

基于蚁群算法的容迟网络概率路由算法

宋 鑫, 王炳庭, 胡 勇, 王贵竹

(安徽大学计算智能与信号处理教育部重点实验室, 合肥 230039)

摘 要: 容迟网络的概率路由算法根据节点与信宿节点之间相遇的概率判断是否转发报文, 但相遇概率不能真正准确地反映报文递交成功的概率。针对该问题, 提出一种基于蚁群算法的概率路由算法, 结合蚁群算法的基本原理改进信息素浓度的更新机制和报文转发机制。仿真结果表明, 与传统的概率路由算法相比, 改进后的概率路由算法报文递交率较高, 开销率较小。

关键词: 容迟网络; 概率路由; 蚁群算法; 贝叶斯因子; 选择概率; 信息素浓度更新

Probabilistic Routing Algorithm Based on Ant Colony Algorithm in DTN

SONG Xin, WANG Bing-ting, HU Yong, WANG Gui-zhu

(Key Lab of IC&SP of Ministry of Education, Anhui University, Hefei 230039, China)

【Abstract】 The probabilistic routing algorithm in Delay Tolerant Network(DTN) called PROPHET, determines whether to forward messages according to the encounter probability between node and destination. However encounter probability can not accurately show the successfully delivered probability. Aiming at this problem, PROPHET-ACA is proposed, and it combines the fundamental of ant colony algorithm with probabilistic routing. The rule both in pheromone update and in message transmission is modified. Simulation results show that compared with the PROPHET, in PROPHET-ACA, the delivery probability of message is obviously increased higher and the overhead ratio is decreased.

【Key words】 Delay Tolerant Network(DTN); probabilistic routing; ant colony algorithm; Bayes factor; selection probability; pheromone concentration update

DOI: 10.3969/j.issn.1000-3428.2013.04.022

1 概述

容迟网络^[1-2](Delay Tolerant Network, DTN)是一种新的网络模型, 容迟网络没有稳定的端到端通信连接, 网络拓扑结构动态变化。2003 年在 ICIR 会议上 Kevin Fall 等科学家提出了这种通用的、面向消息的、可靠的覆盖层网络体系结构。容迟网络与传统网络结构相比, 增加了一个新的协议层, 称为聚束层(Bundle Layer)^[3], 可以通过存储-携带-转发的方式执行报文传输。

网络体系结构的不同导致路由算法的差异, DTN 路由算法成为目前研究的热点。概率路由算法(PROPHET)^[4]是 DTN 中一种常见的路由算法, 这种算法根据历史信息转发报文。蚁群算法(ant colony algorithm)^[5]也是一种基于历史信息的算法, 算法中“蚂蚁”通过“信息素”和环境进行通

信。蚁群算法已应用于很多网络, 并取得了理想效果, 文献[6]利用正向超时蚂蚁和反向广播蚂蚁共同协作, 提高了路由的效率。蚁群算法在 DTN 中也有所应用, 文献[7]提出了一种基于蚁群算法的组播路由算法。

本文考虑将 2 种基于历史信息的算法结合, 找到一种新的判断依据和更新机制, 提出一种基于蚁群算法的概率路由算法, 以提高历史信息的准确度, 改善网络性能。

2 相关工作

2.1 概率路由

DTN 的概率路由协议根据相遇和传输的历史信息来判断是否转发报文。2 个节点相遇时, 若对端节点与信宿节点相遇的概率大于自身与信宿节点相遇的概率, 相应的报文转发给相遇的对端节点, 否则不转发报文。定义 $P_{(a,b)} \in (0,1]$

基金项目: 国家自然科学基金资助项目(61071168); 安徽大学“211 工程”学术创新团队基金资助项目; 安徽大学“211 工程”三期质量工程基金资助项目(40010014, 39010014); 安徽大学通信工程重点建设专业基金资助项目(39020014)

作者简介: 宋 鑫(1987—), 男, 硕士研究生, 主研方向: 容迟网络路由算法; 王炳庭, 硕士研究生; 胡 勇, 讲师、硕士; 王贵竹, 副教授、博士

收稿日期: 2012-04-26 **修回日期:** 2012-06-28 **E-mail:** 115487208@qq.com

为任意节点 a 与节点 b 的相遇概率, 将相遇概率的更新和计算分为 3 个部分。

首先当 2 个节点相遇的时候需要对相遇概率进行更新, 使频繁相遇的节点之间相遇概率更大, 计算更新如式(1)所示, 其中 $P_{\text{init}} \in (0,1]$ 是一个初始化的常数。

$$P(a,b) = P(a,b)_{\text{old}} + (1 - P(a,b)_{\text{old}}) \times P_{\text{init}} \quad (1)$$

若节点 a 和节点 b 有一段时间没有相遇, 相遇概率需要衰减。 $P(a,b)$ 的衰减如式(2)所示, $\gamma \in (0,1]$ 是衰减系数, 可以反映出概率衰减的快慢程度, k 表示从上一次相遇到现在的时间间隔。

$$P(a,b) = P(a,b)_{\text{old}} \times \gamma^k \quad (2)$$

相遇概率还具有传递性, 当节点 a 和另一个节点 c 频繁相遇, 节点 c 又与节点 b 频繁相遇, 节点 c 将会对 $P(a,b)$ 产生影响。传递性更新公式如式(3)所示, 其中, $\beta \in (0,1]$ 是一个传递因子, 表示传递性对相遇概率影响的大小, 这样可以通过另一个活跃的中间节点更新两节点之间的相遇概率。

$$P(a,b) = P(a,b)_{\text{old}} + (1 - P(a,b)_{\text{old}}) \times \frac{P(a,c) \times P(c,b) \times \beta}{P(a,b)} \quad (3)$$

2.2 蚁群算法

蚁群算法模拟蚂蚁觅食的过程, 蚂蚁总是选择信息素浓度大的路径, 所以信息素浓度越大, 选择这条路径的可能性越大, 反之, 信息素的浓度越小, 选择这条路径的概率就越小, 多次迭代后, 最终发现从源节点寻找信宿节点的最佳路径。

(1) 选择机制

假设有 m 只蚂蚁, $P_j^k(t)$ 表示第 k 只蚂蚁在 t 时刻选择节点 j 作为下一跳节点的概率, 计算公式如下:

$$P_j^k(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha [\eta_j(t)]^\theta}{\sum_{s \in \text{allowed}(i)} [\tau_{i,s}(t)]^\alpha [\eta_{i,s}(t)]^\theta} & \text{if } j \in \text{allowed}(i) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

其中, $\tau_{i,j}(t)$ 表示的节点 i 与节点 j 之间的信息素浓度; $\text{allowed}(i)$ 表示一个可以选择的节点集合; $\eta_j(t)$ 是一个影响选择概率的启发函数; α 是信息素浓度的权重系数, 表示路径上的信息量对蚂蚁选择路径的影响大小; θ 是启发函数的权重系数, 反映了启发信息在蚂蚁选择路径时受重视的程度。

(2) 信息素的更新机制

信息素的更新主要有 2 种: 局部更新和全局更新。蚂蚁的经过会增加路径上信息素的浓度, 而没有蚂蚁经过, 信息素的浓度会随着时间逐渐衰减。

1) 信息素浓度的局部更新

根据浓度变化的要求, 用下列公式对局部信息素浓度进行更新:

$$\tau_{i,j}(t+1) = (1 - \rho) \times \tau_{i,j}(t) + \rho \times \Delta \tau_{i,j}^k(t) \quad (5)$$

$$\Delta \tau_{i,j}^k(t) = 1 / \text{cost}(i, j) \quad (6)$$

其中, $\rho \in [0,1]$ 是信息素浓度的一个局部衰减系数; $\Delta \tau_{i,j}^k(t)$ 是第 k 只蚂蚁路过某条路径时信息素浓度的增量。

2) 信息素浓度的全局更新

得到全局的最优解后, 信息素浓度要进行全局的更新:

$$\tau_{i,j}(t+1) = (1 - \lambda) \times \tau_{i,j}(t) + \lambda \times \Delta \tau_{i,j}(t) \quad (7)$$

$$\Delta \tau_{i,j}(t) = \sum_{k=1}^m \Delta \tau_{i,j}^k(t) \quad (8)$$

$$\Delta \tau_{i,j}^k(t) = 1 / \text{cost}(i, j) \quad (9)$$

其中, $\lambda \in [0,1]$ 是信息素浓度的全局衰减系数; $\Delta \tau_{i,j}(t)$ 代表 m 只蚂蚁经过后网络环境中信息素浓度增加的总量。

3 基于蚁群算法的概率路由

将蚁群算法应用到 DTN 中, 首先需要依据蚂蚁选择机制选择下一跳节点, 其次需要建立合适的信息素更新规则, 构建一个满足蚁群算法要求的网络模型。

3.1 概率路由算法中的信息素浓度

在 DTN 的概率路由算法中, 当一个节点携带报文与另一个节点相遇时, 将当前节点与信宿节点相遇的概率和对端节点与信宿节点之间相遇的概率进行比较, 从而做出副本传送或者继续保留的选择。不妨将一个节点看作一个报文的路径选择, 可以看出与信宿节点之间的相遇概率越大, 报文选择这条路径的可能性越大。根据这一推断, 使用一个节点与信宿节点之间的相遇概率来表征一条路径上的信息素浓度。

3.2 贝叶斯因子

在概率论中贝叶斯公式^[8]提供了一种从结果发生概率反推原因发生概率的基本方法, 基本公式如下:

$$P(T_k | A) = P(T_k)P(A | T_k) / \sum_{i=1}^n P(T_i)P(A | T_i) \quad (10)$$

式(10)中事件 A 是产生的结果, 事件 T_k 是产生事件 A 的一组互不相容的原因事件。 $P(T_k|A)$ 表示事件 A 是由事件 T_k 引起的概率, $P(A|T_k)$ 表示在事件 T_k 的条件下发生事件 A 的概率。

将 A 看作成功递交到信宿节点, 事件 T_k 看作是报文经过不同的节点成功递交到信宿节点, T_k 之间互不相容, 满足贝叶斯公式的要求。 当一个报文被成功递交到信宿节点时, 信宿节点根据报文信息得到报文所经过的节点, 由式(10)得到一组关于报文所经过节点的概率值, 定义为贝叶斯因子 B , 使用下面的公式来表征贝叶斯因子:

$$B = P(u, v_i) / \sum_{i=1}^n P(u, v_i) \quad (11)$$

其中, B 为贝叶斯因子; v_i 表示途经的节点; n 表示除信宿节点与信源节点之外途经的节点的数目; $P(u, v_i)$ 为信宿节点与途经节点之间的相遇概率。网络中的节点维护一张动态更新的表来存储一个节点与其他节点之间的贝叶斯因子。

3.3 启发因子

在容迟网络中, 节点发生拥塞时, 要丢弃报文, 为新接收到的报文腾出缓存空间。在这种改进的算法中, 通过估计网络中所有节点的拥塞情况作为启发因子。定义初始

时刻为 t_0 , 每次丢弃报文的时候节点记录一个时刻 t_i , $i=1,2,\dots,n$, 每相邻的 2 个记录时刻 t_i 和 t_{i-1} 做差得到节点没有发生拥塞的时间段 Δt_i , 如式(12)所示:

$$\Delta t_i = t_i - t_{i-1} \quad (12)$$

将所有没有发生拥塞的时间段求和得到当前时刻为止没有发生拥塞的时间段总和, 如式(13)所示:

$$\Delta t_{\text{total}} = \Delta t_1 + \Delta t_2 + \dots + \Delta t_n \quad (13)$$

通过 Δt_{total} 在总时间中所占的比重可以估计节点 s 拥塞程度, 其计算公式如下:

$$F_s = t_{\text{total}} / t \quad (14)$$

其中, t 为当前的时刻距离初始时刻的时间长度; s 表示进行统计的节点。用 $F_s \in (0,1]$ 来表征节点的拥塞情况。 F_s 存储于每个节点中, 作为节点的一个属性随着时间根据式(14)进行实时更新。 F_s 越小节点发生拥塞的可能性越大, F_s 越大节点发生拥塞的可能性越小, F_s 称为反拥塞概率^[9]。

3.4 信息素浓度的动态更新机制

在容迟网络中需要实现信息素在网络中的局部更新和全局更新。

(1)局部更新: 网络中某一节点产生一个报文之后, 报文在没有到达信宿节点之前, 网络中的信息素浓度都在进行局部的更新, 它的更新与概率路由中概率的更新一致, 可以使用式(1)~式(3)进行信息素浓度的局部动态的更新。

(2)全局更新: 当某报文到达信宿节点之后, 信宿节点会得知该报文经过的节点, 根据节点概率表中的数据使用式(11)得到贝叶斯因子。信宿节点与途经节点中的某一个节点再次相遇时, 根据贝叶斯因子对相应节点的信息素浓度进行全局更新:

$$P(u, v) = P(u, v)_{\text{old}} + (1 - P(u, v)_{\text{old}}) \times P_{\text{init}} \times \xi^B \quad (15)$$

其中, $P(u, v)$ 表示节点当前信宿节点与途经节点之间的相遇概率; u 表示当前信宿节点; B 是关于对端的途经节点 v 的贝叶斯因子; ξ 是一个正向趋近于 1 的常量, 这样每次报文到达信宿节点后, 信宿节点与途经节点相关的信息素浓度总是向着增大的方向变化, ξ 的具体数值可以在仿真中进行设定。

3.5 蚁群算法的选择概率计算

基于蚁群算法的概率路由算法中, 节点相遇时根据选择概率判断是否转发报文。由式(4)推断出这种改进算法的选择概率如下:

$$P_{\text{ant}} = P(x_i, y)^\alpha F_i^\theta / \sum_{i=1}^M P(x_i, y)^\alpha F_i^\theta \quad (16)$$

其中, P_{ant} 表示的是蚁群算法中的选择概率; x_i 表示网络中第 i 个节点; y 表示报文的信宿节点; 将一个节点看作一个路径选择, $P(x_i, y)$ 表征路径 x_i 上的信息素浓度; F_i 是 x_i 对应的启发因子, 即式(14)中给出的第 i 个节点 x_i 的拥塞情况; M 是网络中节点的数目; α 是信息素浓度的权重系数; θ 是启发函数的权重系数; P_{ant} 存储在节点的一张概率表中, 随着信息素浓度 $P(x_i, y)$ 和启发因子 F_i 进行更新。

P_{ant} 由信息素浓度 $P(x_i, y)$ 和节点本身的拥塞情况 F_i 决定。在这种算法中, 节点相遇时根据更准确的历史信息判断是否转发报文。当报文递交到信宿节点时, 信息素浓度进行全局更新。贝叶斯因子 B 使得信宿节点中记录的与路径上所有节点相关的信息素浓度增大, 从而增大了选择这条路径的可能性。

4 仿真结果与分析

4.1 仿真参数设置

使用 The ONE(Opportunistic Network Environment simulator)^[10] 仿真器对所提出的方案进行评估。仿真场景大小 $4\,500\text{ m} \times 3\,400\text{ m}$, 在其中放置了 126 个节点。这些节点被分成了 6 个组。仿真中设定报文大小设为 100 KB~200 KB 之间, 报文的 TTL 为 300 min, 初始化常数 $P_{\text{init}}=0.75$, 常数 $\gamma=0.25$, 传递因子 $\beta=0.98$, 在全局更新的式(15)中取 $\xi=1.5$, 信息素浓度的权重系数 $\alpha=7$, 启发函数的权重系数 $\theta=3$, 其他具体参数如表 1 所示。

表 1 仿真参数的设置

节点种类	节点个数	节点移动速度/(m·s ⁻¹)	等待时间范围/s
行人	80	[0.5,1.5]	[0,120]
出租车	40	[2.7,13.9]	[0,120]
有轨电车	2	[7,10]	[10,30]

仿真过程中有 3 个指标需要进行比较:

(1)递交率: 递交成功的报文数量与总的投递报文数量的比值。

(2)平均延迟: 递交成功的所有报文从源节点到信宿节点的时间之和与总的报文个数之比。

(3)开销率: 网络中被中继的报文与成功递交的报文之差再与成功递交的报文之比。

在仿真环境下, 比较蔓延路由(Epidemic)^[11-12]、概率路由(PROPHET)及本文提出的基于蚁群算法的概率路由(PROPHET-ACA)在报文递交率、平均延迟及网络开销率上随时间的变化情况。

4.2 结果分析

如图 1 所示蔓延路由协议的递交率在 10k s 前逐渐上升, 由于报文多副本传输, 10k s 之后有限的缓存空间限制了报文的递交率, 使得报文递交率下降。18k s 后会有一部分报文的剩余 TTL=0 被丢弃, 网络中的拥塞情况得到缓解, 报文递交率小幅度提高, 之后趋于平稳。本文提出的算法在 12k s 之前, 整个网络依据选择概率进行路径探索, 部分报文成功递交到信宿节点, 信息素开始执行全局更新机制, 逐渐找到最佳路径, 此阶段报文的递交率迅速增大。网络中的节点在相遇时综合考虑信息素浓度以及启发因子通过选择概率来决定对报文的转发, 所以 18k s 之后递交率有了明显的提高。在 30k s 之后最佳路径逐渐形成并趋于稳定, 网络中的报文递交率也因此缓慢增长之后渐渐趋于平缓。

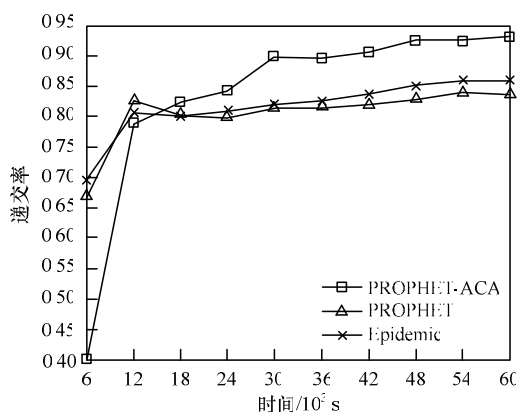


图1 报文递交率随时间的变化

在图2中,开始时蔓延路由协议的平均延迟较小,经过一段时间出现拥塞,许多报文无法递交出去,报文在缓存中排队等待的时间增大,导致蔓延路由的延迟逐渐增大。本文提出的算法中平均延迟大于传统概率路由协议中的平均延迟,报文总是在等待相对更好的节点作为下一个路径选择,导致报文在节点缓存中等待的时间更长,延迟也因此而变大。12k s内处于最佳路径的寻找阶段,得到局部的最优路径,部分报文递交成功。此后信息素执行全局更新,报文在缓存中的等待时间呈现快速的上涨趋势,平均延迟也随之增大。12k s之后逐渐发现全局的最优路径选择,平均延迟继续增加,增长速度明显减慢。30k s之后网络中的最佳路径逐渐形成,网络中的平均延迟在渐渐接近稳定。

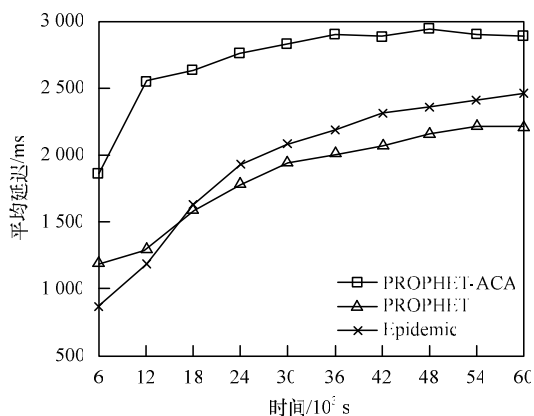


图2 平均延迟随时间的变化

在图3中,前期蔓延路由协议开销率在增大,报文一直在复制转发,网络容易出现拥塞,节点携带的报文增多,两节点之间需要复制转发的报文数目减小,所以开销率会逐渐趋于平稳。本文提出的算法开销率得到了改善,改进了报文的转发机制同时加入了信息素的全局更新机制。根据更为准确的历史信息来判断是否转发报文。报文到达信宿节点,信息素会执行全局更新,报文经过的节点与信宿节点之间的相遇概率呈现增长的趋势,网络经过一段时间找到最佳路径,减少了报文不必要的复制和转发,总体上降低了递交一个报文所消耗代价,所以与传统的概率路由协议相比,这种新的算法开销率明显降低。

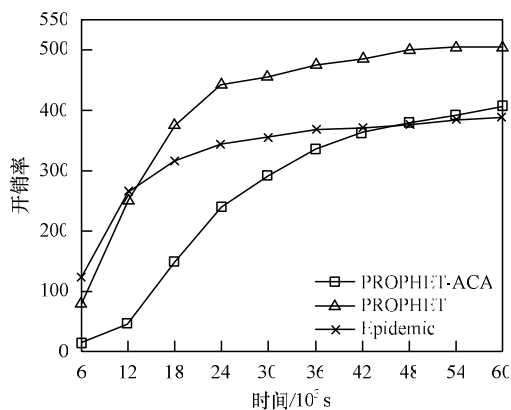


图3 开销率随时间的变化

5 结束语

本文结合蚁群算法的理论改进概率路由协议,提出一种基于蚁群算法的概率路由算法。引入贝叶斯因子,通过贝叶斯因子完成信息素浓度的全局更新。当2个节点相遇时,计算得到一个路径选择概率,根据这个路径的选择概率决定是否转发报文,从而改善DTN的网络性能。仿真结果表明,这种路由算法的报文递交率显著增大,开销率也明显减小。未来将进一步研究各种路由协议的算法和拥塞控制策略,进一步发展和完善DTN的路由协议。

参考文献

- [1] Kevin F. A Delay-tolerant Network Architecture for Challenged Internets[C]//Proc. of ACM Conference on Applications Technologies Architectures and Protocols for Computer Communications. [S. l.]: ACM Press, 2003: 27-34.
- [2] Haykin S. Cognitive Radio: Brain-empowered Wireless Communications[J]. IEEE Journal on Selected Areas in Communications, 2005, 23(2): 201-220.
- [3] Caimi C, Firrincieli R, Livini M. DTN Bundle Layer over TCP: Retransmission Algorithms in the Presence of Channel Disruptions[J]. Journal of Communications, 2010, 5(2): 106-116.
- [4] Lindgreny A, Doria A, Schelén O. Probabilistic Routing in Intermittently Connected Networks[J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2003, 7(3): 19-20.
- [5] Dorigo M, Gambardella L M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem[J]. IEEE Trans. on Evolutionary Computation, 1997, 1(1): 53-66.
- [6] Li Chao, Deng Zhidong. Feedback-enhanced Ant Colony Routing Algorithm for Wireless Sensor Networks[C]//Proc. of Communications and Networking in China. Beijing, China: [s. n.], 2010: 1-5.

(下转第99页)