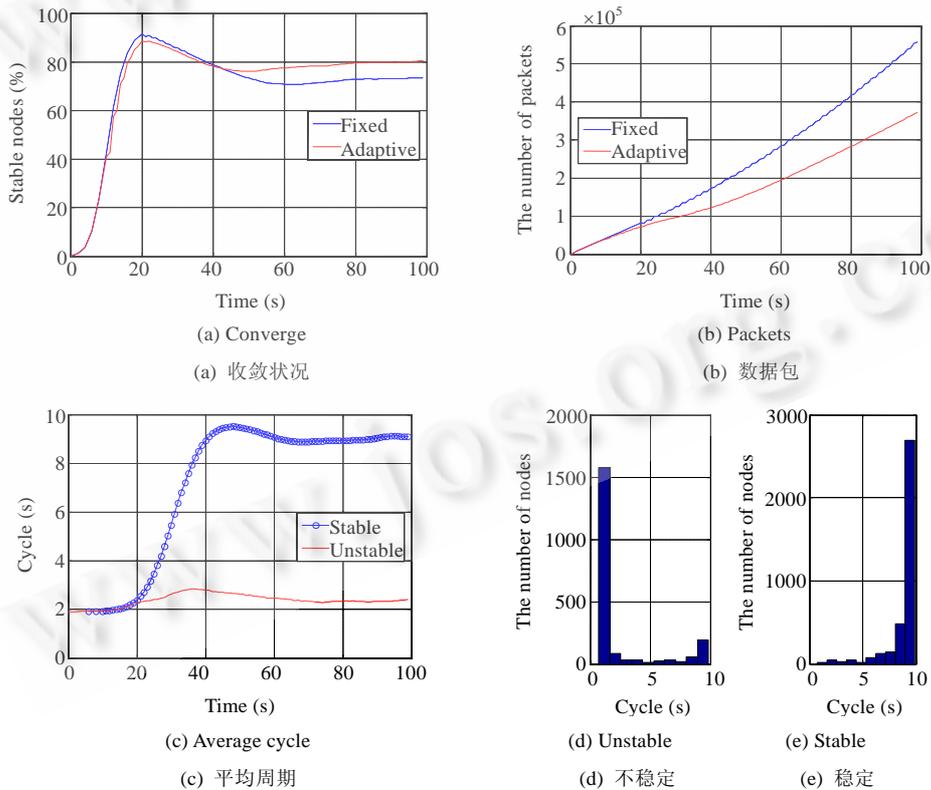


Fig.6 Mixed policy in churn scenario

图 6 混合策略应用于波动场景

## 5 总结

随着流行病传播理论研究的成熟,gossip 机制在 overlay 网络中的相关应用研究是近年来的热点.尤其是采用 gossip 机制进行 overlay 拓扑的自组织构建,是使 overlay 拓扑结构具有自适应性、能够应付复杂动态环境的有效途径.我们在现有基于 gossip 机制构造自组织 overlay 拓扑研究的基础上,提出了基于自适应周期进行数据交换的快速 gossip 机制模型.应用该模型,可以有效地避免网络中无用的数据交换信息,提高数据交换的效率,从而节省网络的带宽、计算能力等资源,并加快自组织的收敛速度.本文进行了算法分析,并利用软件工具进行了仿真验证,仿真结果证实了自适应周期模型的有效性.具体应用中,可以通过灵活选取自适应周期函数的参数以及混合运用不同的周期模型,来进一步提高应用的效率.

Overlay 网络是一个拥有广泛前景的研究领域,overlay 拓扑是这个领域的核心.其拓扑模型、分布式构建机制以及相关应用问题的研究有着极大的学术和应用价值.我们下一步的研究工作集中在自适应周期模型的合理应用以及影响 overlay 拓扑快速收敛的其他因素的研究上.

**致谢** 在此,我们向对本文的工作给予支持和建议的同行以及老师和同事表示感谢.同时感谢 Bologna 大学的 Mark Jelasity 博士和 Ozalp Babaoglu 博士,他们对我们研究中遇到的问题提供了无私的帮助.

## References:

- [1] Babaoglu O, Canright G, Deutsch A, Di Caro G, Ducatelle F, Gambardella L, Ganguly N, Jelasity M, Montemanni R, Montresor A. Design patterns from biology for distributed computing. *ACM Trans. on Autonomous and Adaptive Systems (TAAS)*, 2006,1(1): 26–66.
- [2] Andersen DG. Overlay networks: Networking on top of the network. 2004. <http://www.reviews.com/hottopic/>
- [3] Stoica I, Morris R, Liben-Nowell D, Karger DR, Kaashoek MF, Dabek F, Balakrishnan H. Chord: A scalable peer-to-peer lookup protocol for Internet applications. *IEEE/ACM Trans. on Networking (TON)*, 2003,11(1):17–32.
- [4] Rowstron A, Druschel P. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Guerraoui R, ed. *Proc. of the 18th IFIP/ACM Int'l Conf. on Distributed Systems Platforms (Middleware 2001)*. Berlin, Heidelberg: Springer-Verlag, 2001. 329–350.
- [5] Eugster PT, Guerraoui R, Kermarrec AM, Massoulié L. Epidemic information dissemination in distributed systems. *IEEE Computer*, 2004,37(55):60–67.
- [6] Eugster P, Guerraoui R, Handurukande S, Kermarrec AM, Kouznetsov P. Lightweight probabilistic broadcast. *ACM Trans. on Computer Systems*, 2003,21(4):341–374.
- [7] Kermarrec AM, Massoulié L, Ganesh AJ. Probabilistic reliable dissemination in large-scale systems. *IEEE Trans. on Parallel and Distributed Systems*, 2003,14(3):248–258.
- [8] Ganesh AJ, Kermarrec AM, Massoulié L. Scamp: Peer-to-Peer lightweight membership service for large-scale group communication. In: Crowcroft J, Hofmann M, eds. *Proc. of the 3rd Int'l Workshop on Networked Group Communication*. London: Springer-Verlag, 2001. 44–55.
- [9] Van Renesse R, Minsky Y, Hayden M. A gossip-style failure detection service. In: Davies N, Raymond K, Seitz J, eds. *Proc. of the Middleware'98*. New York: Springer-Verlag, 1998. 55–70.
- [10] Van Renesse R, Birman KP, Vogels W. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Trans. on Computer Systems*, 2003,21(2):164–206.
- [11] Holliday J, Steinke R, Agrawal D, El Abbadi A. Epidemic algorithms for replicated databases. *IEEE Trans. on Knowledge and Data Engineering*, 2003,15(5):1218–1238.
- [12] Jelasity M, Babaoglu O. T-Man: Gossip-Based overlay topology management. In: Brueckner SA, Serugendo GDM, Hales D, Zambonelli F, eds. *Proc. of the Engineering Self-Organising Applications (ESOA 2005)*. Utrecht: Springer-Verlag, 2005.
- [13] Montresor A, Jelasity M, Babaoglu O. Chord on demand. In: Caronni G, Weiler N, Waldvogel M, Shahmehri N, eds. *Proc. of the 5th IEEE Int'l Conf. on Peer-to-Peer Computing (P2P 2005)*. Konstanz: IEEE Computer Society, 2005. 87–94.

- [14] Chawathe Y, Ratnasamy S, Breslau L, Lanham N, Shenker S. Making Gnutella-like P2P systems scalable. In: Feldmann A, Zitterbart M, ed. Proc. of the ACM SIGCOMM 2003. Karlsruhe: ACM Press, 2003. 407–418.
- [15] Zhang H, Goel A, Govindan R. Using the small-world model to improve freenet performance. Computer Networks Journal, 2004, 46(4):555–574.
- [16] Fisk A. Gnutella ultrapeer query protocol v0.1. 2003. [http://groups.yahoo.com/group/the\\_gdf/files/Proposals/Working Proposals/search/Ultrapeer QRP/ultrapeer\\_qrp.html](http://groups.yahoo.com/group/the_gdf/files/Proposals/Working%20Proposals/search/Ultrapeer%20QRP/ultrapeer_qrp.html)
- [17] Daswani S, Fisk A. Gnutella udp extension for scalable searches (guess) v0.1. 2002. [http://groups.yahoo.com/group/the\\_gdf/files/Proposals/Pending Proposals/GUESS/guess\\_01.html](http://groups.yahoo.com/group/the_gdf/files/Proposals/Pending%20Proposals/GUESS/guess_01.html)
- [18] Liu YB, Zhuang ZY, Xiao L, Ni LM. A distributed approach to solving overlay mismatching problem. In: Proc. of the 24th Int'l Conf. on Distributed Computing Systems (ICDCS 2004). Tokyo: IEEE Computer Society, 2004. [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=1281576](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1281576)
- [19] Dabek F, Cox R, Kaashoek F, Morris R. Vivaldi: A decentralized network coordinate system. In: Proc. of the ACM SIGCOMM 2004. ACM Press, 2004. 15–26. <http://portal.acm.org/citation.cfm?id=1030194.1015471>
- [20] Aroiu S, Gummedi PK, Gribble SD. Measuring and analyzing the characteristics of Napster and Gnutella hosts. Multimedia Systems Journal, 2003,9(2):170–184.



孙晓(1981—),男,山东荣成人,讲师,主要研究领域为自适应 overlay networks,网络编码。



姜志宏(1975—),男,博士生,讲师,主要研究领域为 P2P 网络,信任管理。



王晖(1968—),男,博士,教授,博士生导师,CCF 会员,主要研究领域为信息系统工程,多媒体网络。



陶钧(1979—),男,博士生,主要研究领域为 overlay networks,数据存储。



汪浩(1930—),男,教授,博士生导师,主要研究领域为系统工程原理及方法。