

# 分布环境下的 Gossip 算法综述

刘德辉<sup>1,2</sup> 尹 刚<sup>1</sup> 王怀民<sup>1</sup> 邹 鹏<sup>1</sup>

(国防科学技术大学计算机学院 长沙 410073)<sup>1</sup> (中国人民解放军国际关系学院 南京 210039)<sup>2</sup>

**摘 要** Gossip 算法简单、高效,同时具有很好的可扩展性和鲁棒性,很好地适应了无中心、大规模、高度动态的分布式网络环境。近些年在分布计算领域中涌现出了大量 Gossip 相关的研究成果。介绍了 Gossip 算法发展的历史;总结提出了 Gossip 算法的衡量标准和影响因素;对 Gossip 算法在分布环境下的应用进行了综述;最后指出 Gossip 算法研究存在的挑战。

**关键词** Gossip, 分布环境, 交互, 传播, P2P, 聚集计算

中图法分类号 TP393 文献标识码 A

## Overview of Gossip Algorithm in Distribute System

LIU De-hui<sup>1,2</sup> YIN Gang<sup>1</sup> WANG Huai-min<sup>1</sup> ZOU Peng<sup>1</sup>

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)<sup>1</sup>

(PLA University of International Studies, Nanjing 210039, China)<sup>2</sup>

**Abstract** Gossip algorithm is simple, efficacious and scalable, while achieves fault-tolerant information dissemination, it can be applied in decentralized, large scale and dynamic distribute network very well. We introduced the history of Gossip firstly; then the evaluation basis of Gossip Algorithm was proposed, and the factor which will affect the execution of Gossip algorithm was proposed as well; the application of Gossip Algorithm in distribute network was discussed in detail; the challenge in Gossip Algorithm research was discussed lastly.

**Keywords** Gossip, Distribute system, Communication, Spread, P2P, Aggregate computing

## 1 引言

去中心化是网络技术发展的必然趋势。近些年 P2P、无线 AD-HOC 等新型网络技术飞速发展,这些网络具有无中心、可扩展性好、节点高度自治等特点。但是无中心使得网络中的节点只有局部知识,无法进行全局控制;可扩展性使得网络规模日益庞大;节点自治使得网络高度动态,很难保证稳定性和鲁棒性;节点能力上的异构性,导致整个网络或者局部资源(计算、存储、通信)受限。Gossip 算法简单、高效,同时具有很好的可扩展性和鲁棒性,很好地适应了具有上述特点的网络环境。近些年在计算机领域,尤其是分布计算领域中涌现出了大量 Gossip 相关的研究和应用。

本文首先介绍 Gossip 算法的发展历史;介绍 Gossip 相关的重要概念,总结提出了 Gossip 算法的衡量标准和影响 Gossip 算法的因素;综述了已有的 Gossip 相关的应用;最后指出 Gossip 算法研究存在的挑战。

## 2 Gossip 算法的发展历史

1972 年 Hajnal 等人<sup>[1]</sup>首次给出了 Gossip 问题(电话问

题)的描述:有  $n$  个妇女,每个人都知道一条特有的流言,她们通过电话互相联系;任意两个妇女联系上以后,互相交流当前自己知道的所有流言;最少需要多少次联系,使得  $n$  个妇女每个人都知所有流言?这使得对流言问题的研究正式登上历史舞台。

1988 年 Hedetniemi<sup>[2]</sup>等人对计算机网络环境下的 Gossip 问题进行了精确的定义: $A$  是由网络中所有节点组成的集合,每个节点都有自己特有的信息,并需将其传播到网络中其它所有节点;用有序的节点对  $(i, j)$   $i, j \in A$  序列表示信息的传播过程,每个节点对表示两者之间存在信息交换;当序列的最后一个节点对完成交互后,所有节点都知道所有信息,称为 Gossip 过程结束?需要多少次信息交换 Gossip 过程能够结束,需要多长时间?这是 Gossip 研究的核心内容。

1972 年<sup>[3]</sup> Galton-Watson 处理(简单分支处理)模型的出现,使得对 Gossip 的研究有了坚实的理论工具;在简单分支处理模型基础上,1975 年 Bailey<sup>[4]</sup>对 Gossip 进行了更加详尽深入的理论分析;后来马尔科夫链成为 Gossip 算法分析的重要工具。总之 Gossip 算法的研究和分析有着丰富的理论工具:概率、分支处理、马尔科夫链等。

到稿日期:2009-12-10 返修日期:2010-02-26 本文受“973”重点基础研究发展规划项目基金(2005CB321800)和国家自然科学基金项目(60903043)资助。

刘德辉(1979—),男,博士生,主要研究方向为分布计算, E-mail: dhliu1997@gmail.com;尹 刚(1975—),男,博士,主要研究方向为 P2P 信任管理与信息安全;王怀民(1962—),男,教授,博士生导师, CCF 高级会员,主要研究方向为分布计算与信息安全;邹 鹏(1957—),男,教授,博士生导师, CCF 高级会员,主要研究方向为分布计算。

从上个世纪 50 年代到 70 年代,针对 Gossip 的研究层出不穷,人们的研究<sup>[1,3-7]</sup>主要集中在寻找一种合适的模型来很好地描述其运行机理,并对其进行分析;接下来的 10 年里,针对 Gossip 的研究热度有所减低;文献[8]的发表使得 Gossip 研究再次引起人们广泛的关注;新型网络技术的飞速发展、复杂网络相关研究的深入,则进一步推动了 Gossip 研究的发展,近 10 年涌现出了大量关于 Gossip 的研究成果,并在分布计算的众多子领域得到广泛应用,如数据库复制、故障探测、P2P 的拓扑构造及维护、分布环境下的聚合计算等。

### 3 Gossip 算法

#### 3.1 概念

##### 3.1.1 网络结构模型

我们用图对真实网络的结构进行抽象。这里的网络可以是计算机网络、传感器网络,也可以是由人通过不同联系方式组成的社会网络。一个网络可以表示为  $G=(V, E)$ , 其中  $V$  表示节点(网络成员)的结合;  $E$  为边,即节点之间的连接关系的集合。  $N=|V|$ , 指网络中节点的数目;  $A=(a_{ij})$  为  $G$  的邻接矩阵,假设是  $(0, 1)$  矩阵。如果从节点  $n_i$  到  $n_j$  存在边,且  $i \neq j$ , 有  $a_{ij}=1$ , 否则为 0。如果  $G$  为无向图,则有  $a_{ij}=a_{ji}=1$ 。如果  $a_{ij}=1$ , 我们称节点  $n_j$  为  $n_i$  的邻居节点。

##### 3.1.2 时间模型

时间模型是根据系统对时间的约束假设建立的模型,它是 Gossip 算法的基础。不同的时间模型对 Gossip 算法的设计和分析有着很大的影响。约束苛刻的时间模型,甚至使得 Gossip 算法在分布环境下无法实施。目前主要有两类时间模型:同步时间模型、异步时间模型。

##### 同步时间模型

该模型将时间分为时隙,每个节点严格地在每个时隙与自己的邻居节点进行信息交换,邻居节点的选择过程是随机且独立的。该模型要求所有节点的信息交换过程在同一时刻发生,整个网络需要一个全局时钟来同步网络中所有节点的时间。早期人们的研究<sup>[10,11]</sup>都以同步模型为基础。

##### 异步时间模型

文献[12]精确地定义了异步时间模型:每个节点都有自己的时钟,并根据自己的时钟嘀嗒每隔  $\Delta T$  发起信息交换,其信息交换时刻是速率为 1 的泊松过程。所有节点信息交换的时刻独立同分布。将网络中  $N$  个节点信息交换的时刻合成,得到一个速率为  $N$  的泊松过程。假设 Gossip 过程的开始时刻为  $Z_0$ , 且  $Z_{k+1}-Z_k=\Delta T, k \geq 1$ , 那么在  $[Z_k, Z_{k+1})$  时间段内平均有  $n$  个信息交换的事件发生,称  $[Z_k, Z_{k+1})$  为该 Gossip 过程的第  $k$  个周期。异步时间模型放松了对全局时钟同步的要求,使得 Gossip 算法在分布式环境中的实现更加简单。

#### 3.2 衡量标准

研究者设计不同的 Gossip 算法,主要是为了提高效率以及减小完成传播过程需要的通信开销。所以一般衡量 Gossip 算法的好坏有两个标准:效率、通信代价。

##### 3.2.1 效率

效率是指完成 Gossip 传播所需要的时间。早期<sup>[13]</sup>假设 Gossip 传播过程中的信息交换需要一个时间单位,那么 Gossip 算法的效率是指完成传播需要的最少时间单位。文献[14]根据同步时间模型提出了周期的概念,并用完成传播需

要的周期数表示 Gossip 算法执行的效率。由于 Gossip 传播的完成只能在概率上保证,令  $x(t)$  表示在  $t$  周期已感染(为了方便描述,本文用感染表示 Gossip 传播过程中节点接受外来信息的状态)节点的比例,文献[15]给出了  $\epsilon$ -完成时间,即使得  $x(t) > \epsilon$  的最小的周期数  $t$ 。其它还有一些针对不同应用对效率的定义,如在均值计算<sup>[12]</sup>中定义了  $\epsilon$ -平均时间,在有  $n$  个节点的网络中假设每个节点都有一个初始  $x$  值,这些值组成的向量为  $X(0)=(x_0(0), x_1(0), \dots, x_n(0))$ , 其均值为  $x_{\text{true}} = \frac{1}{n} \sum_{i=0}^{n-1} x_i(0)$ , 假设经过  $t$  个周期后所有  $x$  值组成的向量为  $X(t)=(x_0(t), x_1(t), \dots, x_n(t))$ , 定义向量  $X_{\text{true}} = (x_{\text{true}}, x_{\text{true}}, \dots, x_{\text{true}})$ ,  $\epsilon$ -平均时间是使得  $\Pr(\|X(t) - X_{\text{true}}\| \geq \epsilon \|X(0)\|) \leq \epsilon$  的最小的  $t$  值。

##### 3.2.2 通信代价

通信代价指完成 Gossip 传播过程所需要付出的通信开销,一般用完成传播所需要的节点之间的两两交互次数表示。最初,研究者<sup>[2]</sup>一般分析完成传播需要的最少交互次数。但由于通信代价与效率是互相影响的一对因素,因此精确计算理想情况下的交互次数是不合理的,后来的研究一般将两者结合起来讨论。

#### 3.3 影响因素

一般来讲,影响 Gossip 算法执行的因素有很多,我们对其中的主要因素进行分析:网络结构、交互模式以及每个周期、每个节点选择联系的节点个数。

##### 3.3.1 网络结构

网络结构是指网络中节点之间的连接关系,一般用图表示。因为 Gossip 算法在执行过程中的每个周期会选择一个或多个节点,并向其发起信息交换请求,被选择的节点必然与选择节点之间存在连接关系,所以节点之间的连接关系直接影响到 Gossip 算法的执行效果。

##### 规则网络

早期人们以规则网络为背景研究 Gossip 算法,主要包括:完全图、树形。

最初人们主要以电话模型<sup>[1]</sup>为基础进行 Gossip 算法研究,在该模型中任意节点之间都可以建立联系,也就是说底层网络可以用完全图来表示。在该模型下,研究者<sup>[16-19]</sup>给出了在  $n$  个节点的网络(完全图)中完成 Gossip 传播需要的最少通信次数,用函数  $f$  表示:

$f(1)=0; f(2)=1; f(3)=3; f(n)=2n-4, n \geq 4$ 。同时  $f(n)$  还满足以下关系:  $f(n+1) \leq f(n)+2, n \geq 4$ 。

当底层网络为树形结构时<sup>[20]</sup>,完成 Gossip 过程需要的最少通信次数可以用函数  $f(n)$  表示:  $f(n)=2n-3, n \geq 2$ 。

##### 任意结构

上述研究都是以规则网络为研究背景,后来研究者尝试找出不同网络影响信息传播的关键因素,以及适用于任意结构网络的一般结论。当底层通信网络为有  $n$  个节点的任意结构的连通图时<sup>[21]</sup>,完成传播需要的最少通信次数用函数  $f(n)$  表示:  $2n-4 \leq f(n) \leq 2n-3, n \geq 4$ ; 如果图中存在 4 个环时,完成传播需要的最少通信次数为  $f(n)=2n-4, n \geq 4$ 。

近几年出现了一些更有意义的研究成果。研究者<sup>[12]</sup>以分布环境中的均值计算为研究背景,根据节点之间的连接关系定义了一个双随机矩阵,以马尔科夫链为工具分析  $\epsilon$ -平均时间。文献[12]认为任意图形的  $\epsilon$ -平均时间依赖于上述双随机

机矩阵的第二大特征值。作者用文中方法分别对无线传感器网络环境下的几何随机图(Geometric Random Graphs)和互联网环境下的优先连接模型<sup>[22]</sup>(Preferential Connectivity (PC) model)进行分析,得到以下结论:在一个  $d$  维  $n$  个节点的几何随机图  $G_d(n, r)$  中,无线信号的传输半径为  $r$ ,  $1/na$  平均时间为  $\theta(\log n/r2)$ ,  $a>0$ ;在 PC 模型下  $\epsilon$  平均时间为  $\theta(\log e-1)$ 。

文献[23]以社会网络为背景对此问题进行研究,问题定义如下:在网络中随机选择一个节点作为流言的受害者,从该节点的邻居中任选一个作为流言的制造者,在第一个周期制造者只将流言传播给自己或者受害者的邻居中的一个节点,以后的每个周期流言只是传播到受害者的邻居中不知道该流言的节点。当受害者的邻居节点都知道该流言时,传播过程结束。作者研究发现在无标度(scale-free)网络中传播时间  $T$  与受害节点的度数  $k$  存在如下关系:  $T=A+B * \log k$ 。当流言可以传播到整个网络时,在节点数目  $n$  固定情况下传播周期为常数,与受害者节点的度数  $k$  无关。

### 3.3.2 交互模式

交互模式主要回答以下两个问题:节点在什么条件下发起信息交换? 交换什么样的信息内容? 参与交互的两个节点采用何种信息交换方式?

关于第一个问题,文献[8]给出了两类方法:Anti-entropy, Rumor mongering。Anti-entropy:每个节点周期性地随机选择其他节点,然后通过互相交换自己的所有数据来消除两者之间的差异。Anti-entropy 方法非常可靠,但是每次节点两两交换自己的所有数据会带来非常大的通信负担,因此不能频繁使用。Rumor mongering 的主要思想是,当一个节点有了新信息后,该节点变成活跃状态,并周期性地联系其它节点向其发送新信息,直到所有节点都知道该新信息。因为节点之间只是交换新信息,所以大大减低了通信负担。一般来讲,为了在通信代价和可靠性两者之间取得折中,需要将两种方法结合使用,一般情况采用 Rumor mongering,每隔一段时间使用一次 Anti-entropy,以保证信息交换的可靠性。

关于第二个问题,目前节点间的信息交换方法主要有 3 种:Push, Pull, Push&Pull。

Push:发起信息交换的节点随机选择联系节点并向其发送自己的信息,一般拥有新信息的节点才会作为发起节点。采用 Push 方式,在信息传播的初期,已感染节点的数目呈指数增长;当已有一半节点被感染时,每个周期感染节点的数目会迅速减少。其流程如图 1 所示。

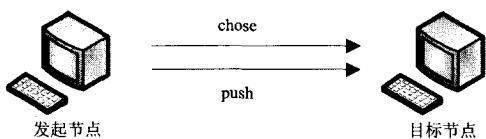


图 1 Push 模式

Pull:发起信息交换的节点随机选择联系节点并从对方获取信息。一般无新信息的节点才会作为发起节点。Pull 方式与 Push 方式相反,在信息传播的初期,感染节点的数目增长缓慢;当已有一半节点被感染时,每个周期未感染节点的数目呈乘性减少。其流程如图 2 所示。

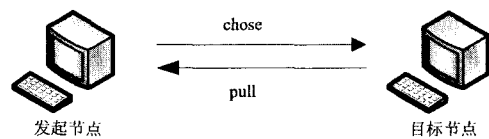


图 2 Pull 模式

在全连通网络中,Push 和 Pull 两种方式完成 Gossip 过程都需要  $O(\ln n)$  个周期和  $O(n \ln n)$  次信息交换。而 Push&Pull 是将两者结合起来,发起信息交换的节点向选择的节点发送信息,同时从对方获取数据,流程图如图 3 所示。

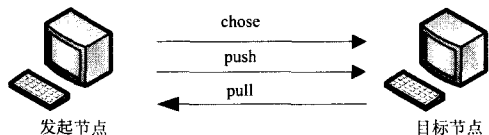


图 3 Push&Pull 模式

文献[10]将 Push 和 Pull 进行结合,感染节点数目在  $\lfloor n/2 \rfloor$  之间时 Push 起主要作用,感染节点数目在  $\lfloor 0, n/2 \rfloor$  之间时 Pull 起主要作用。在全连通网络中采用 Push&Pull 方式需要  $O(\ln n)$  个周期和  $O(n \ln \ln n)$  次信息交换就能将信息传播到网络中所有节点。文献[10]认为 Push&Pull 的执行需要一个精确的全局时钟,用于同步所有节点的时间,同时容错性较差,因此对其进行改进,提出了 median-counter 算法,其基本思路如图 4 所示。在全连通网络中,采用 median-counter 算法仅需要  $\log_3 n + O(\log \log n)$  个周期、 $O(n \log \log n)$  条消息就能完成传播过程。

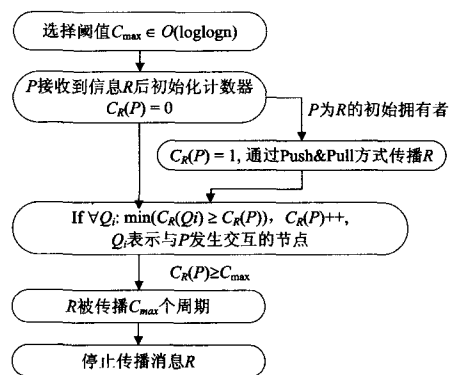


图 4 median-counter 算法主要思想

### 3.3.3 联系节点数目

每个周期联系节点个数不同会对 Gossip 算法的执行产生影响。针对该问题的研究<sup>[24-26]</sup>首先出现在超图中:在一个  $k$ -uniform 的超图中有  $n$  个节点,每条边恰好连接  $k$  个节点,每个周期、每条边连接的所有节点都参与信息交换,完成传播过程最少需要的消息  $f(n, k)$  为

$$f(n, k) = \begin{cases} \left\lceil \frac{n-k}{k-1} \right\rceil + \left\lceil \frac{n}{k} \right\rceil, & 1 \leq n \leq k^2 \\ 2 \left\lceil \frac{n-k}{k-1} \right\rceil, & n \geq k^2 \end{cases}$$

后来人们开始在普通网络下研究该问题<sup>[4]</sup>,在  $n$  个节点的全连通网络中,最初有一个节点被感染,已感染节点每个周期选择  $k$  个其他节点进行传播,经过  $r$  个周期以后被感染节点的比例为  $1/(1+ne^{-kr})$ ,网络中所有节点被感染需要的周期数为  $O(\log_3 n)$ 。

## 4 Gossip 应用

最初人们只是将 Gossip 算法作为一个通用的问题加以研究。文献[8]的发表使得人们在研究 Gossip 算法本身的同时,更加关注如何将其应用到不同的应用环境中,或者说研究适用于某一具体环境的 Gossip 算法。我们主要对数据库复制、聚集计算、基于 Gossip 的 Overlay 构造等几类应用进行综述。

### 4.1 数据库复制

传统的应用于数据副本管理的同步技术只能应用在相对稳定的网络环境以及较小的网络规模下,并且只能管理有限数量的数据副本,无法适应动态、大规模的广域网环境。而 Gossip 算法具有简单、高效、可扩展性好以及较好的鲁棒性等特点,使用 Gossip 机制进行分布数据库的副本管理是一个很好的选择。

文献[8]是第一个将 Gossip 算法引入分布式数据库领域的研究。作者就分布式数据库中副本管理问题提出了 3 种解决方法:direct mail, anti-entropy, rumor mongering, 并对这 3 种方法进行分析 and 比较。

基于 Gossip 实现分布数据管理的一般思路是:在一个节点实现数据更新,通过 Gossip 算法将更新传播到其他节点。作者<sup>[27]</sup>认为该方法能够保证数据一致性的原因是 Gossip 算法能够保证分布式事务的因果序,而这需要分布式事务之间没有冲突作为前提。针对该问题,在广域网环境下,在数据副本较少且所有副本皆可以执行更新操作的前提下,作者以 Gossip 算法为基础,在分布式事务处理框架下,根据不同情况提出了 3 种数据副本管理算法,以保证副本数据的一致性、事务的原子性。

文献[28]从数据存储的持久性和一致性两个角度,在广域网、无线传感器网络、移动 Ad hoc 网络等环境下,对 Gossip 机制的适用性进行了分析,认为在分布式数据存储系统中采用 Gossip 算法进行数据副本管理是可行的。

### 4.2 聚集计算

在大型分布计算环境中,许多应用需要获得全局统计信息,以提高其服务质量或者均衡网络负载。在许多无中心的系统(如 P2P 网络、大型传感器网络等)中,没有中心节点用于存储这些信息,同时这些环境往往具有大规模、高度动态、资源受限等特征,如何设计一种高效率、鲁棒的聚集计算(MIN, MAX, SUM, AVERAGE 等)方法以获取全局统计信息,成为研究者关注的热点。基于 Gossip 的聚集计算完美地契合了这种需要。

Gupta 等人<sup>[29]</sup>首次将 Gossip 算法应用到分布环境下的聚集计算中,文献[29, 30]将 Gossip 算法与层次型结构结合起来完成聚集计算,其最大的缺陷是需要将节点组织成严格的层次结构。Kempe 等人<sup>[11]</sup>对完全图中的聚集计算进行研究,基于 Push 模式的 Gossip 算法给出了(SUM, Averages)等聚集计算的实现方法,同时定义了扩散速度描述信息传播的速度,并证明了在全连通网络中存在 Gossip 算法,其  $1/n$  平均时间为  $O(\log n)$ 。

Jelasity 等<sup>[31]</sup>采用 Push&Pull 模式的 Gossip 算法实现了全连通网络中的聚集计算,重点给出了同步模式下联系节点的选择方法。令节点被选择参与交互的次数是随时间变化的

变量  $\Phi$ , 那么采用文中的方法可以保证网络中所有节点的  $\Phi$  值服从同一分布。文献[32]扩展了研究范围,考察了在随机网络、无标度网络以及小世界网络环境下 Gossip 算法的适应性和效率,分析了网络中节点的动态性对聚集计算的影响,并给出了相应解决方法。

Kashyap 等人<sup>[33]</sup>研究了全连通网络中的聚集计算问题,以通信受限网络为背景,给出巧妙的聚集计算方法,该方法只稍微增加了传播时间( $O(\log n * \log \log n)$ ),而大大减小了通信开销( $n \log \log n$ )。Maya Haridasan 等人<sup>[34]</sup>重点研究多变量的聚集计算问题,并对多种数据摘要技术进行比较分析。

### 4.3 基于 Gossip 的 Overlay 构造

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,其拓扑结构及其维护是其核心。基于 Gossip 构造 Overlay 实际上是以 Gossip 机制为基础,在信息交换过程中对 Overlay 中的节点以及节点之间的关系进行管理,从而维护一个具有某种特性的网络拓扑。

A. J. Ganesh 等人<sup>[35]</sup>首次提出了一种利用 Gossip 机制构造 P2P 网络并进行成员管理的模型。BISON 项目组<sup>[36]</sup>基于 Gossip 制定了一种 Overlay 拓扑管理协议 T-man, 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。同时讨论了构造树型、环型网络的基本方法。文献[37]构造了能够及时反映网络状况的 Chord 环结构。

文献[38]提出了一种基于自适应周期的流言机制,以支持快速构建自组织 Overlay 拓扑。该文献作者认为已有研究在拓扑构造过程中的固定周期的流言报文存在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。

### 4.4 其它应用

Gossip 机制还在分布环境下的其它许多研究领域得到广泛应用,如故障检测<sup>[39]</sup>、网络监控<sup>[40]</sup>、无线网络中的路由技术<sup>[41, 42]</sup>。

在众多的研究领域中, Gossip 算法能够得到广泛的应用是由于其高可扩展性、鲁棒性,以及简单的实现,而各种应用之间的区别在于利用 Gossip 消息的内容不同、信息交换的目的不同。总之,在大规模、动态、异构、通信受限的网络环境中,基于 Gossip 算法解决相应的问题是一个很好的选择。

**结束语** Gossip 算法简单、高效,同时具有很好的可扩展性和鲁棒性,很好地适应了大规模、动态、资源受限的网络环境。近些年在计算机领域,尤其是分布计算领域中涌现出了大量 Gossip 相关的研究和应用。尽管如此,作者认为还有以下挑战有待人们作进一步的研究:

(1) 在不同结构的网络中,影响 Gossip 算法执行的关键因素究竟是什么? 结构化网络的网络结构具有什么样的性质? 这些性质对 Gossip 算法执行有什么影响?

(2) 异步时间模型似乎使得在分布环境下 Gossip 算法的实现成为可能,但是目前绝大部分研究都假设节点在某个周期内只将自己上一周期结束时拥有的信息传播给其他节点,这种假设似乎与现实不相符。

(3) 节点频繁加入、退出, 必然导致网络中的数据不稳定, 如何定义周期长度才能够及时反映网络中节点(或者数据)的变化同时不会带来沉重的通信负担?

## 参考文献

- [1] Hajnal A, Milner E C, Szemerédi E. A cure for the telephone disease[J]. *Canad. Math Bull.*, 1972(15):447-450
- [2] Hedetniemi S, Liestman A. A survey of gossiping and broadcasting in communication networks[J]. *IEEE Networks*, 1988, 18(4):319-349
- [3] Athreya K B, Ney P. *Branching Processes* [M]. New York: Springer Verlag, 1972
- [4] Bailey N T J. *The Mathematical Theory of Infectious Diseases and its Applications*(second edition)[M]. Hafner Press, 1975
- [5] Bavelas A. Communication patterns in task-oriented groups[J]. *J. Acoust. Soc. Amer.*, 1950(22):725-730
- [6] Leavitt H J. Some effects of certain communication patterns on group performance[D]. Massachusetts Institute of Technology, 1949
- [7] Shimbel A. Applications of matrix algebra to communication nets [J]. *Bull. Math Biophys.*, 1951(13):165-178
- [8] Demers A, Greene D, Hauser C, et al. Epidemic algorithms for replicated database maintenance[C]// *Proc. of the 6th ACM Symp. on Principles of Distributed Computing (PODC '87)*. 1987:1-12
- [9] Gupta I, Kermarrec A-M, Ganesh A J. Efficient and Adaptive Epidemic-style Protocols for Reliable and Scalable Multicast[J]. *IEEE Transaction on Parallel and Distributed Systems*, 2006, 17(7):593-605
- [10] Karp R, Schindelhauer C, Shenker S, et al. Randomized rumor spreading[C]// *Proc. Symp. Foundations of Computer Science*. 2000:564-574
- [11] Kempe D, Dobra A, Gehrke J. Gossip-based computation of aggregate information[C]// *Proc. Conf. Foundations of Computer Science*. 2003:482-491
- [12] Boyd S, Ghosh A, Prabhakar B, et al. Randomized Gossip Algorithms[J]. *IEEE Transactions on Information Theory* (Special issue of IEEE Transactions on Information Theory and IEEE ACM Transactions on Networking), 2006, 52(6):2508-2530
- [13] Bavelas A. Communication patterns in task-oriented groups[J]. *J. Acoust. Soc. Amer.*, 1950(22):725-730
- [14] Frieze A M, Grimmett G R. The shortest-path problem for graphs with random arc-lengths[J]. *Discrete Appl. Math.*, 1985, 10:57-77
- [15] Pittel B. On spreading a rumor [J]. *SIAM J. Applied Math.*, 1987, 47:213-223
- [16] Bumby R. A problem with telephones[J]. *SIAM J. Afg. Disc. Meth.*, 1981(2):18-13
- [17] Hajnal A, Milner E C, Szemerédi E. A cure for the telephone disease[J]. *Canad. Math Bull.*, 1972(15):447-450
- [18] Seress A. Quick gossiping without duplicate transmissions[J]. *Graphs and Combinatorics*, 1986(2):363-383
- [19] Tijdeman R. On a telephone problem [J]. *Nieuw Arch. Wisk.*, 1971, 19:188-192
- [20] Harary F, Schwenk A J. The communication problem on graphs and digraphs[J]. *Franklin Inst.*, 1974, 29:491-495
- [21] Golumbic M C. *The general gossip problem*[R]. IBM Research Report RC 4977. IBM, August 1974
- [22] Mihail M, Papadimitriou C, Saberi A. On certain connectivity properties of the internet topology[C]// *Proc. Conf. on Foundations of Computer Science*. 2003:28-35
- [23] Lind P, Silva L, Andrade J, et al. Spreading gossip in social networks[J]. *Phys. Rev. E.*, 2007, 76:036117
- [24] Bermond J C. Le problème des 'ouvroires' (hypergraph gossip problem) [C]// *Colloq. CNRS. Problemes Combinatorics et Theorie des Graphes*. 1976:31-34
- [25] Lebensold K. Efficient communication by phone calls[J]. *Stud. Appl. Math.*, 1973, 52:345-358
- [26] Kleitman D J, Shearer J B. Further gossip problems[J]. *Discr. Math.*, 1980, 30:151-156
- [27] Member J H, Holliday J, Steinke R, et al. Epidemic algorithms for replicated databases[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2003:1218-1238
- [28] Weatherspoon H, Miranda H, Iwanicki K, et al. GOSSIP: Gossiping Over Storage Systems is Practical[J]. *ACM SIGOPS Operating Systems Review*, 2007, 41(5):75-81
- [29] Gupta I, van Renesse R, Birman K. Scalable fault-tolerant aggregation in large process groups[C]// *Proc. Conf. on Dependable Systems and Networks*. 2001:433-442
- [30] van Renesse R. Scalable and secure resource location[C]// *HICSS*. 2000
- [31] Jelasity M, Montresor A. Epidemic-Style Proactive Aggregation in Large Overlay Networks[C]// *Proceedings of the 24th International Conference on Distributed Computing Systems*. 2004:102-109
- [32] Jelasity M, Montresor A, Babaoglu O. Gossip-based aggregation in large dynamic networks[J]. *ACM Trans. Comput. Syst.*, 2005, 23(3):219-252
- [33] Kashyap S, Deb S, Naidu K V M, et al. Efficient gossip-based aggregate computation[C]// *Proceedings of the Twenty-fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. 2006:308-317
- [34] Haridasan M, van Renesse R. Gossip-based Distribution Estimation in p2p Network[C]// *iptps*. 2008
- [35] Ganesh A J, Kermarrec A M, Massoulié L. *Scamp: Peer-to-Peer lightweight membership service for large-scale group communication*[C]// *Crowcroft J, Hofmann M, eds. Proc. of the 3rd Int'l Workshop on Networked Group Communication*. London: Springer-Verlag, 2001:44-55
- [36] Jelasity M, Babaoglu O. T-Man: Gossip-based overlay topology management[C]// *Brueckner S A, Serugendo G D M, Hales D, eds. Proc. of the Engineering Self-organising Applications (ESOA 2005)*. Utrecht: Springer-Verlag, 2005
- [37] Montresor A, Jelasity M, Babaoglu O. Chord on demand[C]// *Caronni G, Weiler N, Waldvogel M, eds. Proc. of the 5th IEEE Int'l Conf. on Peer-to-Peer Computing (P2P 2005)*. Konstanz: IEEE Computer Society, 2005:87-94
- [38] 孙晓, 王晖, 汪浩, 等. 基于自适应周期的流言机制快速构建自组 Overlay 拓扑[J]. *软件学报*, 2008, 19(9):2422-2431
- [39] Van Renesse R, Minsky Y, Hayden M. A gossip-style failure detection service[C]// *Davies N, Raymond K, Seitz J, eds. Proc. of the Middleware'98*. New York: Springer-Verlag, 1998:55-70
- [40] van Renesse R, Birman K P, Vogels W. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining[J]. *ACM Trans. on Computer Systems*, 2003, 21(2):164-206
- [41] Beraldi R. The Polarized Gossip Protocol for Path Discovery in MANETs[J]. *Ad Hoc Networks*, 2008, 6(1):79-91
- [43] Haas Z J, Halpern J Y, Li Li. Gossip-based Ad Hoc Routing[J]. *INFOCOM*, 2002, 3:1707-1716