战术MANET中基于多态转移策略的蚁群优化QoS路由算法*

杜青松[1,2],朱 江[1],张尔扬[1]
(1. 国防科技大学 电子科学与工程学院,湖南 长沙 410073;
2. 湖南大学 计算机与通信学院,湖南 长沙 410082)

摘 要:战术MANET的QoS路由计算是一个NP完全问题,可以采用蚁群优化算法来求解。为了提高蚁群优化QoS路由算法的效率,降低时延和网络开销,提出了基于多态转移策略的蚁群优化QoS路由算法(MTS-AQRA)。MTS-AQRA将链路稳定性和路由拥塞度与常规的QoS路由约束条件结合起来,利用多态转移策略产生的多样化路由搜索蚁群和并行路由搜索处理,能够在MANET网络中快速地建立满足业务QoS要求的稳定路由。仿真实验结果表明,MTS-AQRA在分组到达率、端到端时延、网络吞吐量等指标上综合性能优于AODV、AntHocNet、QoS-Aware ACO等路由算法。

关键词:移动自组织网络;蚁群优化算法;QoS路由;多态转移策略
中图分类号:TP393 **文献标志码**:A **文章编号**:1001-2486(2012)01-0107-08

A novel ant-colony optimized QoS routing algorithm based on multiple transferring strategies for tactical MANETs

DU Qingsong[1,2], *ZHU Jiang*[1], *ZHANG Eryang*[1]
(1. College of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China;
2. School of Computer and Communications, Hunan University, Changsha 410082, China)

Abstract: QoS routes searching in tactical MANETs is a NP-complete problem, to which the ant-colony optimization algorithm is an effective solution. To improve the efficiency of ACO based QoS routing algorithms, an ant-colony optimized QoS routing algorithm (MTS-AQRA) based on multiple transferring strategies was proposed. MTS-AQRA integrates link's stability and route's congestion into conventional QoS requirements in the process of routing. By utilizing the diversity of route-search ants and the concurrent route-searching result from the multiple transferring processes, MTS-AQRA can work rapidly and effectively to establish stable routes which satisfy the QoS requirements of certain services. Simulation results show that MTS-AQRA outperforms AODV, QoS-Aware ACO and AntHocNet in terms of packet delivery ratio, end-to-end delay, end-to-end throughput and route stability.

Key words: mobile Ad hoc network(MANET); Ant Colony Optimization; QoS routing; multiple transferring strategies

战术MANET是为满足数字化战场通信需求而提出来的应用于野战通信环境下的移动自组织网络(Mobile Ad hoc Network)。随着现代化战争发展的需要,通过战术MANET网络传送多媒体信息已经成为迫切的需求[1-2],因此要求战术MANET网络能够为多媒体业务提供合适的QoS支持。

QoS路由算法是MANET网络中一种有效的QoS保障机制[1],而战术MANET网络的QoS路由计算是一个典型的满足多约束条件的多目标优化问题。理论分析和仿真实验表明[3],满足m个加性约束条件和n个乘性约束条件的QoS路由计算是一个NP完全问题,传统的路由算法难以有效解决,一般要采用启发式算法来求解。

蚁群优化算法是意大利学者Dorigo通过模拟自然界蚂蚁觅食行为而提出的一种启发式算法,是解决NP-完全问题的有效方法,已被应用于解决MANET网络的QoS路由问题[4]。近年来,一些基于蚁群优化的MANET网络QoS路由算法已经被提出,如PACO[5]、QoS-Aware ACO[6]、ARMAN[7]、AntNet[8]、AntHocNet[9]、HMAnt-QoS[10]、APAS[11]等,但这些算法普遍存在以下两个局限性:

1)算法在运行时,是通过多批次释放多只蚂

* **收稿日期**:2011-07-07
基金项目:国家部委资助项目
作者简介:杜青松(1971-),男,湖南宜章人,讲师,博士研究生,E-mail:keithdu@263.net;
张尔扬(通信作者),男,教授,博士生导师,E-mail:jiangzhu@nudt.edu.cn

蚁进行路由搜索来模拟蚂蚁的觅食过程,这个搜索过程需要较长的时间,导致路由建立时间过长,不利于传输具有较强实时性要求的业务;甚至可能会由于网络拓扑不停变化导致算法不能收敛,不能得到符合要求的最优路径。

2) 由于没有考虑链路的稳定性,得到的最优路径中有可能包含不稳定链路,从而使得整条路径的生存时间受到影响,导致频繁的路由重建操作,引起更大的时延和网络开销。

针对以上局限性,本文对基本的蚁群优化 QoS 路由算法进行改进,提出了基于多态转移策略的蚁群优化 QoS 路由算法 MTS-AQRA(Ant-Colony Optimized QoS Routing Algorithm based on Multiple Transferring Strategies)。

1 战术 MANET 网络的 QoS 路由模型

1.1 链路生存时间计算方法

链路的生存时间包括两方面因素:根据节点相互运动计算出来的链路维持时间和根据节点剩余能量计算出来的节点生存时间,链路生存时间应该取二者的最小值。

1) 节点生存时间计算方法

假设战术 MANET 中的节点 n 能够精确地测量电池的剩余电量 $E(n)$,并且能够根据自己的收发数据包的统计规律计算出平均功率 $P(n)$,则节点的生存时间为:$life\_time(n)=E(n)/P(n)$。

2) 链路维持时间计算方法

假设战术 MANET 中每一个战术节点都配备了卫星定位系统,能够实时获取卫星定位系统的时间和自己的地理位置;另外,假设节点都按 Random Way Point 模型运动,每一个节点的最大通信覆盖半径为 R,所有的链路均为双向链路。

每个节点都会周期性地向相邻节点发送自己的最新坐标、最新时标以及最新的节点生存时间,据此可以很方便地预测出每条链路的维持时间。

设节点 n 的一个邻居节点 m 前后两次传输给 n 的时标和坐标分别为 $\{t_1,(x_1,y_1)\}$ 和 $\{t_2,(x_1,y_1)\}$,并设 n 在 t_1 时刻的坐标为 (x_{01},y_{01}),在 t_2 时刻的坐标为 (x_{02},y_{02}),则可以在以节点 n 为坐标原点的极坐标系中表示节点 n 和节点 m 的运动关系。如图 1 所示,图 1(a)为两节点相对运动(先接近再远离)的情况,图 1(b)则为两节点反向运动(远离)的情况。

在 t_1 和 t_2 时刻,节点 m 在节点 n 的极坐标系的坐标分别为 (r_1,θ_1) 和 (r_2,θ_2),其值分别为

$$r_1=\sqrt{(x_1-x_{01})^2+(y_1-y_{01})^2},\theta_1=\arcsin(y_1/r_1)$$

$$r_2=\sqrt{(x_2-x_{02})^2+(y_2-y_{02})^2},\theta_2=\arcsin(y_2/r_2)$$

在图 1(a)中,A 点表示节点 m 在 t_1 时刻的位置,B 点表示 m 在 t_2 时刻的位置,线段 AC 表示 m 相对于节点 n 的运动轨迹,箭头表示 m 相对于节点 n 的运动方向。根据前后两个时刻的位置,计算出 A 点到 B 点之间的距离,就可以得出节点 m 相对于节点 n 的运动速度:

$$\angle AOB=\theta_2-\theta_1\triangleq\varphi,L_{AB}=\sqrt{r_1^2+r_2^2-2r_1r_2\cos\varphi}$$

$$v=L_{AB}/(t_2-t_1)$$

设在预测过程中,节点 m 和节点 n 的相互运动关系保持不变(即两者的运动速度、相互运动方向保持不变),则两者之间的链路维持时间可按如下方法进行预测。

在图 1(a)中,节点 m 从当前位置 B 出发,到它脱离节点 n 的通信范围所需要运动的轨迹为线段 BC,则从 t_2 时刻开始,链路还可以维持的时间为

$$\angle OAB\triangleq\theta=\arccos\frac{(r_1^2+L_{AB}^2-r_2^2)}{2r_1L_{AB}}$$

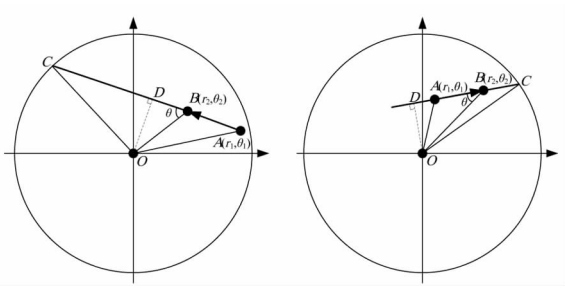
$$L_{BC}=L_{BD}+L_{CD}=(r_1\cos\theta-L_{AB})+\sqrt{R^2-(r_1\sin\theta)^2}$$

$$T_M(e_{nm})=L_{BC}/v$$

$$=\left[(r_1\cos\theta-L_{AB})+\sqrt{R^2-(r_1\sin\theta)^2}\right](t_2-t_1)/L_{AB}$$

$T_M(e_{nm})$ 即为所求得的链路 e_{nm}(节点 n 到节点 m 的链路)的预计维持时间。

图 1(b)的分析方法与图 1(a)相似,结果如下:



(a)节点相对运行 (b)节点反向运动

(a) Nodes move towards (b) Nodes move away

图 1 两节点间的链路维持时间分析示意图

Fig. 1 Analysis of maintaining time between two nodes

$$\angle AOB=\theta_1-\theta_2\triangleq\varphi$$

$$\angle OAB\triangleq\theta=\arccos\frac{(r_2^2+L_{AB}^2-r_1^2)}{2r_2L_{AB}}$$

$$L_{BC}=\sqrt{R^2-(r_2\sin\theta)^2}-(r_2\cos\theta-L_{AB})$$

$$T_M(e_{nm})=L_{BC}/v$$

$$=\left[\sqrt{R^2-(r_2\sin\theta)^2}-(r_2\cos\theta-L_{AB})\right](t_2-t_1)/L_{AB}$$

根据上面的方法,每一个节点都可以预测出

自己到相邻节点的链路维持时间,并与每条链路相应的两个邻居节点的生存时间进行比较,取链路维持时间和链路两个端节点生存时间的最小值作为链路的生存时间 $life\_time(e_{nm})$:

$$lifetime(e_{nm}) = \min\{T_M(e_{nm}), life\_time(n), life\_time(m)\}$$

1.2 QoS 路由模型

我们用有向图 $G=\{V,E\}$ 来表示战术 MANET 的网络模型,其中 V 表示网络中所有的节点集合,E 表示网络中所有节点之间的通信链路集合。链路 $e_{ij}\in E$,表示从节点 V_i 到 V_j 之间的一条链路(设任意两个节点之间最多只有一条通信链路)。

对于任意的链路 $e\in E$,可以定义如下的链路 QoS 参数:时延函数 $delay(e)$,时延抖动函数 $delay\_jitter(e)$,链路生存时间函数 $lift\_time(e)$,带宽函数 $bandwidth(e)$ 和费用函数 $cost(e)$。对于任意节点 $n\in V$,可以定义如下的节点 QoS 参数:时延函数 $delay(n)$,时延抖动函数 $delay\_jitter(n)$,拥塞度函数 $congestion(n)$ 和费用函数 $cost(n)$。

对于一条从源节点 S 和目的节点 D 的路径 $P(S,D)$,该路径的 QoS 参数可表示为

$$delay[P(S,D)] = \sum_{e\in P(S,D)} delay(e) + \sum_{n\in P(S,D)} delay(n)$$

$$delay\_jitter[P(S,D)] = \sum_{e\in P(S,D)} delay\_jitter(e) + \sum_{n\in P(S,D)} delay\_jitter(n)$$

$$bandwidth[P(S,D)] = \min\{bandwidth(e), e\in P(S,D)\}$$

$$life\_time[P(S,D)] = \min\{life\_time(e), e\in P(S,D)\}$$

$$congestion[P(S,D)] = \max\{congestion(n), n\in P(S,D)\}$$

$$cost[P(S,D)] = \sum_{e\in P(S,D)} cost(e) + \sum_{n\in P(S,D)} cost(n)$$

战术 MANET 的 QoS 路由问题就是在网络中寻找一条路径 $P(S,D)$,该路径满足以下约束条件:

(1)带宽约束:$bandwidth[P(S,D)]\geqslant B_q$,$B_q$ 为最小带宽需求;

(2)时延约束:$delay[P(S,D)]\leqslant D_q$,$D_q$ 为最大允许时延;

(3)时延抖动约束:$delay\_jitter[P(S,D)]\leqslant D_J$,$D_J$ 为最大允许时延抖动;

并使得路径 $P(S,D)$ 的总费用最小、拥塞度最小,同时生存时间最长。这是一个多约束条件下的多目标优化问题,适合使用蚁群算法进行求解。

定义待优化的目标函数为:

$$f_1[P(S,D)] = cost[P(S,D)]$$

$$f_2[P(S,D)] = congestion[P(S,D)]$$

$$f_3[P(S,D)] = 1/life\_time[P(S,D)]$$

其中 $f_1[P(S,D)]$ 是路径 $P(S,D)$ 的费用函数,定义为路径 $P(S,D)$ 的跳数;$f_2[P(S,D)]$ 为路径的拥塞度函数;$f_3[P(S,D)]$ 为路径生存时间的倒数。

由此可得 MTS-AQRA 算法多目标优化问题的数学模型描述,如下式所示:

$$\min F[P(S,D)] = \{f_1[P(S,D)], f_2[P(S,D)], f_3[P(S,D)]\}^{\mathrm{T}}$$

$$\text{s.t.}\begin{cases} bandwidth[P(S,D)]\geqslant B_q \\ delay[P(S,D)]\leqslant D_q \\ delay\_jitter[P(S,D)]\leqslant D_J \end{cases}$$

2 路由搜索蚂蚁多态转移策略

2.1 多态转移策略定义

为增加蚁群搜索的多样性,加快搜索速度,增强搜索全局最优的能力,同时为避免蚂蚁的完全洪泛转发所导致的大路由开销, MTS-AQRA 算法中定义了以下五种搜索蚂蚁转移策略,从而形成了蚂蚁状态转移的多态性。

①洪泛转移策略:处于某节点的搜索蚂蚁在选择下一跳节点时,以洪泛的方式向所有的邻居节点转发搜索蚂蚁报文,即把一只搜索蚂蚁复制成多只搜索蚂蚁,增加了并行搜索的蚂蚁数量,有利于加快搜索速度。

②基于稳定度的贪婪转移策略:搜索蚂蚁转移前计算到所有相邻节点的链路生存时间估计值,选择其中链路生存时间最大的链路对应的邻居节点作为下一跳。

③基于拥塞度的贪婪转移策略:搜索蚂蚁转移前计算所有相邻节点的拥塞度,选择其中拥塞度最小的邻居节点作为下一跳。

④轮盘赌转移策略:搜索蚂蚁在选择下一跳节点前,先按下式所示的概率转移规则计算到所有邻居节点的转移概率,根据所有转移概率的大小在(0,1)区间上划分对应大小的子区间,然后产生一个(0,1)区间上均匀分布的随机数,该随机数落在哪个子区间,就选择该子区间对应的节

点作为下一跳节点。

$$P_{R,H}=\begin{cases}\arg\max\limits_{H\in N_R}\{[\tau_{R,H}]^{\alpha}[\eta_{R,H}]^{\beta}\} & q\leqslant q_0, H\in N_R\\ \dfrac{[\tau_{R,H}]^{\alpha}[\eta_{R,H}]^{\beta}}{\sum\limits_{S\in N_R}[\tau_{R,S}]^{\alpha}[\eta_{R,S}]^{\beta}} & q>q_0, H\in N_R\\ 0 & otherwise\end{cases}$$

上式中:R 表示当前节点,H 表示下一跳节点;$\tau_{R,H}$ 是链路 e_{RH} 上的信息素;N_R 是节点 R 的邻居节点集合;α 和 β 是启发因子,分别决定了信息素和启发函数的重要性;q_0 是变异算子,用于防止算法陷入局部最优;$\eta_{R,H}$ 是启发函数,其定义如下式所示:

$$\eta_{R,H}=\frac{1}{cost(e_{RH})}\frac{1}{congestion(R)}life\_time(e_{RH})$$

上式表明,MTS-AQRA 算法倾向于选择具有较小费用、较小拥塞和较大生存时间的链路。

⑤随机转移策略:搜索蚂蚁随机地选择一个邻居节点作为下一跳,选择方法为:设邻居节点数为 M,所有邻居节点从 1 到 M 编号,产生一个在[1,M]区间上均匀分布的随机整数值,选择该值对应编号的邻居节点为下一跳。

2.2 转移策略选择规则

为方便节点灵活地选择搜索蚂蚁的转移策略,特定义以下选择规则:

规则 1 为了在路由搜索的一开始就产生多只寻路蚂蚁,加快路由搜索的收敛速度,MTS-AQRA 算法规定路由建立过程中搜索蚂蚁的前 L_T 级转发节点以概率 1 选择洪泛转移策略,即从源节点开始的前 L_T 级转发节点将蚂蚁洪泛给所有的邻居节点;L_T 的值要根据网络规模来设定。

规则 2 为网络中的所有节点定义一个邻居数目的阈值 N_{Th}[13],不是前 L_T 级的节点收到搜索蚂蚁报文后,检查自己的邻居数目 N,如果 $N<N_{Th}$,则概率 1 选择洪泛转移策略,将蚂蚁洪泛给所有的邻居节点;如果 $N\geqslant N_{Th}$,则按规则 3 选择转移策略;N_{Th}的值要根据网络节点密度来设定:节点密度越大,N_{Th}的值越小[13]。

规则 3 定义五个权值:W_1,W_2,W_3,W_4,W_5($W_1+W_2+W_3+W_4+W_5=1$),分别对应五种蚂蚁转移策略,权值的大小表示对应的搜索蚂蚁转移策略的重要性。将区间(0,1)划分成五个子区间:S_1:(0, W_1],S_2:(W_1, W_1+W_2],S_3:(W_1+W_2, $W_1+W_2+W_3$],S_4:($W_1+W_2+W_3$, $W_1+W_2+W_3+W_4$],S_5:($W_1+W_2+W_3+W_4$, 1),按顺序分别对应五种转移策略。当节点不是前 L_T 级转发节点,并且其邻居数目大于阈值,则产生一个(0,1)区间上均匀分布的随机数,根据随机数落入的子区间号来选择对应的转移策略。

3 MTS-AQRA 算法描述

3.1 路由建立过程描述

当源节点 S 需要往目的节点 D 发送数据时,它检索自己的路由表,如果路由表中不存在 S 到 D 的路由,或已存在路由不满足待传输业务的 QoS 要求,则启动路由建立过程。路由建立过程步骤如下:

步骤 1 源节点 S 构造搜索蚂蚁报文 FANT,并将业务的 QoS 要求填充到 FANT 报文中对应的 QoS 约束指标域中,然后将构造好的 FANT 向邻居节点广播。FANT 报文主要包括如下内容:

- ANT ID:用于识别每一只从源节点发出的搜索蚂蚁;
- 目的节点地址和目的节点序列号,源节点地址和源节点序列号;
- QoS 约束指标值:业务的带宽约束值 B_q、时延约束值 D_q 和时延抖动约束值 D_J;
- 路径上的 QoS 度量值:搜索蚂蚁经过的路径上的 QoS 度量值,包括路径上节点剩余带宽的最小值 $B_{\min}$、时延累加值 D_{accu}、时延抖动最大值 $J_{\max}$、节点拥塞度最大值 $C_{\max}$和链路生存时间的最小值 $LT_{\min}$;
- 中间节点地址列表:FANT 经过的每一个中间节点的地址,按先后顺序添加。

步骤 2 中间节点 R 收到 FANT 报文后,先根据 FANT 中的 ANT ID 和源节点地址判断自己是否已经收到过此 FANT,如果收到过,则丢弃该 FANT,以防止产生路由环路;否则转步骤 3。

步骤 3 R检查自己的地址是否在 FANT 报文的“中间节点地址列表区”中,如果在,则说明自己已经处理过该 FANT,路由搜索过程产生了路由环路,丢弃该 FANT;如果不在,则转步骤 4。

步骤 4 R判断自己是不是目的节点。如果是,转步骤 5;如果不是,则执行以下操作:

① 判断 R 的剩余带宽是否大于等于 FANT 中的带宽约束值。如果不是,则表明自己的带宽资源不满足 QoS 要求,丢弃该 FANT;否则继续执行②。

② 提取 FANT 中的 D_{accu}值,判断 D_{accu}值加上本节点的处理时延是否大于时延约束值。如果是,说明时延不满足业务的 QoS 要求,丢弃该 FANT;否则继续执行③。

③ 判断本节点的处理时延抖动是否小于 FANT 中的时延抖动约束值。如果不是,说明自己的时延抖动不满足业务的 QoS 要求,丢弃该 FANT;否则继续执行④。

④ 为避免节点 R 由于传输过多的数据分组导致拥塞,MTS-AQRA 算法为每个节点定义一个拥塞度阈值。R 判断自己的拥塞程度是否超过预先定义的拥塞度阈值。如果超过,表明要节点接近拥塞,不能接受新的数据传输,丢弃该 FANT;否则继续执行⑤。拥塞度按下式计算:

$$congestion(R,t) = Q(R,t)/Q_{Total}$$

式中 $Q(R,t)$ 为节点 R 在 t 时刻的瞬时 MAC 队列长度,Q_{Total} 为节点的 MAC 队列总长度。

⑤ 将 ANT ID 和源节点地址添加到“蚂蚁 ID 列表”中;同时在本地路由表中添加到源节点的反向路由(如果路由已存在,则更新)。路由表主要包括以下内容:

- 源节点地址和源节点序列号,目的节点地址和目的节点序列号;
- 跳数和下一跳节点地址;
- QoS 约束值:业务的带宽约束值 B_q、时延约束值 D_q 和时延抖动约束值 D_J 值;
- 路由类型:用于区分从源节点到目的节点的多条路由;
- 路由状态:用于表示每条路由的工作状态。

⑥ 更新 FANT 中的跳数计数器 H_c:$H_c \leftarrow H_c + 1$;将本节点的地址和序列号附加在 FANT 的最后面。

⑦ 提取节点 R 的剩余带宽 B_m、处理时延 D_m、处理时延抖动 J_m、拥塞度 C_m 和从上一跳节点到节点 R 的链路生存时间 L_m,按下列方式更新 FANT 中的 $B_{\min}$ 域、D_{accu} 域、$J_{\max}$ 域、$C_{\max}$ 域和 $LT_{\min}$ 域:

$$B_{\min} \leftarrow \min(B_{\min}, B_m), D_{accu} \leftarrow D_{accu} + D_m$$
$$J_{\max} \leftarrow \max(J_{\max}, J_m), C_{\max} \leftarrow \max(C_{\max}, C_m)$$
$$LT_{\min} \leftarrow \min(LT_{\min}, LT_m)$$

⑧ 根据节点 R 的角色按规则 2 或规则 3 选择 FANT 转移策略,将 FANT 广播给邻居节点或单播给选中的下一跳节点,转发的同时对本节点到下一跳节点的链路进行局部信息素更新。如果 H 是选中的下一跳节点,则按式(1)更新链路 e_{RH} 的信息素;否则按式(2)更新:

$$\begin{aligned} \tau_{R,H} &= (1-\rho)\tau_{R,H} + \rho\Delta\tau_{R,H} \\ \Delta\tau_{R,H} &= \ln LT(e_{RH}) + 1/C_m \end{aligned} \quad (1)$$

$$\tau_{R,H} = (1-\rho)\tau_{R,H} \quad (2)$$

其中 ρ 为局部信息素挥发系数。

步骤 5　目的节点收到 FANT 报文后,从中提取相关信息后销毁 FANT,产生全局更新报文 BANT,将 FANT 中的 H_c 值填入 BANT 的 *Hop* 域中,将 FANT 报文中的 $B_{\min}$ 值,D_{accu} 值,$J_{\max}$ 值,$C_{\max}$ 值和 $LT_{\min}$ 值填入 BANT 的相应域中,把中间节点列表反过来附加到 BANT 的最后面,然后将 BANT 报文沿 FANT 来的路径向源节点单播转发,以建立从源节点到目的节点的正向路由。

步骤 6　中间节点收到 BANT 报文后,根据报文中的 $B_{\min}$ 值,D_{accu} 值,$J_{\max}$ 值,$C_{\max}$ 值和 $LT_{\min}$ 值,按照全局信息素更新规则对本节点保存的信息素中相应表项的信息素值进行更新,然后根据中间节点列表将 BANT 报文向源节点方向单播转发。全局信息素更新规则如下式所示:

$$\tau_{i,j} = (1-\delta)\tau_{i,j} + \delta\Delta'\tau_{i,j}$$
$$\Delta\tau_{i,j} = \ln B_{\min} + 1/\ln D_{accu} + 1/\ln J_{\max} + 1/\ln C_{\max} + \ln LT_{\min}$$

其中 δ 为全局信息素挥发系数。

步骤 7　源节点收到 BANT,就建立了源节点到目的节点的正向路由。源节点需要延迟一段时间以收到更多的 BANT,从而得到多条正向路由,然后根据具体的业务对不同 QoS 指标的要求从中得到较优的 M 条路径,选择最优路径作为主路由,其他路由则作为备份路由。

3.2　路由维护过程描述

当路由发生断裂时,MTS-AQRA 算法会立即进入如下所述的路由维护过程:

① 断裂处的上游节点 K 向源节点 S 发送一个 RRI 报文(本地修复通知报文,用于告知源节点路由发生故障正在修复,RRI 报文中包括目的节点和源节点的地址),然后从本地的路由表中查找 S 到 D 的路由表项,从中提取 QoS 约束值,再以 K 为源节点构造一个目的节点为 D 的 FANT 报文,将提取的 QoS 约束值填入 FANT 报文中,然后按照路由建立过程的规则重新搜索从 K 到 D 的最优路由。

② 源节点 S 收到 RRI 后,将正在使用的 S 到 D 的路由(假设为 P_u)设置为故障状态并设置故障恢复定时器,同时启用备用路由 P_b 继续传输数据。

③ 节点 K 到 D 的路由如果重建成功,则向源节点 S 单播一个 BANT 报文;如果重建失败,则向源节点 S 单播一个 RRER 报文(路由故障报文)。

④ 在故障恢复定时器超时之前,源节点 S 收到来自节点 K 的 BANT 报文,就将 P_u 的故障状态清除,并将数据切换到 P_u 上继续传输。

⑤ 如果故障恢复定时器超时,或源节点 S 收

到节点 K 的 RRER 报文,则将 P_u 的路由表项删除。

⑥ 如果 S 的路由表中从 S 到 D 的所有路由都失效,则由 S 重新启动到 D 的路由建立过程。

4 仿真实验和性能分析

4.1 仿真环境设置

我们对 MTS-AQRA 算法进行了仿真试验,并分别就数据分组到达率、端到端时延、归一化路由开销和网络吞吐量四个性能指标与 AODV[12]、AntHocNet 和 QoS-Aware ACO 进行了性能比较。

基本参数设置为:MAC 协议为 IEEE 802.11 DCF;节点传输半径为 250m;节点移动模型为 Random Waypoint;信号传播模型为自由空间传播;节点输出带宽为 2Mb/s;节点队列长度 256KB。

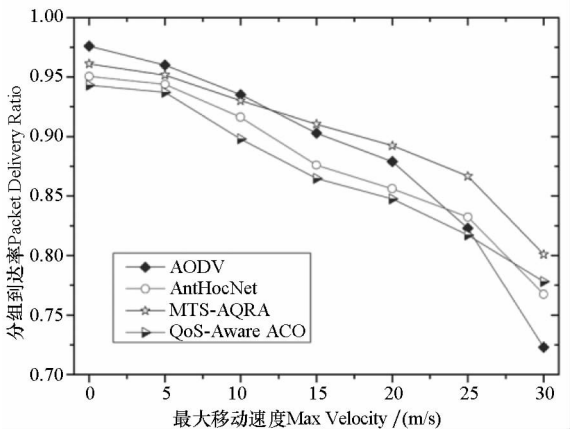
仿真区域大小为 1500m × 1500m,节点数量为 100 个,30 个数据源,数据业务类型为 CBR;源节点发送的数据分组长度为 1024Bytes;分组发送速率为 4 Packets/s。每个节点的平均处理时延设置为 17ms,忽略链路的传输时延。其他参数取值为:各链路的初始信息素值设置为 12;节点拥塞度阈值为 0.8;$\alpha=\beta=2$,$q_0=0.6$,$\rho=\delta=0.85$;五个权值 W_1 到 W_5 均为 0.2;邻居节点数阈值为 $N_{Th}=4$;洪泛转发级别 $L_T=2$。AntHocNet 和 QoS-Aware ACO 算法中的相关参数则分别根据文献[9]和文献[6]进行设置。

仿真实验设置了两个场景:第一个场景不设 QoS 约束,即比较四种算法对于常规数据业务传输的性能;第二个场景传输有 QoS 约束的数据业务。

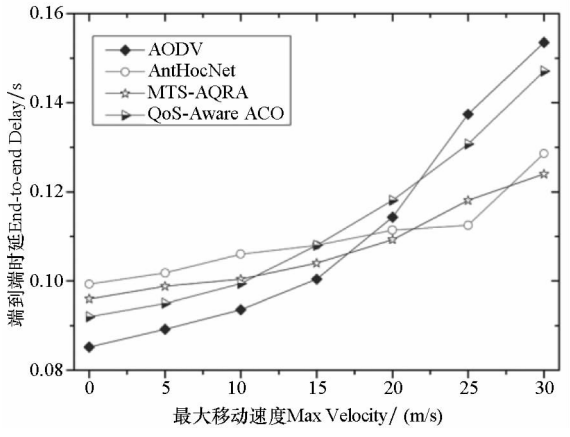
4.2 仿真结果分析

1) 无 QoS 约束场景下的仿真结果分析

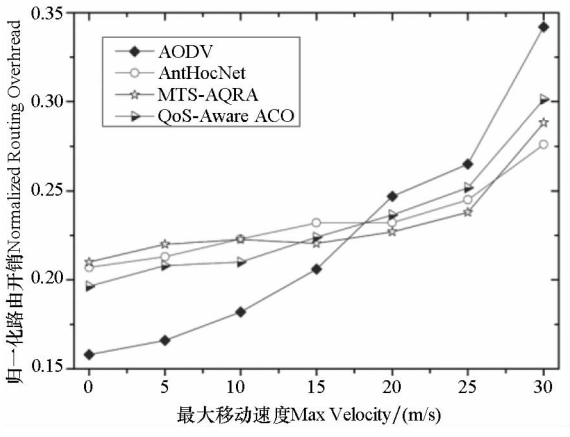
无 QoS 约束意味着带宽约束 $B_q=0$,时延约束 $D_q=\infty$,时延抖动约束 $D_J=\infty$。仿真结果见图 2(a)、图 2(b)、图 2(c)和图 2(d)。



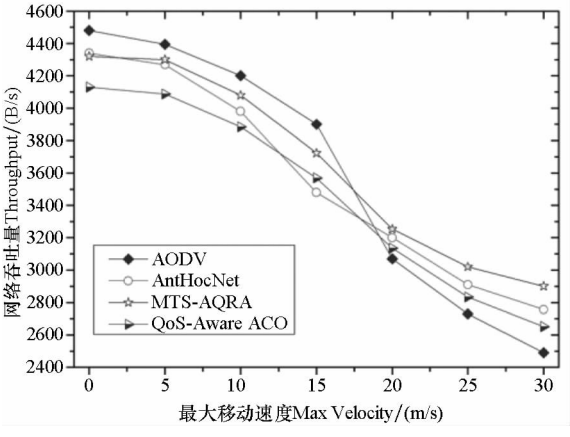
(a) 分组到达率仿真结果
(a) Results of packet delivery ratio



(b) 端到端时延仿真结果
(b) Results of end-to-end delay



(c) 归一化路由开销仿真结果
(c) Results of normalized routing overhead



(d) 网络吞吐量仿真结果
(d) Results of throughput

图 2 无 QoS 约束场景仿真实验结果
Fig. 2 Simulation results for non-QoS scenario

从图 2(a)可以看出,四种算法的分组到达率相差不大。在节点移动速度较低时,AODV 算法的分组到达率略高于其他三种算法。四种算法的分组到达率随节点速度的增加而降低。由于 AODV 没有多径路由的支持,当节点速度增加一定程度时,路由失效率大大增加,AODV 的分组到达率就会低于其他三种有多径路由支持的 ACO 类算法。

图 2(b)则表明,对于无 QoS 要求的业务传输,当节点速度较低时,AODV 算法的端到端时延小于其他三种 ACO 类算法;而当节点速度增加时,由于 AODV 算法要花费越来越多的时间进行路由修复和重路由,因此其端到端时延快速增长,逐渐超过了另三种算法。

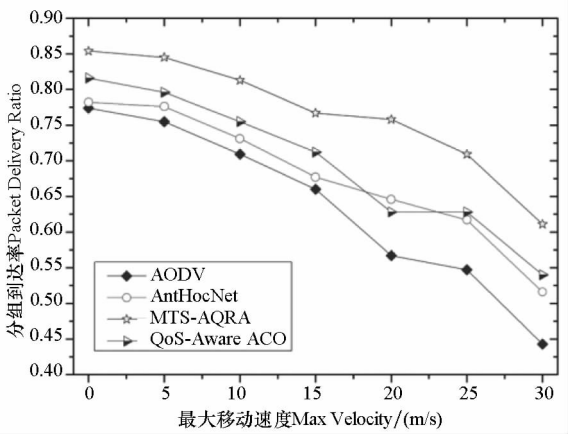
由图 2(c)可知,在没有 QoS 约束的条件下,三种 ACO 类算法的归一化路由开销大体上相当。由于这三种 ACO 类算法的实现复杂度高于 AODV,因此它们的路由协议开销要高于 AODV;但当节点运动较频繁时,AODV 算法的路由失效率要远高于另三种支持多径路由的算法,导致 AODV 算法用于路由修复和重路由的开销大大增加,路由开销渐渐高于另三种算法。

在传输没有 QoS 要求的业务时,AODV 算法的网络吞吐量在节点速度较低时略高于另三种算法;而当节点移动速度达到一定程度后,则会略低于其他三种算法,如图 2(d)所示。

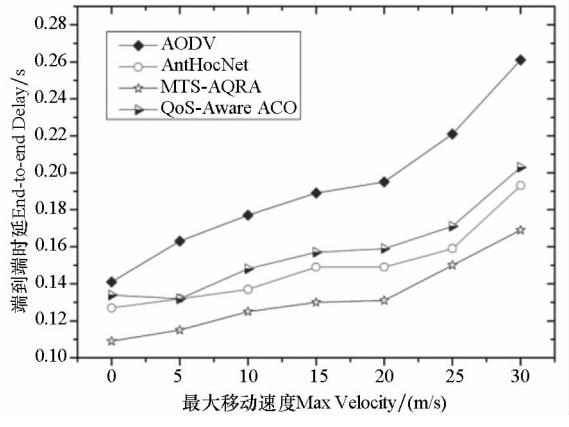
由于 MTS-AQRA 算法在路由过程中优先选择稳定性高而拥塞度小的路由,路由寿命长,路由修复和重路由操作比 AntHocNet 和 QoS-Aware ACO 算法都要少一些,因此在无 QoS 约束条件下,MTS- AQRA 算法的综合性能略优于其他两种 ACO 算法。

2) 有 QoS 约束场景下的仿真结果分析

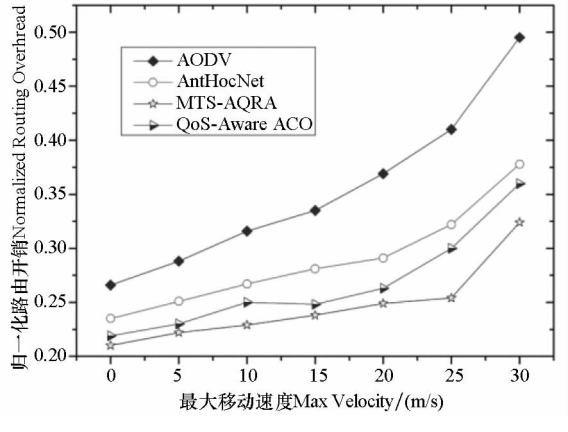
在此场景下,带宽约束 $B_q = 128\text{Kb/s}$,时延约束 $D_q = 200\text{ms}$,时延抖动约束 $D_J = 15\text{ms}$。仿真结果分别见图 3(a)、图 3(b)、图 3(c)和图 3(d)。



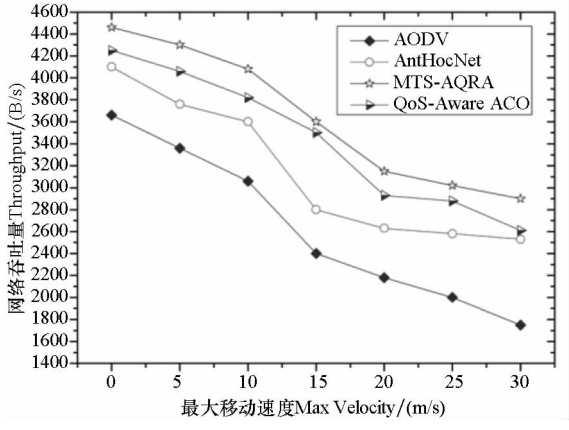
(a) 分组到达率仿真结果
(a) Results of packet delivery ratio



(b) 端到端时延仿真结果
(b) Results of end-to-end delay



(c) 归一化路由开销仿真结果
(c) Results of normalized routing overhead



(d) 网络吞吐量仿真结果
(d) Results of throughput

图 3　有 QoS 约束场景仿真实验结果
Fig. 3　Simulation results for QoS scenario

图3(a)表明,对于有 QoS 约束的业务传输,三种 ACO 类算法的分组到达率明显高于 AODV 算法,这是因为 AODV 建立的路由不考虑 QoS 支持,对于 QoS 业务传输的失败率较高。另外 MTS-AQRA 算法建立的路由考虑了稳定性,并尽量避开了拥塞节点,因此 MTS-AQRA 算法的分组到达率优于 AntHocNet 和 QoS-Aware ACO 算法。

从图3(b)可以看出,随着节点移动速度的提高,四种算法的端到端时延都在增加。由于 AODV 建立的路由不满足 QoS 要求的概率较高,当 QoS 业务发现不能满足要求时,就会要求重新建立路由,导致平均端到端时延随着速度的提高有较大幅度的增加。而三种 ACO 类算法的时延虽然也会增加,但由于平均时延的约束,其端到端时延增加不多。

在图3(c)中,由于 AODV 不能提供 QoS 支持,因此其建立的路由失效率非常高,导致路由开销随着速度的提高而急剧增加;MTS-AQRA 算法由于考虑了 QoS 支持以及路由的稳定性和拥塞回避,而且能够多条备份路由,使得它的路由失效率小于 AODV、AntHocNet 和 QoS-Aware ACO,故其归一化路由开销也是四者中最小的。

图3(d)则表明,四种算法的网络吞吐量都随着节点移动速度的提高而降低。不过 AODV 算法的网络吞吐量降低很多,这是由于有 QoS 约束时,AODV 算法建立的路由有不少因为不能满足 QoS 要求而被废弃,导致重新路由,从而影响了数据分组的传输,大大降低了网络吞吐量。而 MTS-AQRA 能够建立多条较稳定而且拥塞程度小的路由,路由失效率低于 AntHocNet、AODV 和 QoS-Aware ACO,数据传输的连贯性好,因此具有较好的网络吞吐量。

5 结束语

本文采用蚁群优化算法来求解战术 MANET 中多约束条件的 QoS 路由问题,通过在算法中综合考虑路由生存时间和节点拥塞度,提出了基于多态转移策略的蚁群优化 QoS 路由算法 MTS-AQRA。MTS-AQRA 算法能够在网络中快速、并行地搜索多条满足常规 QoS 指标要求,同时又具有较长生存时间和较小拥塞度的路由。仿真实验表明,MTS-AQRA 算法的综合性能优于基于蚁群优化机制的 AntHocNet 算法、QoS-Aware ACO 算法和不支持 QoS 的 AODV 算法,能够为战术 MANET 中多媒体业务的传输提供有效的 QoS 保证。

MTS-AQRA 算法没有考虑在路由过程中进行资源预约,下一步的研究方向是在进行 QoS 路由的同时进行资源预约,进一步提高路由的有效性和可靠性。

参考文献(References)

[1] 郑少仁, 王海涛,等. Ad Hoc 网络技术[M]. 北京:人民邮电出版社, 2005.
ZHENG Shaoren, WANG Haitao, et al. Ad hoc networking technoloby[M]. Beijing: Posts & Telecom Press, 2005. (in Chinese)

[2] Sharret I P. WIN-T-The army's new tactical intranet [C]// Proceedings of IEEE MILCOM 1999, Baltimore, 2: 1383 -1387.

[3] Wang Z, Crowcroft J. Quality of service routing for supporting multimedia applications [J]. IEEE Journal on Selected Areas in Communications, 1996, 14(7): 1228 -1234.

[4] Dorigo M, Stutzle T. Ant colony optimization [M]. Cambridge, MA: MIT Press, 2004.

[5] Liu C Y, Li L Y, Xiang Y. Research of multi-path routing protocol based on parallel ant colony algorithm optimization in mobile ad hoc networks [C]// Proceedings of ITNG, Las Vegas, 2008: 1006 -1010.

[6] Saliba C, Farrugia R A. Quality of service aware ant colony optimization routing algorithm [C]// Proceedings of MELECON, Valetta, 2010: 343 - 347.

[7] Deepalakshmi P, Radhakrishnan S. QoS routing algorithm for mobile ad hoc networks using ACO [C]// Proceedings of INCACEC, Perundurai, India, 2009: 1 -6.

[8] Ahmed T H. Modeling and simulation of a routing protocol for ad hoc networks combining queuing network analysis and ant colony algorithms[D]. Germany: Duisburg-Essen University, 2005.

[9] DiCaro G, Ducatelle F, Gambardella L. AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks[J]. European Transactions on Telecommunications, Special Issue on Self-Organization in Mobile Networking, 2005, 2: 134 -143.

[10] Attia R, Rizk R, Mariee M. A hybrid multi-path ant QoS routing algorithm for MANETs [C]// Proceedings of WOCN'09: Cairo, Egypt, 2009: 1 -5.

[11] Shang F J, Wang Y. An ant system optimization QoS routing algorithm for wireless sensor networks [C]// Proceedings of IWACI, Wuhan, China, 2010: 339 -344.

[12] Perkins C, Belding R E, Das S. Ad-hoc on-demand distance vector (AODV) Routing[S]. IETF RFC 3561, 2003 -07.

[13] Xue F, Kumar P R. The number of neighbors needed for connectivity of wireless networks [J]. Wireless Networks, 2004, 10: 89 -101.