International Journal of Security and Its Applications
Vol.9, No.8 (2015), pp.111-126
http://dx.doi.org/10.14257/ijsia.2015.9.8.09

A Novel Ant Colony Optimization Algorithm for QoS-Based Multicast Trusted Routing in Wireless Ad Hoc Networks

Hui Xia[1,2,3], Xiu-qing Lu[2] and Zhen-kuan Pan[1,2]

[1]*Postdoctoral Research Station of System Science, College of Automation Engineering, Qingdao University, Qingdao, 266071, P.R.C*
[2]*College of Information Engineering, Qingdao University, Qingdao 266071, P.R.C*
[3]*Shandong Provincial Key Laboratory of Software Engineering, School of Computer Science and Technology, Shandong University, Ji'nan 250101, P.R.C*
qduxiahui@gmail.com

Abstract

The wireless ad hoc networks have been attracting increasing attention of researchers owing to its good performance and special application. Multicast routing with QoS multi-constraints problem is a nonlinear combinatorial optimization problem, its goal is to find the minimum cost multicast tree which satisfies multi-constraints such as delay, jitter and bandwidth. Such constrained multicast tree problem is known to be an NP-Hard problem, and there are many heuristic algorithms to solve this problem. However, the execution process of those algorithms is slow and complex, and the security of routing path is not guaranteed. To overcome these shortcomings, we make trust as another QoS constraint, and propose a multicast trusted routing algorithm with QoS multi-constraints basing on a novel ant colony optimization method. In this new algorithm, ants move constantly on the network to find an optimal constrained multicast security tree including source node and destination nodes. Simulation results indicate that our algorithm can quickly find a feasible solution to solve the constrained multicast trusted routing issue. Compared with the conventional ant colony algorithm, the convergence speed and packet delivery ratio of the new algorithm are improved.

Keywords: *wireless ad hoc network; multicast trusted routing; ant colony algorithm*

1. Introduction

A wireless ad hoc network is a set of limited range wireless nodes that function in a cooperative manner. Each node in this network is free to move independently in any direction, and will thus change its links to other nodes frequently. Each must forward traffic unrelated to its own use, and therefore be a router. As the host, the entity needs to run user's application; as the router, it needs to forward data packets according to the routing protocols. The rapid development of wireless network technology gradually enables wireless network services such as video conferencing, image transfer, mobile TV and multi-media applications to become mainstream activities. Multicast is very convenient for those services, which usually have strict QoS (*i.e.*, quality of service) requirements like bandwidth, delay, jitter etc. However, the topology of such network may be highly dynamic due to unpredictable node mobility, which makes QoS-provisioning to applications running inherently hard.

Many useful works have been done on multicast routing. And searching for the shortest path is the routing strategy adopted in conventional ad hoc protocols, e.g., MAODV, ODMRP. Those algorithms have some weak points as spending more time and wasting bandwidth owing to the use of flooding strategy, so the QoS problem is not sufficiently considered in these protocols. Due to the increasing complex applications and inherent

ISSN: 1738-9976 IJSIA

limitations of wireless ad hoc network, multicast routing with guaranteed QoS multi-constraints needs to be achieved better. Previous references have proved that routing with QoS multi-constraints is an NP-Complete problem. The pragmatic way of solving such problem is to apply heuristic algorithm. That is, a less optimal solution which is close to the optimization obtained in certain acceptable time. Meanwhile, multicast routing problem is often attributed to multicast routing tree problem, looking for minimal cost (i.e., Steiner Tree problem). As mentioned above, finding feasible paths with QoS multi-constraints is an NP-complete problem, to solve this question, firstly must reduce its complexity. Therefore, we could use a heuristic or approximate algorithm, the ant colony algorithm (*i.e.*, ACO) is suitable which has been widely used for multicast routing with its advantages of being robust, parallel, flexible, precise and without human interference. Many algorithms based on ant colony are fit for the features of ad hoc networks. Although ant colony algorithm has its advantages in solving QoS multi-constraints routing problem [1~3], it has a major drawback of slow convergence because of that it needs quite a number and generations of ants to find the best path.

Wireless ad hoc networks can exist and work well only if nodes behave cooperatively. Due to the open working environment, such network is susceptible to attack from selfish or malicious nodes, e.g., passive wiretapping, active intrusion, information blocks, counterfeiting, impersonate attack and other modes by using of wireless channels. Moreover, because of the resource limited and less CPU computing power in the embedded environment, the entity can not achieve a complex encryption algorithm, this network is more vulnerable to attacks. However, the research of multicast routing algorithms only focuses on minimal cost problem with QoS multi-constraints, the security of multicast routing is not correlatively studied. The confidential information can not be protected during transmission. With the multicast routing requirements showing diverse, the security factor should be correlatively considered. Traditional security mechanisms rely on either authenticated identities or clients getting credentials authorized to perform certain actions. Generally speaking, these mechanisms rely on well-established infrastructures (*e.g.,* Public Key Infrastructure (PKI)). However, the nature of wireless ad hoc networks is free of fixed infrastructures, so these mechanisms do not fit this network. In addition, the security mechanism basing on the traditional cryptosystem is used to resist external attacks, but it cannot effectively prevent malicious or compromised nodes from doing misbehaviors and solve the internal attacks by malicious nodes. Trust management is considered to be an effective measure to solve those questions, which is used to enhance the security and robustness of network. Similar to human society where one person trusts another to carry out an action, the concept of trust can be introduced to measure an expectation or uncertainty that an entity has about another's future behaviors. The trust-considered routing paths are not absolute security but certainly have an accurate measure of reliability.

Multicast trusted routing with QoS multi-constraints will be a core function of wireless ad hoc networks in the future. Therefore, it is an important and urgent research problem to set up multicast trusted routing quickly with guaranteed QoS multi-constraints. The main objectives in this paper include: (i) introduce the concept of 'trust' and construct a novel trust evaluation model; (ii) make 'trust' as one of QoS multi-constraints into searching multicast trusted routing basing on modified tree-based ant colony algorithm; (iii) guarantee quality for each QoS service for each routing path; (iv) improve the network security and the packet delivery ratio, protect the network from internal attacks. After weighing the strong and weak points of the existing trust models and multicast routing algorithms, we propose a novel ***m***ulticast ***t***rusted routing with QoS multi-constraints based on ***a***nt ***c***olony ***a***lgorithm, named as ***MTACA***. Simulations have been conducted to demonstrate that: (*a*) compared with conventional multicast routing (*i.e.,* MAODV [4]) and multicast routing based on modified path-based ant colony algorithm (i.e., MANSI [1]), our novel multicast routing algorithm not only has less 'cost', but also improves the

packet delivery ratio; (*b*) compared with 'MANSI', new algorithm has higher convergence speed; (*c*) compared with the same algorithm without considering the 'trust' factor (*i.e.*, 'MACA'), in solving the problem of multicast routing, it increases little more cost to exchange with the security of multicast routing paths. Therefore we conclude that our novel algorithm is suitable to the security of multicast routing with QoS multi-constraints in wireless ad hoc networks.

2. Related Work (Modify)

2.1 Multicast Routing Algorithms Based on Ant Colony Algorithm

In [1], Shen *et al.* apply this biologically inspired metaphor to the multicast routing problem in this network. They proposed multicast protocol ('MANSI') adapts a core-based approach which establishes multicast connectivity among members through a designated node (core). This exploration mechanism enables the protocol to discover new forwarding nodes that yield lower total forwarding costs, where cost is abstract and can be used to represent any metric to suit the application.

Yin *et al.* propose a niched ant colony optimization with colony guides (NACOg) algorithm to tackle the MinC/DB problem [2]. The NACOg algorithm first deliberates a constrained tree traversal (CU) strategy that guarantees the search to any feasible trees with respect to the QoS constraints. The proposed CTT strategy employs adaptive memory structure as contemplated in the tabu search and the strategy is more effective in constraint-handling than both the penalty-function and the produce-and-repair strategies which were broadly used in the literature.

A multi-objective evolutionary algorithm is proposed to solve the routing problem in wireless sensor network [3]. Two performance metrics, which consist of the maximization of remaining lifetime of the wireless sensor network and the minimization of transmission delay, are considered. Dominating relationship and similarity between solutions are used to compute the fitness of a solution in a population.

2.2 Trust models in MANET

Onolaja *et al.* [5] proposed a reliable and novel dynamic framework that utilized a data-driven approach for trust management. The framework used past interactions, recent and anticipated future trust values of every identity in the domain.

Based on the theories of fuzzy recognition with feedback, SCGM (1,1) model and Markov chain, Zhang *et al.* [6] present a pattern of prediction making.

Considering a requested (transmitter) node's historical trust value and its capability of providing services, we proposed a new method [7] to predict the node's trust value, based on the fuzzy logic rules prediction mechanism. The new method offers an accurate prediction of future behaviors, which obviously can reduce the probability of risk occurrence for next interaction.

A new approach is proposed in paper [8] for bootstrapping trust of Web services in which the interactions of a Web service with a user are observed during a certain time frame. The observations sequence is modeled as a hidden Markov model and matched against pre-defined trust patterns in order to assess the behavior of such Web service.

3. Trust Model

In trust management mechanism, trust can be defined as the subjective expectation that an entity has about another's future behavior based on the history of their encounters. Trust evaluations are always based on the evidence generated by the previous interactions among nodes within a protocol.

3.1 The Reputation-based Trust Model

A wireless ad hoc network is always comprised of many entities, and each entity is an independent node. We make the following assumptions in our model: (i) nodes always trust themselves; (ii) most of the nodes in the network are normal nodes (i.e., they work well and behave cooperatively); (iii) bad behaviors of normal nodes do not happen frequently; and (iv) only a small portion of the nodes in the network are malicious nodes.

In our model, *TV* denotes for a node's trust value, which is defined in a continuous range between 0 and 1 (i.e. $0 \leq TV_{ij} \leq 1$). Let v_i and v_j represent the evaluating and evaluated nodes, respectively. The trust value 0 signifies complete distrust, while the value 1 implies absolute trust.

Table 1. Trust Levels of Nodes

Level	*Trust Value*	*Meaning*
1	$[0, \eta)$	*Malicious node*
2	$[\eta, 0.7)$	*Less trustworthy node*
3	[0.7, 0.9)	*Trustworthy node*
4	[0.9, 1]	*Absolute trustworthy node*

The trust values can also be shared among neighbors using a higher layer, such as Reputation Exchange Protocol. Considering a file sharing system (*e.g.*, involved in Subsection 3.3), we define simple grading criteria for trust, and an example of node's trust levels is listed in Table 1. A threshold value η, termed as the black-list trust threshold, is used to detect malicious nodes. In other words, if the trust value of a node is smaller than η, it will be regarded as a malicious node by its evaluating node.

We set a monitoring interval Δt, an evaluating node begins to monitor its neighbor node when its neighbor node sends a packet to the next hop, to observe whether this neighbor node normally deals with the packet. After each interaction, node *j* checks whether the neighbor *k* forwards the packet correctly. Any evaluating node who receives reputation exchange packets (*REP*, Reputation Exchange Packet) sent by other nodes, it will firstly evaluate the trust value of the sending nodes. It will accept the updating information only if it believes in those sending node, and then it updates trust value and indirect observation (*IR*, Indirect Reputation, it obtains from exchanging the experience information with corresponding nodes) in its own trust table.

3.2 Trust Derivation

The node's trust value '*TV*' could be calculated according to the following formula [9]:

$$TV = w_1 * SR + w_2 * IR \quad (w_1+w_2=1 \text{ and } w_1 > w_2) \tag{1}$$

Where w_1 and w_2 separately denote for the weight values of *SR* (Subjective Reputation) and *IR* (Indirect Reputation), usually the value of w_1 is greater than w_2, which is used to prevent the enemy's malicious slander.

Each node in our model additionally owns a trust table with items defined as follows.

Table 2. Node v_1's Trust Table

Nb	T_{in}	SR_{in}	IR_{in}	T_{out}	SR_{out}	IR_{out}	ΔR^t	*Black-list*
v_2	0.90	0.92	0.85	0.92	0.95	0.85	0.1	*No*
v_3	0.79	0.88	0.58	0.23	0.4	0.17	0.1	*Yes*

In each row of the table, *Nb* denotes node v_1's neighbor that can communicate with v_1 via a single-hop. T_{in} is the trust value that the neighbor node gets about node v_1; SR_{in} is the subjective reputation that the neighbor node gets about node v_1; IR_{in} is the indirect reputation that the neighbor node gets about node v_1; T_{out} is the trust value that node v_1 has

about the neighbor; SR_{out} is the subjective reputation that node v_1 has about the neighbor; IR_{out} is the indirect reputation that node v_1 has about the neighbor; ΔR^t denotes for the deviation threshold; *Black-List* indicates whether v_1 considers this neighbor as a malicious node or not.

3.2.1 Calculation of Subjective Reputation (SR): The sender places itself in promiscuous mode after the transmission of any packet so as to overhear the retransmission by the forwarding node.

The normal phenomenon: in a time interval ΔT, the number of error packets is smaller than the threshold L, it will have a linear increment in *SR*. After each normal time interval, it adds a *changevalue1* to *SR*, according to the following formula:

$$SR = SR + ChangeValue1 \tag{2}$$

The abnormal phenomenon: in a time interval ΔT, the number of error packets is bigger than the threshold L. There will have an exponential decrement in *SR* accompanied with n.

$$SR = SR - ChangValue2 * 2^{n-1} \tag{3}$$

Where n represent for the penalty coefficient. For instance, if the error packets are 3 times bigger than L, then the value of n is set to 3. *Changevalue2* should be appropriately bigger than *Changevalue1*, if the number of error packets is smaller than L in an interval ΔT, then we will re-count the number of error packets in the next interval. The values of ΔT and L depend on the current network conditions and packet transmission capacity. The value of L depends on the actual environment situation, if the value is too big, the attacker node is possible to intentionally make L-1 error packets; if the value is too small, the penalty coefficient will be too large to make a reasonable attenuation in *SR*.

3.2.2 Calculation of Indirect Reputation (IR): The value of *IR* updates in two ways: triggered update and periodical update. ΔR^t denotes for the deviation threshold.

Triggered update: if the change of trust value exceeds a threshold ΔR (via *SR* or *IR* changes), the node will broadcast the change information within three hops for saving the energy. This method requires an field ΔR adding to the reputation table which is used to describe the changes of reputation (i.e., 'trust'), $\Delta R = \Sigma \Delta R_i$ (i.e., ΔR_i represents the change value of the ith time). If the value of ΔR exceeds a certain threshold ΔR^t, the node will broadcast a packet *REP*. After broadcasting this packet, the value of ΔR is set to be 0. Those nodes who receive the *REP* packet will update their own value of *IR* corresponding to the nodes appeared in the packet by the following formula:

$$IR_{ik} = IR_{ik} + \Delta R * TV_{ij} \tag{4}$$

i represents the evaluating node, j represents the third party node, k represents the evaluated node. TV_{ij} represents the trust value of node j evaluated by node i.

Periodical update: all nodes periodically broadcast the information which is included in their own trust list. A node updates the value of *IR* corresponding to other nodes using the following formula:

$$IR_{ik} = IR_{ik} + \frac{\sum_{j=1, j\neq i, j\neq k}^{n} (TV_{jk} - IR_{ik}) * TV_{ij}}{n} \tag{5}$$

n represents the number of received different packets.

3.3 Calculation of Path Trust

When a source node prepares to discover a routing path for transmitting message to any destination, it needs to assess the credibility of this path. Path trust value is computed according to the intermediate nodes' trust values on the path, which can be defined as a constraint in the trusted routing decision.

Considering the axiom that concatenation propagation of trust does not increase trust, route trust should not be more than the trust values of intermediate nodes. Capturing the notion of social networks, node v_i should give an objective estimate to all immediate nodes on the candidate routing path. So, at time t, the trust of a path P (denoted by $Path\_TV_P$) is equal to the continued product of node trust values in the route, using the following equation.

$$PathTV_P = \frac{\sum_{k=1}^{n} TV_{sk}}{n} = \frac{TV_{s1} + \sum_{k=2}^{n} [TV_{s1} \times TV_{12} \times ... \times TV_{(k-1)k}]}{n} \quad (6)$$

Where v_s is the source node, v_n is the destination node of path P.

In this paper, we divide different shared files into two levels: important documents I, and the regular documents R. The service nodes determine to share the level of files basing on the assessment of path trust. For the above two levels of shared files, combined with trust levels (i.e., Table 1), we define a map function F.

$$F(PathTV_P) = \begin{cases} I & 0.9 \le Path\_TV_P \le 1 \\ R & 0.7 \le Path\_TV_P \le 1 \end{cases} \quad (7)$$

4. Mathematical Notation for Multicast Trusted Routing with Multi-Constraints

A MANET is modeled as a directed, connected graph $G = (V, E)$, where V is a finite set of vertices (network nodes) and E is the set of edges (network links) representing connection of these vertices. Let $|V|$ be the number of network nodes and $|E|$ be the number of network links. The link $e = (u, v)$ from node $u \in V$ to node $v \in V$ implies the existence of a link $e' = (v, u)$ from node v to node u. Four non-negative real value functions are associated with each link e ($e \in E$): cost $C(e)$, delay $D(e)$, trust $TV(e)$ and available bandwidth $B(e)$. The link-cost function, $C(e)$ may be either monetary cost or any measure of resource utilization that must be optimized. The link delay, $D(e)$, is considered to be the sum of switching, queuing, transmission, and propagation delays. The link trust, $TV(e)$ is considered to be the trust value of node v evaluated by node u. The link bandwidth, $B(e)$, is the residual bandwidth functions. $D(e)$, $TV(e)$ and $B(e)$ define the criteria that must be constrained (bounded). Because of the asymmetric nature of communication networks, it is often the case that $C(e) \neq C(e')$, $TV(e) \neq TV(e')$, $D(e) \neq D(e')$, and $B(e) \neq B(e')$. In this paper, the function value of cost is used as criteria for evaluating the fitness of link.

A multicast tree $T(s,M)$ is a sub-graph of G spanning the source node $s \in V$ and the set of destination nodes M $M \subseteq V - \{s\}$. Let $m = |M|$ be the number of multicast destination nodes. We refer to M as the destination group and $\{\{s\} \cup M\}$ the multicast group. In addition, T may contain relay nodes which are not in the multicast group. Let $P_T(s,d)$ be a unique path in the tree T from the source node s to a destination node $d \in M$.

We now introduce the following parameters to characterize the quality of the tree.

The total cost of the tree $T(s,M)$ is defined as the sum of the cost of all links in that tree and can be given by:

$$C(T(s,M)) = \sum_{e \in T(s,M)} C(e) \quad (8)$$

The total delay of the path $P_T(s,d)$ is simply the sum of the delay of all links along $P_T(s,d)$:

$$D(P_T(s,d)) = \sum_{e \in P_T(s,d)} D(e) \quad (9)$$

The total delay of the tree $T(s,M)$ is defined as the maximum value of the delay on the paths from the source node to each destination node:

$$D(T(s,M)) = \max(D(P_T(s,d))), \forall d \in M \quad (10)$$

The total trust of the path $P_T(s,d)$ is defined as the minimum trust value of all links along $P_T(s,d)$:

$$Path\_TV(P_T(s,d)) = \min\{TV(e), e \in P_T(s,d)\} \quad (11)$$

The total trust of the tree $T(s,M)$ is defined as the minimum value of the trust on the paths from the source node to each destination node:

$$TV(T(s,M)) = \min(Path\_TV(P_T(s,d))), \forall d \in M \quad (12)$$

The bottleneck bandwidth of the path $P_T(s,d)$ is defined as the minimum available residual bandwidth at any link along the path:

$$B(P_T(s,d)) = \min\{B(e), e \in P_T(s,d)\} \quad (13)$$

The total bottleneck bandwidth of the tree $T(s,M)$ is defined as the minimum value of the bandwidth on the paths from the source node to each destination node:

$$B(T(s,M)) = \min(B(P_T(s,d))), \forall d \in M \quad (14)$$

The jitter of the tree is defined as the average difference of the delay on the paths from the source node to each destination node:

$$J(T(s,M)) = \sqrt{\sum_{d \in M} (D(P_T(s,d) - delay\_avg)^2)} \quad (15)$$

Where *delay_avg* refers to the average value of the delay on the paths from the source node to each destination node.

Let D_Z be the delay constraint, TV_Z be the trust constraint, B_Z be the bandwidth constraint, J_Z be the jitter constraint of the multicast tree. The multi-constrained least-cost multicast problem is defined as: $Min\{C(T(s,M))\}$

$$\begin{cases} D(T(s,M)) \le D_z \\ TV(T(s,M)) \ge TV_z \\ B(T(s,M)) \le B_z \\ J(T(s,M)) \le J_z \end{cases} \quad (16)$$

5. Algorithm Description

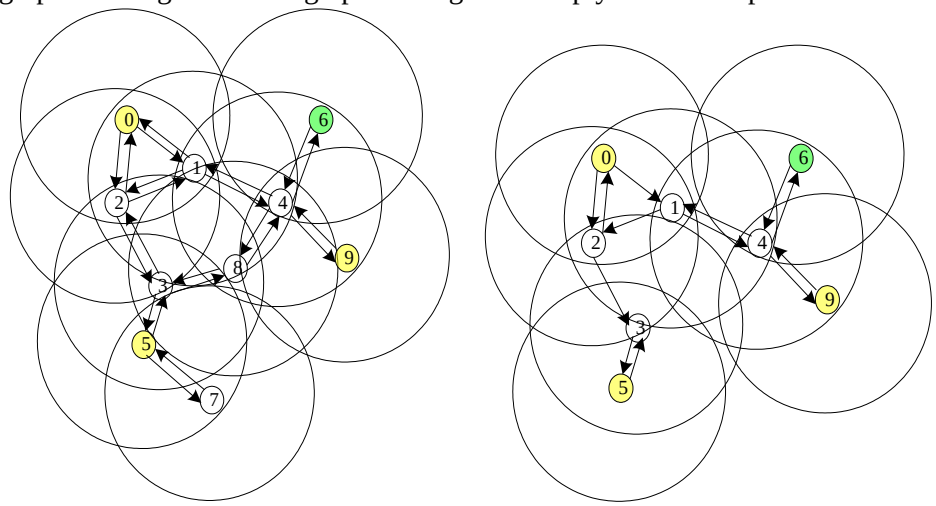
Ant colony algorithms have been applied to the solution of multicast routing, which mentioned ant colony algorithms for the solution of multicast routing inherit faithfully the basic features of the conventional ant colony algorithm in searching for paths. They can be said to be basically path-based ant colony algorithm. Reference [10] proposed a tree-based ant colony algorithm for multicast routing. But all the above methods do not consider the credibility of the paths in the multicast tree, the security of those paths is not guaranteed, thus we incorporate the concept of trust.

5.1 Description of New Algorithm

In this paper we modify the tree-based ant colony algorithm, introduce the trust into it and make 'trust' as another constraint of QoS parameters. We also make many improvements to this ant colony algorithm, and make it more efficient. The main idea of the modified algorithm is that: (*a*) perform pre-pruning operation to the graph *G*; (*b*) transfer the original graph to a sub-graph *G*', new algorithm executes in this sub-graph; (*c*) use optimized data structure to store corresponding data of *G*'.

There is not any only destination node in the algorithm. The tree that the ant has found includes all the destination nodes. There is no only current node for every ant. Every node on the tree that has been found is likely to be the current node. Every step in seeking a multicast tree made by each ant has no other meaning of any path than to enable the current tree to grow further. The only principle observed by the new algorithm is the positive feedback mechanism. The tree that the ant found includes all the destination nodes and can satisfy all multiple constraints.

5.1.1 The Required Data Storage Structure and Pre-pruning: Firstly, we use the depth first search algorithm to traverse the whole network topology from the source node. In the searching process, the leaf nodes which do not belong to the set of multicast group and their associated edges are removed from the graph *G*. And at the same time, the bandwidth of edges which do not meet the requirements bandwidth are also removed, then graph *G* changes to a sub-graph *G*'. Figure 1 simply shows this process.



(*a*) A MANET with 10 nodes (*b*) after pre-pruning

Figure 1. Pre-pruning for Topology

We add a new table data structure to each node (*i.e.*, Table 3) which used to record the calculated multi-constraints information.

Table 3. Node's Table Data Structure

Variable name	***Type***	***Effect***	***Initial***
delay	*int*	Record the path delay from the source node to this node	0
trust	*float*	Record the path trust from the source node to this node	0.0
bandwidth	*int*	Record the path bandwidth from the source node to this node	0

New algorithm uses the adjacent table data structure to store the information of the directed edges, including the values of cost, delay, bandwidth and trust. Because of the dynamics of trust, the pre-pruning process does not deal with this constraint.

If the network topology after the pre-pruning process shows as a non-connected graph, it indicates that there is no multicast tree met the multi-constraint QoS parameters, then the algorithm ends.

5.1.2 Seek a Multicast Tree with Multi-constraints

a) Tree Growth

Step1. Known the source node as *s*; the destination node set is $D=\{d_1,d_{2,\ldots,}\ d_{num}\}$; the counter *count*= 0.

Step2. Set up a null tree, add the source node *s* to *T*; the nodes set *Node*={*s*}; and set up a null link set *E*'.

Step3. Select a link from the set *E*' according to the probability formula and recorded as $e(v_m,v_n)$;

The probability computation formula of link $e(v_i,v_j)$ selection is:

$$P_{ij}=\begin{cases}\dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{e(v_i,v_j)\in E'}[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}} & e(v_i,v_j)\in E' \\ 0 & others\end{cases}$$

Where τ_{ij} is the intensity of pheromone on link $e(v_i,v_j)$; η_{ij} is the heuristic function of link $e(v_i,v_j)$, in the paper we take $\eta_{ij}=\frac{1}{c_{ij}}$, c_{ij} is the cost of link $e(v_i,v_j)$; α and β are mediating factors; E' is a link set, and $E'=\{e(s,v_i)\,|\,e(s,v_i)\in E\wedge trust(e(s,v_i))\geq T_d\}$, where T is the sub-tree found by the ant.

Step4. Add link $e(v_m,v_n)$ to tree T, add nodes v_n to set *Node*. If $v_n \in D$, then *count*++.

Step5. Remove the illegal links from E', $E'=E'-\{e(v_i,v_n)\,|\,e(v_i,v_n)\in E'\}$.

Step6. Add new legal candidate link to E', $E'=E'\cup\{e(v_n,v_j)\,|\,v_j\notin Node\wedge trust(e(v_n,v_j))\geq T_d\}$.

Step7. At this moment if the number of *count* < |*R*|, then go to Step 3.

B) Tree Pruning

With "*Tree growth*" process, the algorithm finds a tree that covers all nodes which in the multicast group members, however there could be some leaf nodes of non-multicast group. Prune the tree *T*, remove the leave nodes of all the non-group members and obtain a multicast tree *T*, return to tree *T*.

The pruning algorithm used in this paper is a recursive algorithm, the core idea of this algorithm is: First, get a tree from the input, prune each of sub-tree belonged to the root node. If the implementation of any sub-tree pruning algorithm returns value 1, then return 1; If the implementations of all sub-tree pruning algorithm return value 0, but the root node is the group member node, then return 1; Otherwise, remove the root, then return 0.

C) Pheromone Update

To find the optimal solution, the ants need to periodically update the concentration of pheromone on the network link, it is used to form a positive feedback mechanism. Update the pheromone according to the following formula.

$$\tau_{ij}=\rho\cdot\tau_{ij}+(1-\rho)\cdot\Delta\tau_{ij}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}$$

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{f_k} & \text{if k-th ant passes e}(v_i,v_j)\text{ in this cycle} \\ 0 & \text{otherwise}\end{cases}$$

Where $\tau_{ij}(t)$ is the intensity of the pheromone of link $e(v_i,v_j)$ at *t*-th moment, ρ is the evaporating speed of the pheromone, $0<\rho<1$, $\Delta\tau_{ij}$ is the augmenting amount of the pheromone on link $e(v_i,v_j)$, $\Delta\tau_{ij}^{k}$ is the amount of the pheromone left by the *k*-th ant on link $e(v_i,v_j)$, *Q* is a constant, T_k denotes the multicast tree found by the *k*-th ant, f_k denotes the total cost of tree that the *k*-th ant has found; *iter*++.

5.1.3 Construct Set of Trees

Step1. The counter of iterative number *iter*=0; the optimal tree $T_{best}=\varnothing$.

Step2. The tree set $SET_{tree}=\varnothing$.

Step3. Transfer process to 5.2.2, and seek a multicast tree *T*.

Step4. Compute the end-to-end delay, trust and bandwidth of T. If these constraints are satisfied, it adds T to the set SET_{tree}; otherwise, go to Step 3.

Step5. Update the optimal value. If the composite function value of T, if $f(T)<f(T_{best})$, then $T_{best}=T$.

Step6. If $|SET_{tree}|<MaxTreeNum$, go to Step 3.

Step7. If the tree founded after many times does not have great changes any longer, the algorithm has already converged, or the number of cycling times has outnumbered the maximal times, i.e. $iter>MaxIterNum$, then output the best solution, end the algorithm. Otherwise go to Step 2.

For all the trees in SET_{tree}, compute their assessment function values $C(T(s,M))$. Then we will obtain the optimal tree with the minimum 'cost' value.

5.2 Algorithm Analysis

5.2.1 Advantages and Disadvantages of Algorithm: Advantages: (*a*) our new algorithm uses only one variable to record the concentration of pheromone on each edge; (*b*) it dose pre-work to a directed, connected graph; (*c*) it uses the efficient data structure to store the information of the directed edges; (*d*) it neither does the process to find paths, nor uses paths to merge a tree, the efficiency of our algorithm compared with original ACO algorithm (e.g., MANSI) is greatly improved; (*e*) the route maintenance is simple. It could be used to solve multicast trusted routing problem with QoS multi-constraints. With the help of trust, we could easily construct a security multicast tree in a wireless ad hoc networks.

Disadvantages: the new algorithm, however, is aimed at single source multicast trusted routing. Therefore it is not suitable for other scheme problem. Moreover, this algorithm uses global information, so it is not easy to realize distributed computation and not suitable for fast-moving networks.

5.2.2 Analysis of Space and Time Complexity for MTACA: In a v-nodes network, each node mostly has links with (v-1) nodes. In our research, the edges are directional, so the graph mostly has $v*(v-1)$ directed edges. We use the adjacent table data structure to store the information for edges, and all ants share this adjacent table which contains four categories of information. Another form of data storage, node's table, contains three categories of information. Therefore, the data space for storage required is about $O(4v(v-1)+3v)\approx O(v^2)$.

There are *antnum* ants to find trees simultaneously, the algorithm ends when it meets the convergence condition or after several iterations. So it's total time complexity is about $O(iter\times antnum|E||V|)=O(iter\times antnum\times v(v-1)\times v)$.

6. Simulation Results and Analysis

The proposed algorithm is implemented with Visual C 6.0 and runs on an Intel Pentium 4, 3.00 GHz machine with 2 GB memory, running Windows XP. The network topology used in the experiment is randomly generated according to the approach of Waxman model, $P(u,v)=\beta\cdot\exp\frac{-l(u.v)}{\alpha\cdot L}$ where P (u, v) represents the probability existence of the links between node u and v, l(u, v) indicates the Euler Distance between node u and v, and L is the maximum distance between any two nodes in the network. α and β are two real parameters between 0 and 1, α indicates the ratio between the long distance links and the short distance links (a smaller α value means more short links); β controls the average degree of network (a greater β value means lager average degree). The network topology with real world network features is generated through selecting the value of α and β in Waxman model.

Simulation is made to compare our new multicast trusted algorithm (MTACA) with MAODV [4], MANSI [1] and MACA (*i.e.*, our new algorithm without trust constraint). The size of topology ranges from 20 to 200, the source node (single source node in next experiments), destination nodes and link weight (*i.e.*, 'cost') of the multicast are all generated randomly, and the total number of destination nodes is 15% of the total number of topology. This paper assumes α is 0.07 and β is 2.8. The ant colony maximal iteration number *iter* is half of $|V|$, and the number of ants at initialization is 25. Other parameters are set in Table 4 corresponding with Section 3.

Table 4. Parameters Setting

ΔR^t	ΔT	L	*ChangeValue1*	*ChangeValue2*
0.1	3s	20	0.03	0.05

To decrease the disturbance of random error, every experiment repeats 30 times and the average experiment results are computed.

6.1 Influencing Experiment for Pre-pruning

Our purpose of pre-pruning is to remove those leaf nodes which do not belong to the set of multicast group, their associated edges and the bandwidth of edges which do not meet the requirements bandwidth.

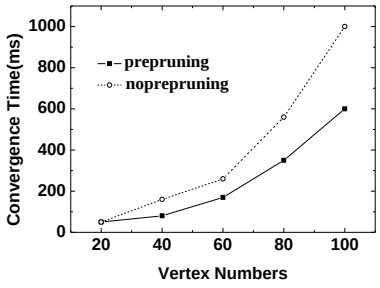


Figure 2. The Comparison Results of Algorithm Convergence Time

We can see that after pre-pruning operation for the network topology where the time complexity is $O(2|E|)$, the new algorithm not only significantly reduces the searching time for useless edges, speeds up the convergence time, but also improves the possibility to obtain an optimal solution.

Figure 2 shows the comparison result of algorithm convergence time between the algorithm runs with the pre-pruning process and the modified algorithm runs without pre-pruning process.

6.2 Wireless Ad Hoc Network with 10% Malicious Nodes

Different algorithms were proposed inclined to different QoS constraints. To make the comparisons be rationality, we relax their own QoS constraints (i.e., when these algorithms are running, there are no constraints except considering their minimal costs, however, MTACA adds an additional constraint-trust).

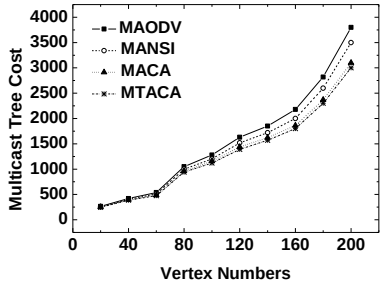


Figure 3. The Cost Comparison on Different Scales of Network Topology

Figure 3 illustrates the cost comparison of the algorithms on different scales of network topology. It can be easily seen from the figure that, with the growth of network topology, the cost of multicast tree obtained through different algorithms increase as well. As shown in Figure 3, the cost of our new algorithm is always smaller than that of the other two algorithms. With the growth of topology, the performance is more obvious. This is due to that, our algorithm based on the principle of modified tree-based ant colony algorithm, which will choose an approximate optimal multicast tree until converging. Compared to the traditional path-based (i.e., ants are sent out to find paths from a source node to every destination node, and then certain algorithm is used to integrate every path into a tree through removing loops, pruning, substituting and other operations, or the two processes of searching a path from a source node to a single destination node and integrating a tree are conducted meanwhile, that is, adding a path to the tree after finding it.) ant colony algorithms 'MANSI', MACA has a greater chance to select a better link to reduce the global cost of finding tree. Therefore, it is possible to obtain better results. MAODV does not consider the cost of multicast route, when searching for a multicast shared tree. The cost of MTACA is slightly bigger than that of MACA, because of that, MTACA introduces the concept of trust into finding routes, which balances the tradeoff between security and cost. When searching for a multicast tree, the ants will avoid some un-trusted nodes (i.e., malicious or selfish nodes), they will also avoid some links associated with those nodes, despite relatively lower cost of these links, and finally select other links whose cost maybe higher, not the optimal ones. MTACA and MACA belong to the same algorithm, there is less difference in their convergence time, so we only compare the convergence time between MAODV, MANSI and MACA.

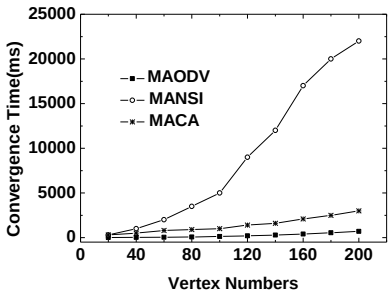


Figure 4. The Convergence Time Comparison on Different Scales of Network Topology

Figure 4 illustrates the convergence time comparison of the algorithms on different scales of network topology. It can be seen from the figure, with the growth of network topology, the convergence time of all algorithms increase as well. As shown in Figure 4, the convergence time of MAODV is the shortest one. MAODV is based on the shortest

path (i.e., smallest-hop), which generally only does one calculation and can be done in polynomial time. MACA and MANSI belong to the heuristic algorithm, their convergence time are calculated until their results converging. But it takes shorter time for MACA to converge than MANSI. This is more obvious with the increase of the topology scale. There are three reasons for this: Firstly, what MACA has found is a tree while MANSI has initially found is paths, which MANSI must combine into a tree. Secondly, the time MACA spends in finding a tree is not longer that the time MANSI searches paths and combines them into a tree. Thirdly, because of the updating of pheromones of a whole tree when MANSI searches a path from source to destination node, the ant may be misled by the pheromone leading to another path, thus the convergence speed of the algorithm will be delayed. In contrast, the ant of MACA cannot be misguided.

6.3 Varying Number of Malicious Nodes

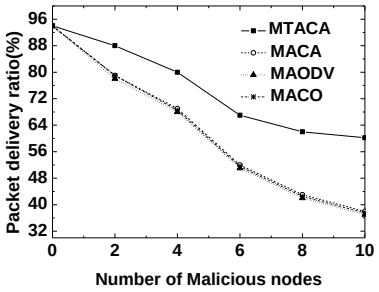


Figure 5. The Delivery Ratio with Different Number of Malicious Nodes

In this experiment, there are 30 nodes, and we evaluate these algorithms by varying the number of malicious nodes.

As shown in Figure 5, when there are no malicious nodes, the packet loss rate is about 6% in MAODV, MANSI, MACA and MTACA, and the delivery ratios in all algorithms degrade sharply as the number of malicious nodes increases. The delivery ratio of MTACA drops from 94% to 64% as the number of malicious nodes varies from 0 to 10. Lower packet delivery ratio means less network throughput. Malicious nodes essentially limit interactions between nodes in the network. However, the delivery ratio of MTACA is always higher than that of MAODV, MANSI and MACA. Because of that, the MTACA introduces the concept of trust into finding route, those nodes in MTACA, when detecting attacks (*e.g.,* gray hole or black hole attack), which can try another trusted route to forward packets and thus packet delivery ratio is improved. MAODV, MANSI and MACA almost have the same delivery ratios.

7. Conclusions and Future Works

This article studies the questions related with the definition and synthesis of node's trust and multicast routing with QoS multi-constraints for wireless ad hoc networks. Due to the inherent characteristics of this network, a reputation-based trust evaluation model is proposed, which consequently enhances network's security and performance. Combined with the trust model and modified tree-based ant colony algorithm, a novel multicast trusted routing with Qos multi-constraints (*MTACA*) is proposed to discover trustworthy multicast forward paths and alleviate the attacks from malicious nodes. This protocol provides a flexible and feasible approach to choose a security multicast tree in all candidates, which balances the tradeoff between security and cost. The simulation results analyze the effectiveness of our protocol.

We would continue our work in the following three directions: 1. Make a further improvement for the trust model proposed in this paper, we plan to incorporate other

decision factors to our trust model; 2. We will consider an adaptive trust level classification of nodes taking into account the average trust value of all nodes. The problem of dynamic behavior modification will also be considered; 3. Make the proposed routing protocol be adapted to network changes.

Acknowledgments

We would like to thank anonymous referees for their helpful suggestions to improve this paper.

This research is sponsored by the Natural Science Foundation of China (NSFC) under Grant Nos. 61402245 and 61272425, the Project funded by China Postdoctoral Science Foundation under Grand No. 2014M551870, the Shandong Provincial Natural Science Foundation No. ZR2014FQ010, the Project funded by Qingdao Postdoctoral Science Foundation, the Open Project Foundation of Shandong Provincial Key Laboratory of Software Engineering under Grant No. 2013SE01 and Foundation of Huawei under Grant No.YB2013120027.

References

[1] C. Shen and C. Jaikaeo, "Ad hoc multicast routing algorithm with swarm intelligence", Mobile Networks and Applications, vol. 10, nos. 1-2, (**2005**), pp. 47-59.
[2] P.-Y. Yin, R.-I. Chang and C.-C. Chao, "Niched ant colony optimization with colony guides for QoS multicast routing", Journal of network and computer applications, vol. 40, (**2014**), pp. 61-72.
[3] S. Su, H. Yum and Z. Wu, "An efficient multi-objective evolutionary algorithm for energy-aware QoS routing in wireless sensor network", International journal of sensor networks, vol. 13, no. 4, (**2013**), pp. 208-218.
[4] E. Cheng, "On-demand multicast routing in mobile ad hoc networks", M. Eng. thesis, Carleton University, Department of Systems and Computer Engineering, (**2001)**.
[5] O. Onolaja, R. Bahsoon and G. Theodoropoulos, "Trust dynamics: a data-driven simulation approach", Trust Management V, vol. 358, (**2011**), pp. 323-334.
[6] F. Zhang, Z. Jia, H. Xia and E. Sha, "Node trust evaluation in mobile ad hoc networks based on multi-dimensional fuzzy and Markov SCGM (1,1) model", Computer Communications, vol. 35, no. 5, ((**2012**), pp. 589-596.
[7] H. Xia, Z. P. Jia, X. Li, L. Ju and E. Sha, "Trust prediction and trust-based source routing in mobile ad hoc networks", Ad Hoc Networks, vol. 11, no. 7, (**2013**), pp. 2096-2114.
[8] H. Yahyaoui and S. Zhioua, "Bootstrapping trust of web services based on trust patterns and hidden Markov models", Knowledge and Information Systems, vol. 37, no. 2, (**2013**), pp. 389-416.
[9] H. Xia, Z. Jia, L. Ju and Y. Zhu, "Multicast Trusted Routing with QoS Multi-Constraints in Wireless Ad Hoc Networks", Proc. of the Second IEEE International Symposium on Advanced Topics on Embedded Systems and Applications (IEEE ESA 2011), (**2011**), pp. 1277-1282.
[10] H. Wang and Z. Shi, "The tree-based ant colony algorithm for multi-constraint multicast routing", Proc. of the International Conference on Advanced Communication Technology (ICACT 2007), (**2007**), pp. 1544–1547.

Authors

Hui Xia, he was born in 1986, he is currently a Lecture in the College of Information Engineering at Qingdao University, China. His research interests focus on network and information security, trust computing, mobile computing, embedded system and cryptology. (*qduxiahui@gmail.com*)

Xiu-qing Lu, she was born in 1975, she is a Lecture in the College of Information Engineering at Qingdao University, China. Her research interest focus on wireless sensor networks, embedded system and wireless mobile communications. (*xqlu@qdu.edu.cn*)

Zhen-kuan Pan, he was born in 1966, is a Professor and Ph.D. supervisor in the College of Information Engineering at Qingdao University, China. His main research interests include virtual reality technology, computer vision, image science, network and information security, embedded system and trust computing. (*zkpan@qdu.edu.cn*)