

Ad Hoc 网中基于熵的长寿分布式 QoS 路由算法*

沈 晖^{1,2+}, 石冰心¹, 邹 玲¹, 石 坚¹, 周建新¹

¹(华中科技大学 电子与信息工程系,湖北 武汉 430074)

²(中兴通讯股份公司,广东 深圳 518057)

A Distributed Entropy-Based Long-Life QoS Routing Algorithm in Ad Hoc Network

SHEN Hui^{1,2+}, SHI Bing-Xin¹, ZOU Ling¹, SHI Jian¹, ZHOU Jian-Xin¹

¹(Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

²(ZTE Corporation, Shenzhen 518057, China)

+ Corresponding author: Phn: +86-27-87552490, E-mail: nickshenhui@yahoo.com.cn, http://www.hust.edu.cn

Received 2003-06-25; Accepted 2004-03-01

Shen H, Shi BX, Zou L, Shi J, Zhou JX. A distributed entropy-based long-life QoS routing algorithm in Ad Hoc network. *Journal of Software*, 2005,16(3):445–452. DOI: 10.1360/jos160445

Abstract: On the basis of analyzing the problem of unicasting QoS routing in Ad Hoc networks, a new distributed QoS routing algorithm in the Ad Hoc network—EBLLD (entropy-based long-life distributed QoS routing) algorithm, is proposed. The key idea of EBLLD algorithm is to construct the new metric-entropy and select the long-life path with the help of entropy metric to reduce the number of route reconstruction so as to provide QoS guarantee in the Ad Hoc network whose topology changes continuously. The EBLLD algorithm uses the local multicast mechanism, orders and sorts the outgoing link with the heuristic function and entropy metric to reduce the message overhead. The simulation shows that the EBLLD algorithm can acquire a higher routing success ratio with the low message overhead. In addition, the EBLLD algorithm is scalable and applicable to large-scale Ad Hoc network.

Key words: Ad Hoc; entropy; long-life; stability; QoS

摘 要: 在分析 Ad Hoc 网的单播 QoS 路由问题的基础上,提出了一种新的 Ad Hoc 网的分布式 QoS 路由算法——EBLLD(entropy-based long-life distributed QoS routing)算法.其核心思想是提出了衡量路径稳定性的新尺度——熵,并利用熵来选择长寿的路径,减少了重建路由(或路由修复)的次数,从而在 Ad Hoc 网的网络拓扑频繁变化的环境中尽可能地提供 QoS 保证.同时该算法还利用本地组播机制和启发函数排序过滤和熵尺度排序过

* Supported by the National Natural Science Foundation of China under Grant No.60172077 (国家自然科学基金); the Sunshine Young Project in Wuhan City of China under Grant No.20015005038 (武汉青年科技晨光计划资助)

作者简介: 沈晖(1975—),男,湖北应城人,博士,主要研究领域为无线网,Ad Hoc 网的 QoS 路由;石冰心(1946—),男,教授,博士生导师,主要研究领域为宽带通信网;邹玲(1968—),女,博士,副教授,主要研究领域为网络管理,路由技术;石坚(1966—),男,博士,讲师,主要研究领域为高速网的 QoS 路由;周建新(1976—),男,博士,主要研究领域为 TCP 性能分析.

滤减小了其路由消息开销.仿真结果表明,EBLLD 算法能够以较小的路由消息开销获得较高的路由成功率.此外,EBLLD 算法具有可扩展性,可以应用于较大规模的 Ad Hoc 网中.

关键词: Ad Hoc 网;熵;长寿;稳定性;QoS

中图法分类号: TP393 文献标识码: A

在 Ad Hoc 网中构造合适的 QoS 路由算法是一个极具挑战性的问题.近年来,国际上提出了一些 Ad Hoc 网的 QoS 路由算法^[1-5],其中性能最好的要算是 Shigang Chen 博士提出的 TBP 算法^[5],并且很少是仅仅利用局部状态信息进行路由选择的完全分布式 QoS 路由算法.同样地,在 Ad Hoc 网中利用位置信息进行路由选择虽然作了一些工作^[6-8],但是将它们应用于 Ad Hoc 网的 QoS 路由中还不多见.Ad Hoc 网的一个最显著的特点是其频繁的流动性,故所选得的满足 QoS 约束的路径可能在数据传送完成之前断掉了,需要重建路由或者进行路由修复.因此在 Ad Hoc 网的 QoS 路由中选择长寿的路径对于在 Ad Hoc 网的环境中能够提供 QoS 保证而言非常重要.但是在长寿(或者稳定)路径的 QoS 路由方面仅有很少一些工作^[9-11],而且这些工作主要是通过节点的移动性预测链路(路径)的寿命,还没有一个很好的表征路径稳定性的尺度.鉴于这种情况,本文提出了一个表征路径寿命(稳定性)的新尺度——熵,并在此基础上提出了一种延时约束最小费用的 Ad Hoc 网中的完全分布式 QoS 路由算法 EBLLD(entropy-based long-life distributed QoS routing)算法,选择长寿(稳定)的路径.

1 网络模型和问题的表述

Ad Hoc 网表示为带权图 $G(V, E)$,其中, V 为 Ad Hoc 网的移动节点集, E 为节点间能够互相通信的双向链路集.在 Ad Hoc 网中 E 是动态变化的. $N(i)$ 是移动节点 i 的邻居节点集.对于 $\forall l(i, j) \in E$, 链路 $l(i, j)$ 的状态信息包括: $l(i, j)$ 的延时 $D(i, j)$ 和费用 $C(i, j)$. 对于任意给定的路径 $P(a, b)$, 路径 $P(a, b)$ 的费用为

$$C_{P(a,b)} = \sum_{l(i,j) \in P(a,b)} C(i,j),$$

路径 $P(a, b)$ 的延时为

$$D_{P(a,b)} = \sum_{l(i,j) \in P(a,b)} D(i,j).$$

延时约束的单播路由优化问题可以表述为:给定图 $G(V, E)$, 信源 s , 信宿 d , 寻找从信源到信宿的路径 p^* , 使得 p^* 满足条件:

$$C_{P^*(s,d)} = \sum_{l(i,j) \in P^*(s,d)} C(i,j) \rightarrow \min \quad (1)$$

$$D_{P^*(s,d)} = \sum_{l(i,j) \in P^*(s,d)} D(i,j) \leq DB \quad (2)$$

其中式(1)描述了优化目标,式(2)描述了 QoS 要求. DB 为业务所允许的延迟的上限值.

此外,本文作以下假设:(1) 在 Ad Hoc 网中的每个移动节点都有一个唯一的标识符;每个移动节点 $v \in V$ 都知道自己所处位置的坐标 $pos(v) = (x(v), y(v))$, 这可以通过在每个节点上安装 GPS(全球定位系统)设备或通过其他的位置定位技术实现^[12].(2) 存在某 MAC 协议^[13], 该协议解决介质竞争, 支持资源预留, 并能保证在局部广播范围内的邻居节点中, 只有想要接受消息的节点保持消息, 其他节点都丢弃消息.(3) 已存在某邻居节点发现协议^[14]: 每个节点周期地广播 BEACON 包来标识自己, 使得任意节点 v 不仅知道它的邻居节点集 $N(v)$, 而且知道它与各邻居节点间的链路的当前状态.

2 熵尺度

在 Ad Hoc 网中, 设计 QoS 路由算法的最重要的一个问题是如何设计能够适应网络拓扑快速改变的 QoS 路由算法. 即希望选择一条稳定的路径或者长寿的路径. 为此, 我们需要提出一种尺度来衡量路径的稳定性或者路径的寿命. Ad Hoc 网又称为自组网, 而每个可变化的自组织系统都有特征变量来表征. 例如一个由运动粒子所组成的系统用位置矢量和速度矢量来表征其状态. 所有这些特征变量合起来就能够确定该系统的特征.

直觉上,我们有理由认为:表征 Ad Hoc 网的状态的因素有 3 个方面:各个移动节点的位置、各个移动节点的速度和各个移动节点的发射距离.令节点 i 的特征变量为 $a_{i,j}$,其中节点 j 是节点 i 的邻居节点;则

$$a_{i,j} = a_{i,j}(pos(i,t), pos(j,t), v(i,t), v(j,t), R_i, R_j) \quad (3)$$

其中 $pos(i,t), pos(j,t)$ 分别是节点 i 和节点 j 在 t 时刻的位置向量; $v(i,t), v(j,t)$ 分别是节点 i 和节点 j 在 t 时刻的速度向量; R_i, R_j 分别是节点 i 和节点 j 的发射距离.

实际上, $a_{i,j}$ 应该与节点 i 和节点 j 的相对速度和相对位置有关,即:

$$a_{i,j} = a_{i,j}(pos(i,j,t), v(i,j,t), R_i, R_j) \quad (4)$$

其中 $pos(i,j,t) = pos(i,t) - pos(j,t)$ 表示 t 时刻节点 i 和节点 j 之间的相对位置向量, $v(i,j,t) = v(i,t) - v(j,t)$ 表示 t 时刻节点 i 和节点 j 之间的相对速度向量.

基于此,我们构造特征向量

$$a_{i,j} = \frac{1}{N} \sum_{i=1}^N \frac{\|pos(i,j,t_i) + v(i,j,t_i) \times \Delta\| - \|pos(i,j,t_{i+1})\|}{R_i} \quad (5)$$

这里, N 是在时间间隔 Δ 内离散时刻 t_i 的个数(每隔一个时间间隔 Δ , 相对位置向量和相对速度向量被计算和更新). 基于以上定义,我们定义熵:

$$H_i(t, \Delta) = \frac{-\sum_{k \in F(i)} P_k(t, \Delta) \log P_k(t, \Delta)}{\log C(F(i))} \quad (6)$$

这里,

$$P_k(t, \Delta) = \frac{a_{i,k}}{\sum_{l \in F(i)} a_{i,l}} \quad (7)$$

$F(i)$ 表示节点 i 的邻居节点集或者其子集, $C(F(i))$ 表示集合 $F(i)$ 的势. 如果我们想计算局部网络的稳定性(节点 i 附近的), 那么 $F(i)$ 就可以设为节点 i 的邻居节点集; 若我们仅对某一特定路径的一部分的稳定性感兴趣, 那么 $F(i)$ 就可以设为那条路径上的节点 i 的两个邻居节点.

从以上讨论中, 我们可以看出: 我们所构造的熵函数 $H_i(t, \Delta)$ 并不是统计物理学或者信息论中严格意义下的熵. 而只是借助于熵的意义和形式所构造的能够表征路径稳定性的尺度. 此外, 根据此定义, 给定区域变化越剧烈, $H_i(t, \Delta)$ 越小.

对于从节点 i 到节点 j 的路径 P , 其熵定义为

$$H_{P(i,j)} = \prod_{k=1}^{N_p} H_k(t, \Delta) \quad (8)$$

这里, N_p 表示从节点 i 到节点 j 的路径 P 上的节点数.

由此定义可知: 路径 $P(i,j)$ 越稳定, 则 $H_{P(i,j)}$ 越大. 我们再令

$$H'_i(t, \Delta) = -\ln H_i(t, \Delta) \quad (9)$$

即路径 $P(i,j)$ 越长寿(稳定), 则

$$-\ln H_{P(i,j)} = -\sum_{k=1}^{N_p} \ln H_k(t, \Delta) = \sum_{k=1}^{N_p} H'_k(t, \Delta) \quad (10)$$

越小. 即通过取负对数, 我们得到了表征路径稳定性的加性尺度——熵尺度为

$$H'_i(t, \Delta) = -\ln H_i(t, \Delta).$$

3 QoS 路由选择算法

首先我们定义两个操作:

定义 1. 排序操作 \prec : 对于给定的序列 $S = \{s_1, s_2, \dots, s_n\}$, 其对应的关键字序列为 $\{h(s_1), h(s_2), \dots, h(s_n)\}$, 则

$$\prec_h(S) = \{s_{p_1}, s_{p_2}, \dots, s_{p_n}\}$$

使得

$$h(s_{p_1}) \leq h(s_{p_2}) \leq \dots \leq h(s_{p_n}),$$

其中 $\{s_{p_1}, s_{p_2}, \dots, s_{p_n}\}$ 是 $\{s_1, s_2, \dots, s_n\}$ 的一种排列.

定义 2. 过滤操作 Fil: 对于一个给定的按关键字 $h(s_i)$ 有序序列 $S = \{s_1, s_2, \dots, s_n\}$, 其中

$$h(s_1) \leq h(s_2) \leq \dots \leq h(s_n).$$

那么 $Fil_h(S) = \{s_1, s_2, \dots, s_\tau\}$, 其中 $\tau = \left\lceil \left(\frac{n}{2} \right) \right\rceil$ 表示大于 $\left(\frac{n}{2} \right)$ 的最小整数.

为了减小路由消息开销, 本路由算法使用“本地组播”机制. 在一个 Ad Hoc 网中, 每个消息在共享介质中通过本地广播进行传递. 消息要么是一个本地广播消息, 即其每个邻居节点都是接受者; 或者是一个单播消息, 即它是发往某一个接受者, 而被所有其他的节点所忽略. “本地组播”就是允许邻居节点的一个子集作为接受者. 在当前的 Ad Hoc 网中还并不存在“本地组播”的硬件支持, 但是我们可以利用本地广播消息来进行软件实现. 考虑某个节点 i , 令其接受节点集为 Ω_i . 节点 i 并不是发送单播路由消息到 Ω_i 中的每个节点, 而是构造一个本地广播路由消息, 这个本地广播路由消息有一个附加域来指明 Ω_i . 这个路由消息被广播到所有邻居节点, 每个邻居节点处理收到的路由消息并且检查它是否属于 Ω_i , 如果属于 Ω_i , 则此节点接受它, 否则丢弃它. 这样, 通过“本地组播”, 每个节点最多发送一个路由消息而不是对每个出口链路发送一个单播路由消息.

QoS 路由算法具体如下:

每当收到连接请求, 源节点 s 发起路由过程. 每个连接请求包含如下信息 $(s, pos(s), d, pos(d), cid, DB)$, 其中 $s, pos(s), d, pos(d), cid, DB$ 分别是源节点、源节点的位置向量、目的节点、目的节点的位置向量(目的节点的位置信息可以通过位置服务机制得到^[6])、连接 id 号和延迟约束. 源节点 s 首先构造路由消息 *Discovery Message* $(s, pos(s), d, pos(d), cid, DB, hop_initial, hop_count, Delay_so_far, Path_so_far, Entropy_so_far)$; 其中 $hop_initial$ 是指所允许最大跳数的初始值, 设为 $\xi \frac{dist(s, d)}{R_s} + \eta$, 其中 ξ, η 是两个系统参数(仿真表明, 尽管随着节点密

度的不同, ξ, η 有不同的最佳值, 但是当 $\xi=3, \eta=3$ 时, 一般能够得到令人满意的路由成功率和花费相对较小的路由消息开销); hop_count 是指该路由消息所允许的最大跳数的当前值; $Path_so_far$ 是指该消息迄今为止所经历的路径; 初值设为源节点 id; $Entropy_so_far$ 是该路由消息迄今为止所经历的路径的熵(是指 $\sum_{k=1}^{N_r} H'_k(t, \Delta)$), 其初值设为 0. 源节点计算 $\Omega_s = Fil_{h'}(\prec_{h'}(Fil_h(\prec_h(N'(s))))))$, 其中 $N'(s) = \{k | Delay_so_far + D(i, k) \leq DB, k \in N(s)\}$, h 为启发式函数, 可以有多种选择, 如: $h(i, j) = dist(j, d)$ (在仿真中, 选择这种形式的启发式函数). $h(i, j) = dist(j, d) \times cost(i, j)$,

$$h(i, j) = \begin{cases} \frac{dist(j, d) \times cost(i, j)}{DB - [Delay_so_far + D(i, j)]} & Delay_so_far + D(i, k) \leq DB \\ \infty & \text{elsewhile} \end{cases}$$

然后, 源节点 s 按计算所得的 Ω_s 域中的值将路由消息本地组播到 s 的部分邻居节点(在 Ω_s 中的节点)中去. 任何一个中间节点 i 从节点 j 收到路由消息 *DiscoveryMessage* $(s, pos(s), d, pos(d), cid, DB, hop_initial, hop_count, Delay_so_far, Path_so_far, Entropy_so_far)$ 后, 它首先检查以前是否收到过该消息, 若收到过该消息, 则丢弃该路由消息; 若没有收到过该路由消息, 则检查本节点是否存在于 $Path_so_far$. 若本节点在 $Path_so_far$ 中, 则丢弃该路由消息; 若本节点不在 $Path_so_far$ 中, 则检查 $(hop_count-1)$ 是否大于 0, 若 $(hop_count-1)$ 的值不大于 0, 则丢弃该路由消息; 若 $(hop_count-1)$ 的值大于 0, 则更新路由消息为 *DiscoveryMessage* $(s, pos(s), d, pos(d), cid, DB,$

$hop_initial, \min\{(hop_count-1), (hop_initial \left(1 - \frac{dist(s, j)}{dist(s, d)}\right) - 1)\}, (Delay_so_far + D(j, i)), (Path_so_far + i), (Entropy_so_far + H'_i(t, \Delta))$), 并且计算 $\Omega_i = Fil_{h'}(\prec_{h'}(Fil_h(\prec_h(N'(i))))))$, 其中 $N'(i) = \{k | Delay_so_far + D(i, k) \leq DB$, 并且

$k \neq j, k \in N(i)\}$, 然后按计算所得的 Ω_i 将上述路由消息本地组播出去. 以上过程重复进行, 直到路由消息到达目的节点 d . 若目的节点 d 收到第 1 个路由消息, 则它等待一个预定的时间间隔 t_d . 目的节点 d 从它在此时间间隔 t_d 内所收到的路由消息中选择一个 $Entropy_so_far$ 域的值最小的路由消息, 以此路由消息的 $Path_so_far$ 域中所记录

的路径作为从源节点 s 到目的节点 d 的路由路径.然后目的节点沿着此路径的逆向路径向源节点 s 发送路由证实消息,并且沿途预留资源.当路由证实消息到达源节点后,则该连接已成功地建立了路由,可以开始数据传送过程了.

4 性能分析和正确性分析

下面将给出本路由算法的性质及其证明:

性质 1. 本算法所选得的路径是无环的.

证明:令 $DiscoveryMessage(s, pos(s), d, pos(d), cid, DB, hop_initial, hop_count, Delay_so_far, Path_so_far, Entropy_so_far)$ 为目的节点 d 所收到的熵最小的路由消息, $Path_so_far$ 为其所经过的路径.由上节中的算法描述可知:目的节点选 $Path_so_far$ 所代表的路径作为其路由路径.如果 $Path_so_far$ 所代表的路径有环,则 $Path_so_far$ 上肯定有一个节点 $i(i \neq d)$ 收到并且转发路由消息 $DiscoveryMessage$ 两次.这与本算法的构造相冲突.本算法要求每个节点收到路由消息后首先检查自己是否在 $Path_so_far$ 中,若在其中,则丢弃该消息.因此 $Path_so_far$ 所代表的路径肯定是无环的. \square

性质 2. 本算法最坏情况下的时间复杂度是 $O(2|V|+t_d)$, 最坏情况下的消息复杂度是 $O(|E|+|V|)$, 其中 $|V|$ 是 Ad Hhoc 网的节点数, $|E|$ 是其链路数.

证明:令 l 为所选路径的长度.由于本算法需要一个消息花费一个来回的时间加上目的节点收集路由消息所等待的时间 t_{wait} 来建立一条连接.若我们假定在通常的条件下,一个消息花费一个单位时间穿过一条链路(包括节点的缓冲时间和协议处理时间),则本算法时间复杂度为 $O(2l+t_{wait})$ 单位时间.而任何简单路径的最大长度是 $|V|-1$, 并且目的节点的等待时间总是小于等于 t_d .故本算法最坏情况下的时间复杂度为 $O(2|V|+t_d)$.

此外,因为本算法每条链路最多发送一个路由发现消息(由上节中的算法描述可知),所以对于一个单连接请求而言,本算法最坏情况下的路由发现消息数为 $O(|E|)$.此外,路由证实消息数应为 $O(l)$, 而任何简单路径的最大长度是 $|V|-1$, 故本算法最坏情况下的路由证实消息数为 $O(|V|)$.因此,总的来说,本算法最坏情况下的消息复杂度为 $O(|E|+|V|)$. \square

5 仿真实验

为了评价 EBLLD 算法,我们进行了广泛的仿真实验,对如下的 4 个性能尺度:BEACON 持续滴答数、路由重建次数、路由成功率、平均消息复杂度进行了检验.它们的定义如下:

BEACON 持续滴答数是指在两个相连的路径错误之间的平均 BEACON 信号数(在仿真中, BEACON 周期取为 0.1s, 且假设:如果在 0.5s 以后没有收到 BEACON 信号,就认为那个链路断开了).

路径重建次数是指数据连接请求成功建立路由到完成数据传送之间所需重建路由的次数(在本仿真中,假设在出现路径错误之后,就重建路由).

$$\text{路由成功率} = \frac{\text{成功建立的连接数}}{\text{总的连接请求数}},$$

$$\text{平均消息复杂度} = \frac{\text{发送的消息数}}{\text{总的连接请求数}}.$$

对于消息数我们是这样规定的:在一条链路上发送一个路由消息,消息数被计为 1.因此一个路由消息经过一个 l 跳的路径,则消息数要计为 l .

本仿真实验的硬件环境为:毒龙 650 微机, 2M Cache 和 256M 内存.所用的仿真软件是我们在 QoS 路由由仿真软件 MCRSIM(由 Hussein Salama 开发)的基础上所开发的 QoS 路由由仿真软件 QNRS(该软件用 C++ 编写,采用 OSF/Motif 编写图形用户界面,运行于 Linux 平台).在我们的仿真实验中,网络拓扑是随机生成的.40 个移动节点被随机地放在 $18 \times 18 \text{m}^2$ 的区域中;节点的发射范围被限制在以 r 为半径的圆内, r 上均匀分布于 $[0.5, 3] \text{m}$. 节点的移动速度均匀分布于 $[0.5, 3] \text{m/s}$. 当两个节点处在彼此的发射范围内时,彼此之间就存在一条链路.节点的平均度

数是 4.2.每个连接请求的源节点、目的节点、延时约束被随机地产生.延时约束均匀分布于[30,160]ms.每条链路
 的费用均匀分布于[0,200],每条链路的时延均匀分布于[0,50]ms.每条连接的持续时间服从指数分布,其平均
 持续时间为 5s.

我们将本文所建议的 EBLLD 算法和 TBP 算法(基于票据的路由探针算法)^[5]、Flooding 算法(泛洪法)、Paul
 算法^[10]进行了比较.具体的仿真结果如图 1、图 2 所示(图中的每个点都是 500 个独立的随机产生的连接请求的
 运行结果的平均值).

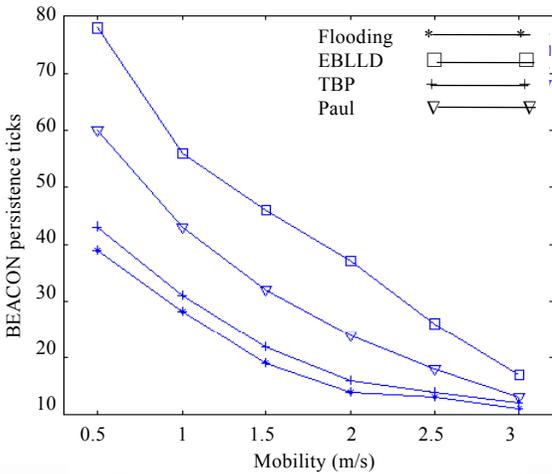


Fig.1 BEACON persistence ticks against mobility

图 1 BEACON 持续滴答数

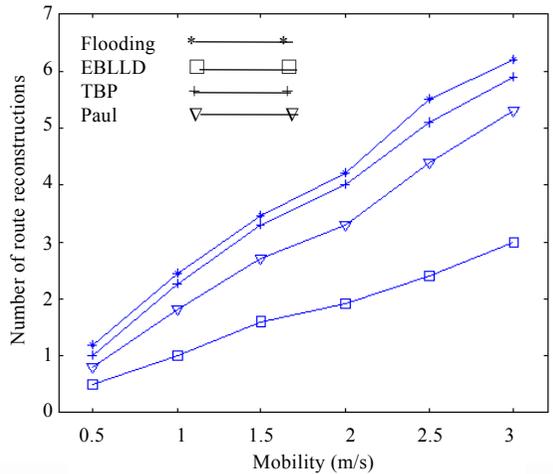


Fig.2 Number of route reconstructions against mobility

图 2 路径重建数

图 1 给出了 4 种算法的 BEACON 持续滴答数的仿真结果.我们测量一直到由于节点移动而引起的路径错
 误为止所选路径上的节点产生 BEACON 信号持续了多长时间(即 BEACON 持续滴答数).显然, BEACON 持续
 滴答数是路径稳定性的一种度量尺度.由图中可以看出, BEACON 持续滴答数随着节点移动率的增大而减
 小. EBLLD 算法的 BEACON 持续滴答数最高, Paul 算法次之, Flooding 算法最低. 这表明:(1) 本文所提出的熵尺
 度的确是衡量路径寿命(稳定性)的尺度.(2) EBLLD 算法能够选出长寿(即稳定)的满足 QoS 要求的路径.

图 2 给出了 4 种算法的路径重建次数的仿真结果.本仿真中假设当出现由于节点移动性所引起的路径错误
 时就重建路由(实际上,更有效的方式是进行路径修复).因此路径重建的次数也在一定程度上表征了路径的稳
 定性.由图中可知:EBLLD 算法的路径重建次数最低, Paul 算法的路径重建次数低于 TBP 算法的路径重建次
 数, Flooding 算法的路径重建次数最高.这也表明 EBLLD 算法所选的路径是最稳定的.

图 3 给出了 4 种算法的路由成功率的仿真结果.图 4 给出了 4 种分布式算法的消息复杂度的比较.

可以看出, Flooding 算法的路由成功率最高, EBLLD 算法的路由成功率高于 TBP 算法, 远高于 Paul 算法.我
 们认为:尽管 TBP 算法能够处理不精确信息,但是 TBP 算法却仍然需要维持某些全局状态,是一个不完全分布
 式的路由算法,而 EBLLD 算法则是一种完全分布式路由算法,不需要维持全局状态,仅仅依靠局部信息选择路
 由,因此在 Ad Hoc 网的高度动态的环境中,能够取得比 TBP 算法更好的路由成功率.

可以看出, EBLLD 算法的消息复杂度略低于 TBP 算法,而远远低于 Paul 算法和 Flooding 算法的消息复杂
 度.这主要是由于 EBLLD 算法采用了本地组播、最大跳数限制、两次排序过滤等多种机制减少了路由消息
 开销.

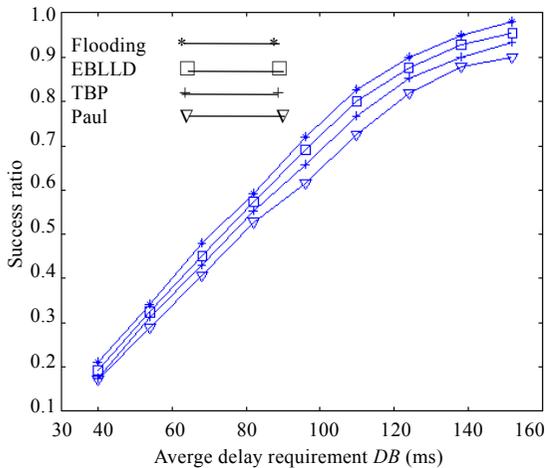


Fig.3 Routing success ratio

图3 路由成功率

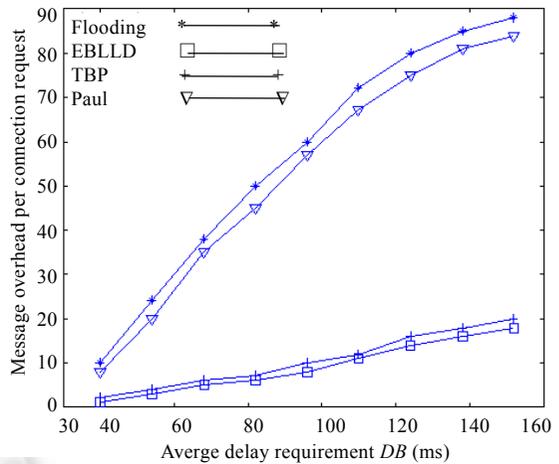


Fig.4 Message overhead

图4 消息数复杂度

6 结论

本文提出了一种延时约束最小费用的 Ad Hoc 网中的完全分布式路由算法——EBLLD 算法。EBLLD 算法的基本思想是提出了一种衡量路径寿命(路径稳定性)的新尺度——熵;利用所提出的熵尺度,通过两次排序过滤,采用最大跳数限制、本地组播等机制,寻找满足延时约束的长寿的路径。EBLLD 算法最坏情况下的时间复杂度是 $O(2|V|+t_d)$,最坏情况下的消息复杂度是 $O(|E|+|V|)$ 。仿真结果表明,EBLLD 算法能够以较小的路由消息开销取得较高的路由成功率。其路由成功率比 TBP 算法更高;而且 EBLLD 算法总能够找到长寿的(即稳定的)路径。EBLLD 算法由于其线性的消息复杂度和时间复杂度以及其仅利用局部状态信息分布式地进行路由选择的特征,使其具有可扩展性,可以应用于较大规模的 Ad Hoc 网中。同时,EBLLD 算法适合于 Ad Hoc 网的高度动态的环境,并且容易扩展到其他 QoS 约束的路由问题。

致谢 在此,我们对本文的工作给予支持和建议的同学和老师表示感谢。

References:

- [1] Royer E, Toh CK. A review of current routing protocols for Ad-Hoc mobile wireless networks. *IEEE Personal Communications Magazine*, 1999,6(2):46–55.
- [2] Iwata A, Chiang CC, Pei G, Gerla M, Chen TW. Scalable routing strategies for Ad Hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 1999,17(8):1369–1379.
- [3] Sivakumar R, Sinha P, Bharghavan V. CEDAR: A core-extraction distributed Ad Hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 1999,17(8):1454–1465.
- [4] Lin CR, Liu JS. QoS routing in Ad Hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 1999,17(8):1426–1438.
- [5] Chen S, Nahrstedt K. Distributed quality-of-service routing in Ad Hoc networks. *IEEE Journal on Selected Areas in Communication*, 1999,17(8):1488–1505.
- [6] Mauve M, Widmer A, Hartenstein H. A survey on position-based routing in mobile Ad Hoc networks. *IEEE Network*, 2001,15(6): 30–39.
- [7] Jain R, Puri A, Sengupta R. Geographical routing using partial information for wireless Ad Hoc networks. *IEEE Personal Communications*, 2001,8(1):48–57.
- [8] Ko Y, Vaidya NH. Location-Aided routing (LAR) in mobile Ad Hoc networks. In: *Proc. of the MOBICom'98*. 1998. 66–75

- [9] Kim D, Toh CK, Choi Y. Location-Aware long-life route selection in wireless Ad Hoc networks. Electronics Letters, 2000,36(18):1584-1586.
- [10] Paul K, Bandyopadhyay S, Mukherjee A, Saha D. A stability-based distributed routing mechanism to support unicast and multicast routing in Ad Hoc wireless network. Computer Communications, 2001,24(18):1828-1845.
- [11] Beongku A, Papavassiliou S. An entropy-based model for supporting and evaluating route stability in mobile Ad Hoc wireless networks. IEEE Communications Letters, 2002,6(8):328-330.
- [12] Capkun S, Hamdi M, Hubaux J. Gps-Free positioning in mobile Ad Hoc networks. In: Proc. of Hawaii Int'l. Conf. System Sciences. 2001. 3481-3490.
- [13] Tsai J, Gerla M. Multicenter, mobile, multimedia radio network. ACM Baltzer Journal of Wireless Networks, 1995,1(3):255-265.
- [14] Toh CK. Associativity-Based routing for Ad Hoc mobile networks. Wireless Personal Communications, 1997,4(3):103-139.

第 1 届中国分类技术及应用研讨会(CSCA 2005)

征文通知

2005 年 9 月 23-25 日 北京

CSCA 2005 由中国计算机学会人工智能与模式识别专业委员会主办, 由北京交通大学承办。分类是知识处理的基本问题, 本次会议旨在推动分类技术研究及相关应用的发展, 促进相关科技单位和个人的科技合作和学术交流, 以及探讨分类与数据分析技术的研究与应用所面临的挑战性问题及关键性研究课题。会议录用论文将由《计算机研究与发展》(正刊, 增刊)正式出版, 会议还将评选大会优秀论文和研究生优秀论文。我们诚征有关分类和数据分析领域的最新创新性成果, 包括分类和数据分析的原理、方法、算法以及特定领域的实际应用等。

征稿范围 (不局限于下述范围):

分类技术基础理论: 监督学习, 半监督学习, 聚类技术, PLS 路径建模和分类, 集成分类技术, 多标签分类和 Preference 学习, 多事例分类, Multimode 聚类和降维, 差异性和聚类结构, 分类和聚类算法复杂性.....

领域相关的分类和聚类技术: 数据密集场景中的分类, 文本分类和聚类, Web 页面分类和聚类, 时间序列的分类和聚类, 图像与视频检索, 计算机视觉中的分类, 生物特征识别中的分类.....

分类技术应用: 银行、金融、保险、市场营销、经济分析, 商务智能, 知识工程, 目标识别, 生物信息学、生物统计学, 医药和健康科学, 信息安全.....

投稿要求: (1) 论文应是未发表的研究成果, 论文要求中文, 采用 word 文件排版, 论文请参照《计算机研究与发展》网页“作者须知”中的“最终修改稿要求”(http://crad.ict.ac.cn)书写, 论文格式参考 2005 年第 1 期执行。(2) 会议论文采用网上提交方式, 在提交论文的同时, 必须提交一份投稿声明 (从 http://crad.ict.ac.cn 网站下载), 作者逐一签字后邮寄或传真到大会会务组, 对不提交投稿声明的论文, 会议将不予受理。

重要日期: 截稿日期 2005-04-25

录用通知日期 2005-05-25

论文提交日期 2005-06-10

来稿请寄: 100044 北京交通大学计算机学院 联系人: 田凤占

电话: 010-51688451, 传真: 010-51840526, E-mail: fztian@center.njtu.edu.cn

www.jos.org.cn

www.jos.org.cn