A Three-Tier Framework Supporting Soft QoS Routing in MANET[1]

Jin Xin, Wang Hongbo, Zhang Yaoxue, and Meng Bin

Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, China
jx01@mails.tsinghua.edu.cn

Abstract. Mobile Ad hoc Networks (MANET) is a collection of randomly moving wireless devices within a particular area. Unlike in cellular networks, there are no fixed base-stations to support routing and mobility management. Further more, many resources such as power energy and bandwidth are very limited in MANET. Concentrating on resolving these problems, we present a three-tier framework. The framework contains three cooperative algorithms, SSCA, DSRU and SQAR. SSCA is mainly responsible for topology management, DSRU is responsible for updating the routing information and the responsibility of SQAR is to select path which satisfies the QoS requirement. Experiment on the GloMoSim simulator shows that the framework proposed in this paper results in a notable reduction on energy consumption, routing overhead, packet collision times and rerouting times, and a notable improvement on network throughput and link stability, especially for the networks composed of high-speed mobile hosts.

1   Introduction

MANET is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. Numerous challenges [1] must be overcome to realize the practical benefits of MANET, because the network is highly dynamic and transmissions are susceptible to fading, interference, or collision from hidden/exposed stations.

In this paper, we concentrate on solving the problem of efficient routing caused by nodes moving with a relatively high velocity. For these fast-moving nodes, their location updates become obsolete by the time they reach the correspondent nodes. So, to get the exact position information of a mobile node needs a large routing overhead. Our overall goal is to build a system that can carry through efficient routing in such a dynamic environment, at the same time economize power consumption and maximize throughput of the network. Our overall solution is a cooperative three-tier framework. As shown in Fig.1, The framework consists of three algorithms, SSCA [2], DSRU [3] and SQAR. They have respective functions. The function of SSCA is topology management. It consists of three parts: mobility prediction, power control and

clustering. DSRU is based on SSCA and it is a routing update algorithm that combines the proactive policy and the reactive policy. SQAR is on the top level of the Framework, it uses the information got from DSRU to find a path that satisfies the QoS requirement of the two communication nodes. We will introduce them separately in Section 4.
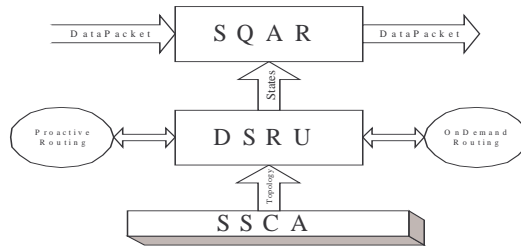

Fig. 1. System Framework

The organization of the rest of this paper is as follows. Section 2 summarizes the related works. In Section 3, we describe our framework in detail. Performance evaluation is done in Section 4. Section 5 concludes this paper and previews the future work.

2   Related Works

2.1   Mobility Prediction

As all known, mobility is the source of all difficulties. It causes frequent topological changes, makes the task of finding and maintaining routes in mobile ad hoc networks being non-trivial, and let fixed power provision be impossible. Many researchers have presented their mobility models and proposals [4, 5] to deal with this problem. These mobility models focus on the individual behavior in successive epochs, which are the smallest periods in a simulation, in which mobile hosts move in a constant direction at a constant speed. Recently, mobility prediction based on these mobility models has been reported as an effective means to decrease call-dropping probability and to shorten handover latency. J. chan et al in [6] have compared many kinds of mobility predication schemes, and concluded that the Direction Criterion has the best performance and that a high level of statistical randomness in users' movements may cause low prediction accuracy.

2.2   Power Control Scheme

Power energy is a very scarce and expensive resource for mobile hosts, and configured power transmit range of a host influences the total wireless network throughput. A recent paper [7], based on a simple interference model, derives a very interesting

result. If there are N nodes in a bounded region attempting arbitrary point-to-point communication, the throughput per node decreases at 1/N. Obviously, it indicates that the congestion and collision control becomes more critical in larger scale cluster. The selection of optimal transmit range to maximize throughput is studied in [8, 9]. However, they do not describe any techniques for actually controlling the power, nor do they concern themselves with connectivity. Other proposals [10] aim at balancing the power consumption to prolong the life span of network, but they don't consider how to save power energy.

2.3   Clustering Scheme

Though mobility prediction and power control are important in ad hoc networks as discussed above, single scheme is not sufficient to efficient routing in ad hoc network. It is necessary to take these factors into consideration to get an integrated solution. Clustering scheme, which is easy to implement adjustment to control routing overhead and to provide stable topology, plays a crucial role in ad hoc wireless networks for efficient routing. Many clustering schemes proposed in [11, 12] are 1-hop clustering algorithms, in which every node can be reached with at most 2 hops from any other nodes in the same cluster, but there is no clusterhead. Proposal proposed in [13] tends to reelect existing clusterheads as cluster governors even when the network configuration changes.

3   A Three-tier Framework

This section presents our framework in detail. As shown in Fig.1, the framework consists of two algorithms, SSCA, DSRU and SQAR. SSCA is a GPS based clustering mechanism. Its function is topology management, and it implements its function through three steps: mobility prediction, power control and clustering. By predicting the next location of mobile node with its historic trajectory, it adjusts the node's transmit power in advance, and controls all nodes in suitable size clusters. Based on the clusters, DSRU is proposed in order to control the routing overhead while gets relatively exact global topology information. DSRU is a hybrid routing algorithm. Its essential idea is to find a balance between optimal path and routing overhead. At last, we propose the SQAR. It is responsible for selecting and maintaining the paths that can satisfy the QoS requirement of the nodes. In the following sections, we will introduce them separately.

3.1   Suitable Size Clustering Algorithm (SSCA)

3.1.1   Mobility Prediction.
By interacting with Global Position System (GPS), any host can get its location (x,y,z). In a very short period, because of the inertia effect, we can assume that the force acting on the host moving with high speed is constant, and this force can be decomposed in

three dimensions, so we can also assume that the velocity variance is constant in three directions separately.

As all know the principle motion law:

$$s = v*t + \tfrac{1}{2}a*t^2 = \bar{v}*t \tag{1}$$

and

$$V = v + a * t \tag{2}$$

Here, S is the displacement in the period t, v is the initial velocity and α is the acceleration with same direction of v. we employ V denoting the final velocity after period t.
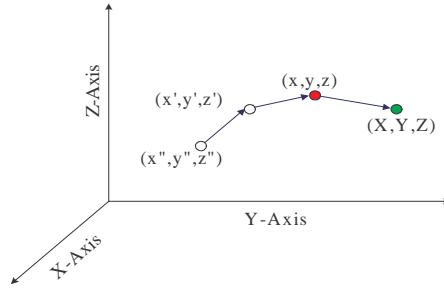

Fig. 2. Illustrating node motion trajectory, where (x'',y'',z'') and (x',y',z') are the history location. Current node locates $L(x,y,z)$. we predict it will be (X,Y,Z) in the next hits.

Fig.2 shows the trajectory of a mobile node. Now we employ v_x', v_x to denote the node average motion velocity in the segment from (x'',y'',z'') to (x',y',z') and the segment from (x',y',z') to (x,y,z) in the X-axis(same mean as v_y, v_z), and T to denote location sampling cycle, then we can get (3) from (1),

$$
\begin{aligned}
v_x' &= \frac{x'-x''}{T}, v_x = \frac{x-x'}{T} \\
v_y' &= \frac{y'-y''}{T}, v_y = \frac{y-y'}{T} \\
v_z' &= \frac{z'-z''}{T}, v_z = \frac{z-z'}{T}
\end{aligned}
\tag{3}
$$

(4) From (2)

$$
\begin{aligned}
v_x &= a_x*T + v_x' \\
v_y &= a_y*T + v_y' \\
v_z &= a_z*T + v_z'
\end{aligned}
\tag{4}
$$

Additional, in very short slice T, we assume that the acceleration is the same as the last slice, so the next most probable location can be predicted as,

$$
\begin{aligned}
X &= x + (v_x + a_x*T)*T \\
Y &= y + (v_y + a_y*T)*T \\
Z &= z + (v_z + a_z*T)*T
\end{aligned}
\tag{5}
$$

Replace v, a, and T in (5) with (3), (4), we get a simpler expression (6).

$$X = 3*x - 3*x' + x''$$
$$Y = 3*y - 3*y' + y''$$
$$Z = 3*z - 3*z' + z''$$

(6)

3.1.2   Power Control Scheme

Power control is a necessity in multi-hop networks, both to save power and to maximize the network throughput. In this context, we present an efficient power control scheme. As mentioned in Section 4.1.1, each node deploys a geologic method to find its physical location. Now, we derive the formula employed in SSCA to adjust the power. It is based on a well-known generic model for propagation [14] by which the propagation loss function varies as some ε power of distance. The value of ε is usually between 2 and 5, depending on the environment, specifically, if R is the loss in dB, then

$$R(d)= R (dthr), \text{ if } d<dthr$$

(7)

$$R (d)= R (dthr)+10*\varepsilon*log10(d/dthr) \text{ if } d\geq dthr$$

(8)

where d is the distance, and dthr is a threshold of distance below which the propagation loss is a constant; all logarithms in the remainder of this section are based on 10.

Let sc, pc, respectively, denote the current cluster size and current clusterhead transmit power. We need an expression for new transmit power p_d, so that the cluster has the desired size s_d.

Let cd_i^j, ed_i^j, respectively, denote the current distance and expected next distance from i to its neighbor j, and p_e denotes adjustment targeted power, $[d_{min}, d_{max}]$ is the range of adjustable power transmit distance.

$$cd_i^j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}$$
$$ed_i^j = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2}$$

(9)

$$cd = \max\{cd_i^j\}, j\in N_i$$

(10)

$$ed = \min\{d_{max}, \max\{ed_i^j\}\}, j\in N_i$$

(11)

As noted previously, S is the receiving sensitivity for all nodes. Then as $cd > d_{thr}$ & $ed > d_{thr}$, the following hold:

$$p_c - (\Re(d_{thr}) + 10*\varepsilon*\log(\frac{cd}{d_{thr}})) = S$$

(12)

$$p_d - (\Re(d_{thr}) + 10*\varepsilon*\log(\frac{ed}{d_{thr}})) = S$$

(13)

From (12) and (13), we get a simpler equation

$$p_d = p_c - 10 * \varepsilon * \log(\frac{ed}{cd}) \tag{14}$$

In our system, we employed $\varepsilon = 4$, but ε can also be configured depending upon the environment. Equation (14) can thus be used to calculate the new power periodically. We note that the formula applies for both power increasing and decreasing to bring the cluster size close to s_d.

3.1.3   Clustering Scheme.

In this section, we present a suitable size clustering scheme. Its purpose is to override the high threshold bounds and to adjust the power if the topology change is indicated by the routing update results in undesirable connectivity. It is triggered whenever an event driven or periodic link-state update arrives and it is incremental, in which it calculates new transmit power not from scratch, but being based on the currently used values.

Initially, all nodes start with the maximum possible power. With 1-CONID [15] clustering algorithm, it results in a maximally connected network, which enables successful propagation of updates and the initialization of a network topology database at each node. After this initialization, clusterheads conduct power control to maintain proper size of cluster around the configure value s_d by adjusting its pilot signal level, as follow:

$$s_c < s_d :$$
$$ed = \max\{ed + \text{rand}(\Lambda), d_{max}\}$$
$$\text{recalculate } p_e \text{ with (14)}$$
$$s_c > s_d :$$
$$ed = ed_i^{S_d}, \text{where} ed_i^1 < ed_i^2 < \cdots < ed_i^3 < \cdots < ed_i^{S_c}$$
$$ed = \min\{ed, d_{min}\}$$
$$\text{recalculate } p_e \text{ with(16)}$$
$$\text{adjust transmit power to} p_e$$

Λ is the system configurable value that is related to the power adjustment capability.

If a cluster has too many nodes including ordinary nodes (mobile stations) and gateways, the clusterhead reduces its power signal to make the area of the cluster shrink. If a cluster is suffering from isolation or has too little connectivity, its clusterhead increases power signal. Since both parties (clusterhead and mobile station) can control transmit power, a power signal should embed its transmit power level. Otherwise, the open loop power control would be impossible because the open loop control assumes a predefined power level of pilot signals.

3.2 Dynamic Self-adaptive Routing Update Algorithm (DSRU)

Based on the established cluster, we propose a hybrid routing algorithm. It is the combination of proactive policy and reactive policy. Intra-cluster routing uses a proactive policy, whereas the inter-cluster routing is reactive. In networks with low rates of mobility, clustering provides an infrastructure, which is more proactive. This enables more optimal routing by increasing the distribution of topology information when the rate of change is low. When mobility rates become higher, cluster size will diminish and reactive routing will dominate. The hybrid policy accompanies a better balance between routing overhead and quick routing.

The routing update includes two procedures, intra-cluster routing update and inter-cluster routing update. Intra-cluster routing update cycle is shorter than that of inter-cluster routing cycle.

The intra-cluster routing update is implemented by clusterhead. The clusterhead sends intra-cluster route status packet to its cluster members periodically, and the members will update their routing table after they receive the packet.

The inter-cluster routing update can employ any proposed proactive routing schemes. The clusterhead designates some gateway nodes as inter-cluster routing updaters. These updaters execute inter-cluster routing update procedure as follow:

1. The clusterhead initiates inter-cluster route status packet and sends it to its gateway nodes periodically.
2. Any gateway node receiving a inter-cluster update packet performs below actions:

   a) Integrates the routing status information of this packet to its local routing table and records the updating path of the source cluster.

   b) Refreshes timers of routing table items according to the new arrival inter-cluster update packet.

   c) Checks the travel path of this packet with that of last update. If successive update packets initiating from the same cluster have traveled on the same cluster path, the updater forwards new update packet to its direct neighbors except for the coming cluster, otherwise no forwarding is performed.
3. The timeout route items are removed from local routing table when its timer event arrives.
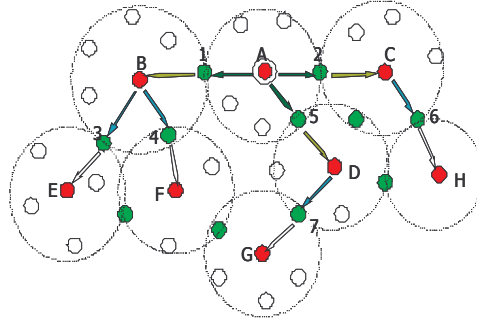
Fig. 3. Example of inter-cluster routing update

Fig.3 illustrates the process of above procedure. Node A initiates an inter-cluster route update packet and sends to node B transferred by node 1 (similar to node C and D), if node B has received inter-cluster update packet of the cluster delegated by node A in the successive inter-cluster update periods, node B forwards this update packet to node E and F, so node E and other nodes which are in the same cluster with E know the topology and link status of the cluster delegated by A. At this moment, local routing table of node E at least includes the status of three clusters, which delegated by E, B and F respectively.

3.3   Soft QoS Assurance Routing (SQAR) Algorithm

In this section, we have proposed a Soft QoS Assure Routing Algorithm. SQAR is based on SSCA and DSRU, its function is to select the path which is satisfied the QoS request of the application, guarantee the validity of the path, and balance the load.

3.3.1   Route request procedure.

When a source node wants to communicate with a destination node and has no routing information about this destination, it initiates a route-request procedure to find a route to the destination by broadcasting a route-request (RREQ) message to its neighbors as shown in Fig.4 and sets a route discovery expiration timer. The purpose of this timer is to detect whether the destination is reachable or not. The route discovery expiration time depends on the size of the network. The RREQ message has contained the following variant: bcastId, destAddr, destSeq, srcAddr, srcSeq, lastAddr, hopCount and bandwidth. The bcastId is incremented whenever the source node issues a new RREQ message. The intermediate node uses the pair (srcAddr, bcastId) to identify a RREQ message. When duplicate copies of RREQ arrive and their hop counts are greater than the hop count value recorded in routing table, they are discarded. The srcSeq number is used to maintain freshness information about the reverse route to source and destSeq number specifies the most recent routing information of the route to destination maintained in source node. When a node receives a RREQ message, it performs one of the following steps:

1. If the receiving node knows a route to the destination node, it checks to see if the route is current by comparing the destSeq in its own route entry to the destSeq in RREQ. If the destSeq in RREQ is not greater than that recorded in its own route entry and there is enough bandwidth in the path to satisfy the requirement, the intermediate node sends back a route reply (RREP) message.
2. If the receiving node does not know a route to the destination node or the destSeq in RREQ is greater than that recorded in its own route entry, it decreases the hopCount in RREQ by one. If the hopCount is zero, or the bandwidth of the node can't satisfy the QoS requirement in RREQ, or the role of the node is normal then the node will discard the RREQ. Otherwise, the receiving node attempts to build reverse links to the nodes that sent the RREQ message and then re-broadcast the RREQ message to their neighbors.

Each intermediate node repeats above procedure until an intermediate node finds a route to the destination, or the destination is reached. When an intermediate node knows a route to the destination, or the destination node sends RREP message back along the reverse link, the route request procedure is terminated.

For each intermediate node, after it has relayed a RREQ message, it begins to time. If it has not received a RREP message after ($2*(Delay_{End-to-end} - Delay_{Current\,Request})$), then it will re-relay the RREQ message. After several times, it will send a route-error (RERR) message to upstream node. Each intermediate node will do the same procedure until an intermediate node finds a route to node D, or node D is reached

3.3.2 Route reply procedure.

After the route-request procedure, the RREQ message will arrive to the destination or a node that possesses a route to the desired destination. Then the receiving node will send an RREP message to the source along the reverse links. The RREP message has contain the following variant: bcastId; destAddr; destSeq; srcAddr; srcSeq; lastAddr; hopCount; bandwidth; avaBandwidth. Note that the destSeq is extracted from the RREQ message and the intermediate nodes use the pair (destSeq, destAddr) to identify a RREP message. The avaBandwidth records the max available bandwidth of the path. As the RREP message travels to the source, each node along the reverse path will perform one of the following operations.

1. If the RREP message is not a duplicate message and the receiving node is the source, then it will create a forward link, update its routing table and start communication.
2. If the RREP message is a duplicate message from another neighbor, the receiving node will set up a forward link, update its backup table, and then discard the RREP message; otherwise, it will discard the message.
3. If neither 1 nor 2 described above is true, the receiving nodes will create a forward link, update its routing table and send an RREP message back along the reverse link.

3.3.3 Route maintenance procedure.

Once a next hop becomes unreachable, upstream nodes must perform appropriate operations to recover the routing path. In SQAR routing protocol, intermediate nodes are responsible for finding new routes when the next hops become unreachable. This

can be done by maintaining multiple next-hops in each mobile host. When link failures occur during communication, upstream nodes detect the failures and eliminate invalid routes. If these upstream nodes have more than one next hop in their routing tables, they select new one, otherwise they inform their upstream nodes along the reverse links. These upstream nodes then become responsible for reconstructing new routes. Thus, the number of new route reconstructions is reduced.

4 Simulation and Performance Evaluation

4.1 Simulation environment

We have implemented our algorithms within the GloMoSim [16] library. The GloMoSim library is a scalable simulation environment for wireless network system using the parallel discrete-event simulation language called PARSEC [17]. Our simulation models a network within 1000*1000 meter square and the nodes in the network are placed uniformly. Radio propagation range for each node is 150 meters and channel capacity is 2 Mbits/sec. In most of experiments unless specified, the network consists of 100 nodes and the average moving speed varies from 5m/s to 45m/s. Each simulation executed for 10 minutes of simulation time. We run each scenario three times and the data collected are averaged over those runs.

To validate the effectiveness of mobility prediction, we compare the performances of our clustering algorithm in two cases. One is calculating the next transmit power range only based on the prediction distance (Abbr. as **olnp**), and the other based on the maximum of current distance and prediction distance (Abbr. as **nowp**). In order to test the advantage of power control, we have simulated the version of no power control (**noAdj**), which is 1-CONID. In addition, we also simulate **FSR** [18] algorithm for performance comparison.

4.2 Simulation results

We first compare the packet collision, transmit power radio among the **olnp**, **nowp**, **noAdj** and **FSR**, then we compare the routing overhead between **FSR** and **DSRU**. We will show how node density and moving speeds impact the network performance.
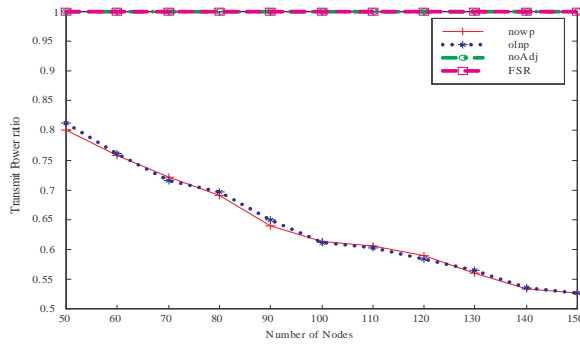
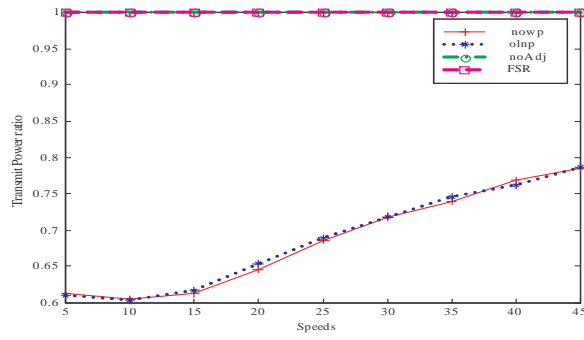**Fig. 4. (a)** Transmit power ratio comparison by number of nodes

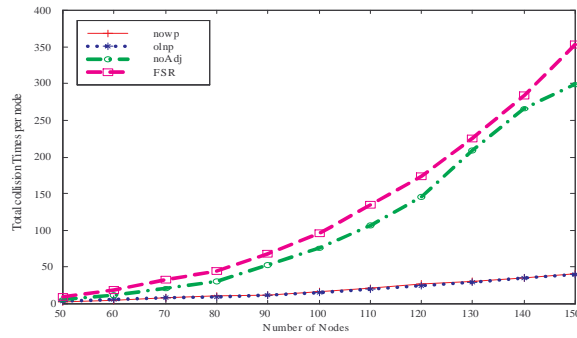**Fig. 4. (b)** Transmit power ratio comparison by mobility speeds

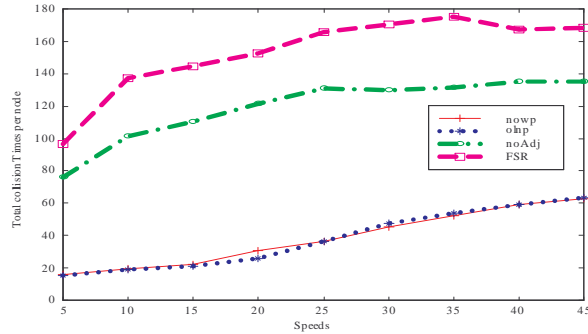**Fig. 5. (a)** Packet collisions per node comparison by number of nodes

Fig. 5. (b) Packet collisions per node comparison by mobility speeds

We can see how the transmit power ratio is impacted by number of nodes and the node moving speeds from Fig.4. With the number of nodes increasing, the node density increases because the terrain is constrained in 1000mx1000m, which means average distance between two nodes is shortening. In this situation, every node shrinks its transmit power ratio. From Fig.4.(a), we can see that we will save more power energy to prolong the life span of total network. For a special ad hoc wireless network composed of 100 nodes in the specified area, we can save about 36% transmit power energy. As shown in Fig.4.(b), transmit power ratio increases as the node moving speed increasing. The reason is that higher moving speed increases the probability of current active neighbor moving away the clusterhead. For maintaining the connection, the clusterhead must increase its transmit power, then the dominated neighbor increases its own transmit power so that it keeps connected with clusterhead.

Fig.5 shows how the packet collisions increase with the number of nodes and the node moving speeds increasing. Fig.5.(a) shows that collisions of **onAdj** and that of **FSR** increase rapidly as the node density increasing due to the node propagating broadcast to the network. Fig.5.(b) shows that collisions of all the four situations increase as the node speeds increasing. Because transmit power increases as the node moving speeds increase, more nodes will enter clusterhead covered area, so the average collisions increase.
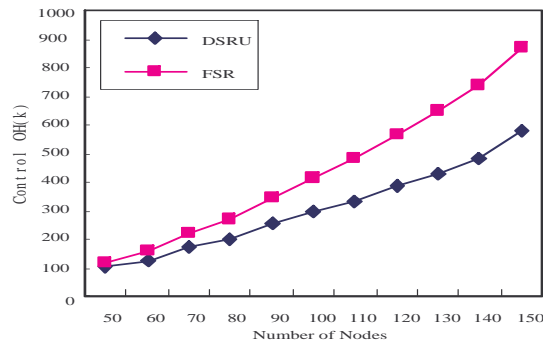

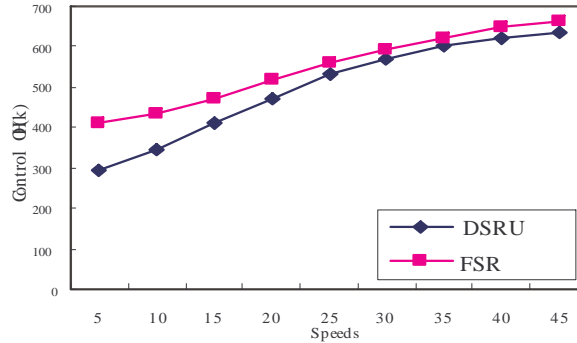**Fig. 6. (a)** Control Overhead comparison by number of nodes

Fig. 6. (b) Control Overhead comparison by mobility speeds

Fig.6.(a) reports the comparison of routing overhead between **FSR** and **DSRU** by the number of nodes. When the number of nodes increases, the routing overhead of the both algorithms increases. But the routing overhead of **DSRU** is always lower than that of **FSR**. Because in **FSR**, every node makes itself the centre of a circle and sends routing update message to other nodes with the frequency corresponding to the scope radius, while in **DSRU**, the routing update process is based on clusters. So when the number of nodes increased only a part of new nodes participate in the routing update process. Therefore the routing overhead of **FSR** is increasing faster than that of **DSRU**. Especially when the number of node is more than 100, **DSRU** reduces more than 40% of routing overhead compared with **FSR**.

Fig.6.(b) reports the comparison of routing overhead between **FSR** and **DSRU** by the moving speeds. When the node moving speed is slow the routing overhead of **DSRU** is much lower than **FSR**. With the speed increasing, the routing overhead of **DSRU** and **FSR** are both increasing, but even in the worst case that the node moving speed is very fast and the cluster can not be maintained, because the **DSRU** can dynamically reduce the scope of routing update, so the routing overhead of **DSRU** is still lower than that of **FSR**.

The performance evaluation of SQAR is under way. In the future work, we will add the results into the whole framework.

5   Conclusions

In this paper, we have proposed a cooperative three-tier framework for QoS routing in MANET. It tries to resolve the problem from three aspects: topology management, routing update and selecting path according to QoS parameter. The first two parts of the framework, SSCA and DSRU, have been implemented in GloMoSim simulator. The simulation results show that these two algorithms can effectively maintain the link stability between the nodes, limit the power consumption, and reduce the delay, collision times and packet loss ratio.

References

1. Chakrabarti, S., Mishra, A. QoS issues in Ad hoc wireless network. IEEE Communications Magazine, 2001,39(2):142~148.
2. Wang Hongbo, Zhang Yaoxue. Suitable size clustering algorithm for Ad hoc wireless networks. Journal of Software, September 2002.
3. Jin Xin, Wang Hongbo, Zhang Yaoxue. A Dynamic Self-adaptive Routing Update Algorithm for MANET. In the International Congerence on Communication Technology(ICCT,2003), Beijing, China.
4. Zonoozi, M., Dassanayake P. User mobility modeling and characterization of mobility patterns. IEEE Journal on Selected Areas in Communications, 1997,15(7):1239~1252.
5. Chiang, C.-C. Wireless network multicasting [Ph.D. Thesis]. Los Angeles: University of California, 1998.
6. Chan, J., Zhou, S., Seneviratne, A. A QoS adaptive mobility prediction scheme for wireless networks. In: Weber, J., ed. Proceedings of the IEEE Globecom'98. Sydney: IEEE Communication Society Publisher, 1998.
7. Gupta, P., Kumar, P.R. The capacity of wireless networks. IEEE Transactions on Information Theory, 2000,46(2):388~404.
8. H.Takagi, L.Leinrock. Optimal Transmission ranges for randomly distributed packet radio terminals. IEEE Transactions on Communications, 1984, 32(3): 246~257
9. T. Huo,V.O.K.li. Transmission range control in multihop radio networks. IEEE   Transaction on Communications, 1986, 34(1) :38~44.
10. S. Singh, M. Woo, and C. S. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In: Proceeding of ACM/IEEE MOBICOM '98, New York, 1998. 181~190.
11. Taek Jin Kwon, Mario Gerla. Clustering with Power Control. In Proceedings of IEEE MILCOM'99, Atlantic City, NJ, 1999.
12. Chunhung Richard Lin, Mario Gerla. Adaptive Clustering for Mobile Wireless Networks. IEEE Journal on Selected Areas in Communications, 1997, 15(7): 1265-1275.
13. Alan D.Amis,Ravi Prakash,Thai H.F. Vuong,Dung T.Huynlh. Max_Min D-Cluster Formation in Wireless Ad Hoc Networks. In Proceedings of INFOCOM'2000, Tel Aviv, Israel, 2000.
14. Rappaport, T.S. Wireless Communications, Principles and Practice. Prentice-Hall, 1996.
15. Chen, G., Garcia, F., Solano, J., et al. Connectivity based k-hop clustering in wireless networks. In: Sprague, R.H., ed. Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002). Hawaii: IEEE Computer Society Publisher, 2002.
16. M.Takai, L.Bajaj, R,Ahuja, R.Bagrodia and M.Gerla. GloMoSim: a scalable network simulation environment. Technical report 990027, UCLA,Computer Science Department, 1999.
17. R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H.Y.Song. PARSEC: a parallel simulation environment for complex systems. IEEE Computer, vol. 31, no. 10, Oct. 1998, pp.77-85.
18. Mario Gerla, Xiaoyan Hong, Guangyu Pei.," Fisheye state routing protocol (FSR) for ad hoc networks", In Internet-Draft, draft-ietf-manet-fsr-03.txt, Jule 2002