Performance Study of a Multipath Routing Method for Wireless Mobile Ad Hoc Networks*

Kui Wu, Janelle Harms
Computing Science Department
University of Alberta, AB, Canada T6G 2E8

Abstract

A Mobile Ad hoc NETwork (MANET) is a collection of wireless mobile computers forming a temporary network without any existing wire line infrastructure. Due to the dynamic nature of the network topology and the resource constraints, routing in MANETs is a challenging task. Multipath routing can increase end-to-end throughput and provide load balancing in wired networks. However, its advantage is not obvious in MANETs because the traffic along the multiple paths may interfere with each other. In addition, without accurate knowledge of topology, finding multiple disjoint paths is difficult. In this paper, we propose an on-demand method to efficiently search for multiple node-disjoint paths and present the criteria for selecting the multiple paths. We also perform a simulation study on the proposed method. The purpose of this paper is to present the advantages as well as some difficulties of deploying multipath routing in MANETs.

1. Introduction

A Mobile Ad hoc NETwork (MANET) is formed by a collection of mobile computers and does not rely on fixed based stations or a wired backbone infrastructure. Messages in MANETs may be forwarded through multiple hops due to the limitation of radio transmission range in every mobile computer. Finding paths, or routing, is an essential mechanism to support multiple-hop radio transmissions. However, node mobility and limited communication resources make routing in MANETs very difficult. Mobility causes frequent topology changes and may break existing paths. A routing protocol should quickly adapt to the topology changes and efficiently search for new paths. On the other hand, the limited power and bandwidth resources in MANETs make quick adaptation very challenging. More importantly, resource constraints in MANETs require a routing protocol to fairly distribute routing tasks among the mobile hosts. However, most proposed routing protocols for MANETs [1, 4] do not take fairness into account. They tend to have a heavy burden on the hosts along the shortest path from a source to a destination. As a result, heavily loaded hosts may deplete power energy quickly, which will lead to network partitions and failure of application sessions.

Multipath routing aims to establish multiple paths between source-destination pairs and thus requires more hosts to be responsible for the routing tasks. Although a lot of benefits have been explored for multipath routing in wired networks [5, 6, 7], the advantage of multipath routing is not obvious in MANETs because the traffic along different paths may interfere with each other due to the broadcast feature of radio transmission. Some protocols in MANETs, such as the Dynamic Source Routing (DSR) [1] and the Temporally Ordered Routing Algorithm (TORA) [4], use multiple paths. However, the multiple paths are utilized as a backup or auxiliary method in these protocols. In order to explore the benefits of multipath routing in MANETs, how to efficiently search for multiple paths, how to choose proper multiple paths, and how to use them deserve further study.

In [2], A. Nasipuri and S. R. Das prove that the use of multiple paths in DSR can keep correct end-to-end transmission for a longer time than a single path. In other words, the frequency of searching for new routes is much lower if a node keeps multiple paths to a destination. This is the first deep study on performance benefits of multipath routing in MANETs. However, they did not study the performance improvement of multipath routing on network load balancing. Their performance study is based on theoretical analysis, where it is difficult to take into account the influence of nodes' arbitrary movements and unreliable radio transmission. M. R. Perlman et al. demonstrate that multipath routing can balance network loads in [3]. They proposed a *diversity injection* method to find more node-disjoint paths compared to DSR. However, their work is based on multiple

---

*This research was supported by a grant from the Canadian Institute for Telecommunications Research under the NCE program of the Government of Canada.

0-7695-1315-8/01 $10.00 © 2001 IEEE

channel networks, which are contention free but may not be available in some application scenarios. In [9], we proposed a different method to search for node-disjoint paths, which could find more node-disjoint paths than the *diversity injection* method. However, the bandwidth cost for the control overhead is large, which may limit the benefits of deploying the multipath routing method in MANETs.

There are mainly two routing strategies in MANETs: proactive and reactive routing approaches. Proactive routing protocols try to maintain the routes to all possible destinations, regardless of whether or not they are needed. This category of routing protocols must periodically send control messages in order to maintain correct route information. They are also called table-driven protocols. Each node in these protocols usually has the whole or partial topology information. In contrast, reactive routing methods, which are also called on-demand methods, initiate the route discovery on demand of data traffic. Routes are only needed to those desired destinations. This routing approach can dramatically reduce the routing overhead when the network is relatively static and the active traffic is light. In this method, each node has little or no topology information. Without complete and accurate knowledge of the topology, how to efficiently find node-disjoint or edge-disjoint multiple paths is difficult. All problems discussed in this paper are based on the reactive, or on-demand, routing approach.

In this paper, we propose an on-demand approach with low control overhead to search for multiple node-disjoint paths. We also perform a simulation study on the proposed multipath routing method. The motivation of this paper is to explore the benefits and present the difficulties of deploying on-demand multipath routing in a shared channel wireless mobile network. The rest of the paper is organized as follows. In Section 2, we discuss the interference of the data traffic along different paths and its influence on the performance of multipath routing in MANETs. Section 3 introduces a multipath calculation algorithm based on an on-demand routing method. Section 4 describes the simulation model. We present our performance results in Section 5, and conclude the paper in Section 6.

2. Multiple Path Selection Criteria

If we assume all mobile hosts' radio transmission ranges are the same, then a MANET could be modeled as an undirected graph G = (*V*, *E*), where *V* is a set of $|V|$ nodes and *E* is a set of $|E|$ undirected links connecting nodes in *V*. Each node has a unique identifier and represents a mobile host with a wireless communication range of R. There is an undirected link (i, j) connecting two nodes i and j when the two nodes are within each other's transmission range.

In [9], we define a metric, *correlation factor*, to describe the interference of traffic between two node-disjoint paths. *The correlation factor (η) of two node-disjoint paths* is defined as the number of the links connecting the two paths. If there is no link (η=0) between two node-disjoint paths, we say the two node-disjoint paths are unrelated. Otherwise, the two node-disjoint paths are η-related. For example, in Fig. 1(a), two node-disjoint paths, S→a→b→c→D and S→d→e→f→D, are unrelated. But in Fig. 1(b), the two node-disjoint paths are 7-related. The total correlation factor of a set of multiple paths is defined as the sum of the correlation factor of each pair of paths. The correlation factor represents chances that the transmission along the different paths could interfere with each other in a shared channel model, in which all hosts use the same radio spectrum and compete for the radio channel. For example, in Fig. 1(b), if S is sending messages to a, then node d could not send messages to node e since both transmissions will collide at node a.

Through simulation, we demonstrate in [9] that in a static MANET with a shared channel radio model, the larger the correlation factor between two paths, the larger the average end-to-end delay for both paths. In addition, with the increase of the correlation factor, the difference between the average end-to-end delay along the two paths also increases even if the two paths have the same length and the same traffic load. When mobility is introduced, the initial selection of multiple paths with different correlation factors can influence the end-to-end delay when the mobile speed is low. However, with mobility, the initial selection of multiple paths with different correlation factors does not influence the routing performance in terms of control overhead, bandwidth cost for data transmission, and load balancing. We also demonstrate that if two paths are only edge-disjoint, good performance can not be guaranteed [9].

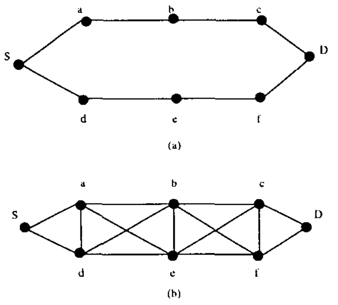

Figure 1. Different initial topologies

Our path selection criteria in MANETs include three properties: node-disjoint, small length difference between the primary (shortest) path and the alternative paths, and small correlation factor between any two of the multiple paths. Node-disjoint is the first standard to select multiple paths and how to search for multiple node-disjoint paths is one of the main concerns in this paper. A longer path will waste more bandwidth and increase the end-to-end delay. Multipath routing, however, may inevitably use longer paths since several shortest node-disjoint paths may not exist. The length difference between the shortest path and the alternative paths needs more buffer space in the destination to handle the disordered data packets and more time for message delivery. Thus, we should take the influence of different path length into account when selecting multiple node-disjoint paths. The third criterion is utilized because we have found that the initial selection of multiple path with small correlation factor can benefit the performance of end-to-end delay when mobile speed is not too fast [9].

3. Multipath Routing in MANETs

Finding node-disjoint multiple paths is not an easy task when the topology is unknown. The Dynamic Source Routing (DSR) [1] protocol finds multiple paths and uses the multiple paths as backup routes in case the used one breaks. However, it does not take the property of node-disjoint or edge-disjoint into account. We introduce our approach to find node-disjoint multiple paths and discuss how we use the multiple paths in this section.

3.1. Route Discovery in DSR

In DSR, if a source node does not know a route to a destination, it will initiate a route discovery by flooding a Route REQuest (RREQ) message. The RREQ message carries the sequence of hops it passed through in the message header. When a node receives a RREQ, if it is the first time for the node to receive this RREQ message, then this node will broadcast it again. Otherwise, the node will drop this RREQ packet. Once a RREQ message reaches the destination node, the destination node will reply with a Route REPly (RREP) packet to the source, using the reverse path contained in the RREQ packet. When the RREP packet traverses backward to the source, the source and all traversed nodes will know a route to the destination.

The method of broadcasting RREQ in DSR greatly reduces the possibility of finding multiple node-disjoint paths, because it quenches the diversity of the multiple paths. That is, the obtained multiple paths usually have some common nodes. In our simulation, using this method, the chance of finding node-disjoint multiple paths is almost zero. The reason is that later-received RREQ packets, which may include node-disjoint paths, are dropped at internal nodes. For example, in Fig. 1(b), if S broadcasts a RREQ, node a and d will receive and re-broadcast it. Assume node d transmits a little bit ahead of node a. Node b and e will receive the RREQ packet from node d and drop the later RREQ packet from node a. Finally, the destination D will just receive two paths having a common node, d.

3.2. Our Multipath Calculation Method

Definition 1 A *path P* between node v_1 and node v_n is an ordered sequence of distinct nodes $<v_1, \dots, v_n>$ such that (v_{i-1}, v_i) is a link for all $2 \leq i \leq n$. Node v_i is called an *internal node* if $1 < i < n$. If no path exists between node v_1 and node v_n, then node v_1 and node v_n are *partitioned.*

Definition 2 A *spanning tree* rooted at node S ($S \in V$) in a graph $G= <V, E>$ is a subgraph of G, which is a tree with root S and includes all nodes in V. A subtree of a spanning tree is called a *primary subtree* if this subtree's root's direct father is the root of the spanning tree. The *primary subtree nodes* in a spanning tree with root S is a set of nodes whose direct father is node S in the spanning tree.

Remark 1 Using the route discovery method in DSR, if two paths $P_1 = <S, v_1, \dots, v_n, D>$ and $P_2 = <S, u_1, \dots, u_m, D>$ received by node D have k ($k < m$ and $k < n$) common nodes excluding node S and D, then $v_1 = u_1$, $v_2 = u_2, \dots$, and $v_k = u_k$.

proof: Assume the route discovery method of DSR is used. Supposed to the contrary that two paths $P_1 = <S, v_1, \dots, v_n, D>$ and $P_2 = <S, u_1, \dots, u_m, D>$ received by node D have k ($k < m$ and $k < n$) common nodes (excluding node S and D) but there is an $i \leq k$ such that $v_i \neq u_i$. Then there must be a $j > k$ where $v_j = u_j$. Otherwise, the total number of common nodes could be less than k. Thus, node j will receive at least two RREQ packets: one was previously transmitted from v_i while the other was from u_i. However, according to the message broadcast method in DSR, node j will at least drop one of the packets because a node will not broadcast the same RREQ message twice. Therefore, only one path, either P_1 or P_2, could arrive at node D. Node D can not receive both P_1 and P_2. This is a contradiction. $\diamond$

From Remark 1, if the paths received by the destination are not node-disjoint, then only the first several hops of the paths are the same. So the union of all paths received by the destination will be a part of a spanning tree rooted at the source node if we do not consider the last hop of these paths. This phenomenon provides us with useful information to search for diverse multiple paths. Note that Remark 1 is true for any node that is reachable from the source.

Remark 2 For a spanning tree rooted at node S, if node v and node u belong to different primary subtrees and there is a direct link between node v and u, then there are at least two node-disjoint paths between node v (or node u) and node S.

proof: The proof is straightforward. Assume node v belongs to primary subtree B_1 and node u belongs to primary subtree B_2. Then one path is from node v (or u) to node S along the tree links in B_1 (or B_2). The other path is from node v (or u) to node u (or v) and then to node S along the tree links in B_2 (or B_1). The two paths are node-disjoint since node *u* and node *v* belong to different primary subtrees of the spanning tree. Note that two different primary subtrees of a spanning tree have no common nodes. ◇

Our multipath calculation method uses Remark 1 and Remark 2 as heuristic information to search for node-disjoint paths. In our method, the broadcast method of RREQ is very similar to DSR except that the later-received RREQ packets are cached instead of dropped in the middle nodes. A RREP packet in our method includes a label *isRedirection* to indicate whether the RREP packet should be redirected when traversing back to the source node. Due to Remark 1, to check whether two paths received by the destination are node-disjoint, we only need see if their first hops are the same. The first hops of the paths included in the RREQ packets are also the primary subtree nodes in the spanning tree rooted at the source node. When the destination node D receives a RREQ packet, if the path *P* included in this RREQ packet is node-disjoint with all paths included in previously received RREQ packets, then a RREP packet is sent to node S, using the reverse path of *P* and the label *isRedirection* is set to FALSE. Otherwise, a RREP packet is sent back, using the reverse path of P and the label *isRedirection* is set to TRUE.

When a middle node receives a RREP packet, it utilizes the algorithm in Fig. 2 to check the next hop to forward the RREP packet. The main idea here is to redirect those RREP packets whose *isRedirection* label is TRUE. The reason is that a RREP packet with *isRedirection* label of TRUE includes a path having some common nodes with a previously sent RREP packet, which is labeled as *beforeRrep* for the convenience of later description. The middle node then checks if it has cached a path to the source node, which is node-disjoint with the remaining hops included in the RREP packet. Based on Remark 2, redirecting the RREP packet to such a cached path will forward the RREP to a different primary subtree, where it is highly possible to find a path that is node-disjoint with the path included in the *beforeRrep*. Once the RREP packet is redirected, the label *isRedirection* in this packet is set to FALSE. Since the algorithm still uses source routing to forward the RREP, loops can be easily removed and the RREP packet will finally arrive at the source node.

In order to help the source nodes select good node-disjoint paths, we combine the path selection criteria in Section 2 with the above path calculation method as follows. Every node has a neighborhood table to record its neighbors. The contents of the neighborhood table are refreshed by any received control and data messages. If a neighbor has not been refreshed for a timeout value, it is obsolete and erased from the table. When the reply message traverses from the destination to the source node, it piggybacks the neighborhood information along the path. The source node will calculate the path correlation factor using the neighborhood information piggybacked in the reply message. Two parameters, *d*, which indicates the permitted maximal difference of the length between the primary (shortest) path and the alternate paths, and *f*, which is the permitted maximal total correlation factor among the selected multiple paths, are used in the node-disjoint paths selection.

The worst case of the communication complexity of our multipath searching method in terms of the number of message transmissions is O(N+ M*W), where N is the total number of mobile hosts, M is number of paths received by the destination which is less than or equal to the number of the destination's neighboring nodes, and W is the permitted maximal path length limited by the Time-To-Live (TTL) field in a packet. The first part, N, represents the number of message transmissions for one global flooding. The second part, M*W, is the maximal number of message transmissions for all RREP packets traversing back to the source node.

Note that our algorithm can not guarantee to find all node-disjoint paths. However, as we will show below through simulation, with the help of heuristic information, our method can efficiently find most node-disjoint paths. We compare the ability of finding multiple node-disjoint paths using the *diversity injection* method [3], DSR, and our method. The *diversity injection* method is proposed to increase the number of node-disjoint multiple paths compared to DSR. The broadcast of route requests in this method is the same as in ours. When a route reply message is received, the remaining path back to the source is replaced with "the shortest of the least selected cached paths that does not create a reply loop" [3]. The main difference between the *diversity injection* method and our approach is that the *diversity injection* method uses a different criterion for redirecting the RREP packets. As we will show below, the *diversity injection* method may redirect the RREP packets in a way that reduces the number of node-disjoint multiple paths found.

In the simulation model described in Section 4, we randomly choose 200 source-destination pairs and calculate the maximal number of node-disjoint paths between each source-destination pair, using an off-line algorithm (maximal flow) with knowledge of the whole topology. In the meantime, we separately use the *diversity injection* method, DSR, and our method to search for node-disjoint paths. In this experiment, we do not use the limits of maximal length difference (*d*) and maximal correlation factor (*f*) since we want to know the maximal number of node-disjoint paths that can be found. Fig. 3 shows the result of the ratio of the number of obtained node-disjoint paths using different searching methods to the maximal number of node-disjoint paths using the off-line algorithm. The result shows that our method can find most node-disjoint paths and more than the

```
if (the label isRedirection is set to FALSE in the RREP) or
  (there are no cached RREQ packets) {
        Forward the RREP packet to the prior hop along the path included in the RREP packet;
        Return.
}
else{
        Get S= {All cached RREQs that include paths whose first hop is different with the first hop of path P, where P is the
                reverse path included in the RREP};
        if (S is not empty) {
                repeat {
                        Get the cached RREQ with shortest route in S. If several cached routes in S have same shortest length,
                        then select one of them randomly;
                        Replace the remaining forward path back to the source in the RREP packet with the path in the RREQ;
                        if (the new RREP does not include a loop) {
                                Forward the new RREP packet using the new path;
                                Set the label isRedirection in the new RREP to FALSE;
                                Return;}
                        else Remove the RREQ from S;
                }
                until (all RREQ in S are checked or a proper RREQ in S, that can construct a new RREP without a loop, is
                        searched )
        }
        /* In the following case, S is empty or S does not include proper RREQ packets to satisfy the above requirement */
        Get $S_2$= {all cached RREQ – S};
        repeat{
                Search for a RREQ with shortest route in $S_2$. If several cached routes have same shortest length, then select one
                of them randomly;
                Replace the remaining forward path back to the source in the RREP packet with the path in the RREQ;
                if (the new RREP does not include a loop) {
                        Forward the new RREP packet using the new path;
                        Return.}
                else Remove from the RREQ from $S_2$;
        }
        until (a proper RREQ in $S_2$, that can construct a new RREP without a loop, is searched)
         /* The proper RREQ should be in $S_2$ since $S_2$ includes a path, which is the same as the one in the RREP. */
}
```

Figure 2. The algorithm for forwarding the RREP packets

diversity injection method; while DSR has almost no chance to find node-disjoint paths.

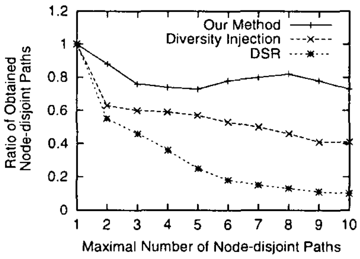

Figure 3. Ratio of the number of obtained node-disjoint paths to the number of maximal node-disjoint paths

3.3. Multipath Routing

There are several ways to use the multiple paths. In [1] and [2], the multiple paths are not used simultaneously. The data packets are transmitted along one path. Other paths are kept as backup paths in case the used one fails. When all possible paths are broken, a new multipath discovery procedure is promoted again.

Our approach of using the multiple paths is different. In order to balance the network load, we use the multipath simultaneously as in dispersity routing [7], which disperses the data traffic along different paths. Dispersity routing can be divided into redundant and non-redundant routing. In redundant dispersity routing, only part of the multiple paths are used to transfer data, and the other remaining paths are used to transfer redundant information such as error-correcting codes. In contrast, in non-redundant dispersity routing, all multiple paths are used to transmit data simultaneously. We use non-redundant dispersity routing. If a path fails, an error message is sent back to the source node and the traffic on that path will be transferred to other paths that are still alive. When all paths are broken, a new multiple path discovery is initiated again.

4. Simulation Model

We use a simulation model based on GloMoSim [8] to study the performance of multipath and unipath routing. The multipath routing includes our proposed method and the *diversity injection* method. Both methods choose a path to send data packets with a probability inversely proportional to the length of the path. Note that we just use the method of finding node-disjoint paths (*diversity injection*) from [3]. The underlying radio channel model and the way the multiple paths are used are different from what are used in [3]. The unipath routing method studied here is a simplified DSR, which does not include the optimization such as promiscuous learning of source route, leveraging the route cache, and piggybacking on route discoveries, etc. [1]. The reason to exclude these mechanisms in the simulation is that our main focus is on the performance difference when dispatching the traffic along single versus multiple paths.

In our simulation, the channel capacity of mobile hosts is set to the same value: 2 Mbps. A free space propagation model with a threshold cutoff is used as the channel model. In the free space model, the power of a signal attenuates as $1/r^2$, where r is the distance between mobile hosts. In the radio model, capture effects are taken into account. We use the Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs as the MAC layer protocol. It has the functionality to notify the network layer about link failures. The power consumption calculation is based on the NCR WaveLan [10] model. The power consumption in doze, receive, transmit mode is around 50mw, 900mw, and 1425mw respectively. We assume that a mobile host will be in doze mode immediately after transmitting or receiving a packet and can be revoked immediately before transmitting or receiving a packet. Note that a mobile host should passively receive any heard packets even if they are not for the mobile host. It is the network layer that decides how to process the packets.

In our simulation, 50 mobile nodes move in a 1500 meter x 500 meter rectangular region for 900 seconds simulation time. Compared with a square region, the rectangular region can enlarge the average route length so that we can easily observe the performance difference between unipath and multipath routing. Initial locations of the nodes are obtained using a uniform distribution. We assume each node moves independently with the same average speed. All nodes have the same transmission range of 250 meters. The mobility model is the random waypoint model. In this mobility model, a node randomly selects a destination from the physical terrain. It moves in the direction of the destination in a speed uniformly chosen between the minimal speed and maximal speed. After it reaches its destination, the node stays there for a *pause time* and then moves again. In our simulation, the minimal speed is 5 m/s and maximal speed is 10 m/s. We change the *pause time* from 0 seconds to 900 seconds to investigate the performance influence of different mobility. A *pause time* of 0 seconds presents continuous motion, and a *pause time* of 900 seconds corresponds to no motion.

The simulated traffic is Constant Bit Rate (CBR). 15 source nodes and 15 destination nodes were chosen randomly with uniform probabilities. The interval time to send packets is 250ms. The size of all data packets is set to 512 bytes. A packet is dropped when no acknowledgement is received after several retransmissions or there is no buffer to hold the packet. The buffer size is set to 64 packets. The maximal total correlation factor is set to 15. The maximal number of multiple paths is 4. The maximal difference of the length between the shortest path and the alternative paths is 3. And the maximal path length is 9. All traffic is generated and the statistical data are collected after a warm-up time of 30 seconds in order to give the nodes sufficient time to finish the initialization process. For each scenario, eight runs with different random seeds were conducted and the results were averaged. When calculating the confidence intervals, the confidence levels are set to 95%.

5. Simulation Results

5.1. Performance Metrics

We will compare the performance of unipath routing and multipath routing under different mobility. We evaluate the performance according to the following metrics:

- *Control overhead*: The control overhead is defined as the total number of routing control packets normalized by the total number of received data packets.
- *Bandwidth cost for data*: The bandwidth cost for data is defined as the total number of data packets transmitted at all mobile hosts normalized by the total number of received data packets.
- *Average end-to-end delay*: The end-to-end-delay is averaged over all surviving data packets from the sources to the destinations. It includes queuing delay and propagation delay.
- *Load balancing*: We use a graph G=(V, E) to denote the network, where V is the node set and E is the link set. We define a state function $f : V \to I$ where I is the set of positive integers. f(v) represents the number of data packets forwarded at node v. Let CoV (f) = standard variance of f / mean of f. We use CoV (f) as a metric to evaluate the load balancing. The smaller the CoV (f), the better the load balancing.
- *Energy balancing*: As above, we use CoV(g) to evaluate the energy balancing, where g(v) represents the energy consumption at each node. The energy consumption in a node is calculated as the sum of T_i * P_i (i = 0, 1, 2), where T_i represents the time spent in the three different modes (doze, receive, and transmit mode) and P_i represents the power consumption in the corresponding modes.

- *Average energy consumption*: The energy consumption is averaged over all nodes in the network.

5.2. The Performance Comparison of Multipath Routing with Unipath Routing

Fig. 4 shows the result of total number of route discovery phases versus the mobility. The frequency of route discovery for multipath routing is less than for the unipath routing. This result is coincident with the theoretical analysis in [2]. The frequency of route discoveries for our multipath routing is less than the *diversity injection* method since our method could find more node-disjoint paths as shown in Fig. 3. Since every route discovery needs approximately the same control overhead in our multipath method, the *diversity injection* method, and the unipath routing method, the reduction of route discovery frequency will reduce the total control overhead. Fig. 5 shows that the control overhead for our multipath routing is the lowest among the three approaches. This is an important improvement over our method in [9].

Although the trend of the total control overhead increases with increased mobility, we observe that the curves in Fig. 5 are not strictly "smooth". This is because the connectivity of the network is different at different speeds. As the average speed is increased, for a given simulation time, the number of simulated nodes' movements increases. Thus, although the simulated mobile model is the same, a particular network configuration, such as the occurrence of partitions, that may not have occurred at a lower speed could occur at the higher speed. This phenomenon could be observed in [11, 12] as well. A network partition is a temporary state when a network node has no way of reaching another network node. This is likely the main reason for the non-monotonic curves in Fig. 5 and later figures.

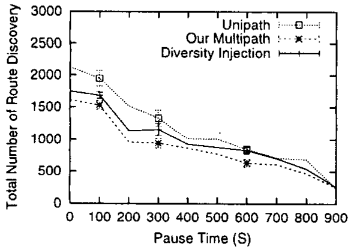

Figure 4. The number of route discovery

Fig. 6 shows the result of total bandwidth cost for data transmission. The bandwidth cost of data transmission for the unipath routing tends to be the smallest. This is because the unipath routing usually uses the optimal path from a source to a destination. The alternate paths in multipath routing are usually sub-optimal, which will cost more bandwidth. An interesting phenomenon is that the bandwidth cost for data transmission in the *diversity injection* method tends to be larger than our method at low speed. However, the variability in the results may indicate that these differences are not significant.

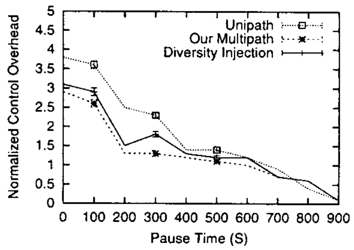

Figure 5. The normalized control overhead

Fig. 7 shows the results of average end-to-end delay. The end-to-end delay includes the queuing delay in every host and the propagation delay from the source to the destination. Multipath routing will reduce the queuing delay because the traffic is distributed along different paths. On the other hand, it will increase the propagation delay since some data packets may be forwarded along sub-optimal paths. From Fig. 7, the unipath routing has higher average end-to-end delay compared to our multipath routing. This demonstrates that our multipath routing could distribute the traffic and improve the end-to-end delay, but the improvement is limited below *pause time* of 300 seconds. With the decrease of *pause time*, the average end-to-end delay for both multipath routing and unipath routing increases, because the network topology changes more frequently. More route discoveries will be promoted and thus the queuing delay of the data packets in the source nodes increases, which leads to the increase of the average end-to-end delay.

The *diversity injection* method, however, shows a larger end-to-end delay than the unipath routing method at low speed. The main reason for this strange phenomenon is that the *diversity injection* method uses fewer and longer node-disjoint paths. As described above, the *diversity injection* method use the shortest cached route, which has been used the least number of times, to re-direct RREP packets. However, this does not mean that the source node will finally receive shorter node-disjoint paths than our method. The criteria of least used and the possible multiple redirecting may result in longer paths. In contrast, our method only re-directs a RREP packet once when a proper direction is found. Through observation, those RREP packets that happen to be re-directed to longer paths are much likely to include node-disjoint paths in the *diversity injection* method. In other words, the *diversity injection* method finds fewer and longer node-disjoint paths. The improvement of end-to-end delay due to traffic distribution can not compensate for the degradation of end-to-end delay when using longer paths. It shows that multipath routing does not improve end-to-end delay in all scenarios. This is an important lesson on deploying multipath routing in MANETs.

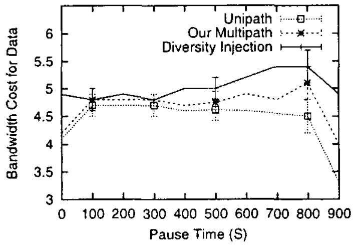

Figure 6. The bandwidth cost for data

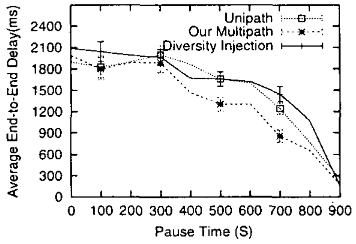

Figure 7. The average end-to-end delay

Fig. 8 gives the results of load balancing. The CoV of network load is the highest for the unipath routing and the lowest for our method. This is because the multipath routing can distribute the network traffic along different paths. On the other hand, the unipath routing always uses the shortest paths between the sources and the destinations, which will unfairly assign more duties to the nodes along the shortest paths. With the decrease of *pause time*, the CoV of network load for the unipath routing and the multipath routing also decreases. This shows that the increase in mobility could result in better load balancing of the traffic among the nodes. "Hot spots" are likely removed due to mobility. Our method is better than the *diversity injection* method in terms of load balancing because our method could find more and shorter node-disjoint paths.

5.3. The Energy Consumption Comparison

Based on the NCR WaveLan model, Fig. 9 shows the results of energy balancing. The CoV of energy consumption for the unipath routing is higher than that for our multipath routing. This demonstrates that our multipath routing can assign the routing tasks more fairly than the unipath routing. However, note that the scale of the y-axis is much smaller than the one in Fig. 8. The improvement of energy balancing with multipath routing is trivial compared to the improvement of load balancing. This is because the nodes, even with no routing tasks, have to passively listen to neighboring nodes' radio transmission, which inevitably consumes battery energy. This is also the reason why the *diversity injection* method presents very similar results of energy balancing as the unipath routing.

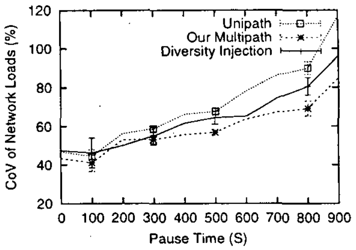

Figure 8. The CoV of network load

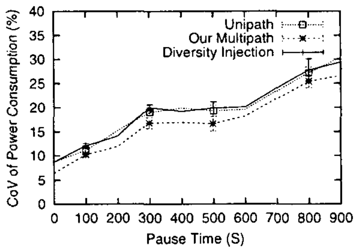

Figure 9. The CoV of energy consumption

From the results in Fig. 10, we can see that our multipath routing and the *diversity injection* method have smaller average energy consumption than the unipath routing when mobile speed is high (pause time is less than 400 seconds in the simulation). When mobile speed is low, the average energy consumption in the unipath routing is smaller than in our multipath routing and the *diversity injection* method. The battery energy of a network node is mainly consumed on forwarding control and data packets. Multipath routing usually increases the energy consumption on the transmission of data messages because some data packets traverse sub-optimal paths. On the other hand, it will decrease the energy consumption on the transmission of control messages from the results in Fig. 5. When mobile speed is high, the energy cost on routing control is too large to compensate for the energy saving on data transmission along the optimal paths in the unipath routing. This is why energy consumption is higher in the unipath routing when mobile speed is high.

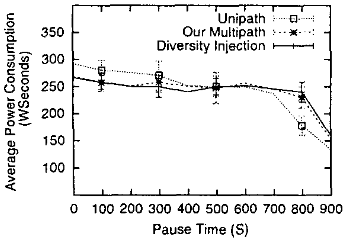

Figure 10. The average energy consumption

6. Conclusions

Without accurate knowledge of topology, how to find multiple node-disjoint paths is difficult. In this paper, we propose a new on-demand multipath calculation method based on the heuristic redirection of RREP packets. We also studied the performance of the proposed on-demand multipath routing method. The results show that our multipath routing can provide load balancing, reduce the frequency of route discovery and control overhead, provide fair energy consumption among the network nodes, and save total energy consumption when mobile speed is high.

References

[1] J. Broch, D. Johnson, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks," *http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-04.txt*, Nov. 2000, IETF Internet Draft.

[2] A. Nasipuri and S.R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," *Proceedings of the 8th Int. Conf. On Computer Communications and Networks (IC3N)*, Boston, October 1999.

[3] M.R. Pearlman, Z.J. Hass, P. Sholander, and S.S. Tabrizi, "On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks," *Proceedings of IEEE/ACM MobiHoc 2000*, Boston, Aug. 2000.

[4] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of IEEE INFOCOM'97 Conf.* , April 1997.

[5] H. Suzuki and F.A. Tobagi, "Fast Bandwidth Reservation Scheme with Multi-link & Multi-path routing in ATM networks," *Proceedings of IEEE INFOCOM*, 1992.

[6] P. Georgatsos and D. Griffin, " A Management System for Load Balancing Through Adaptive Routing in Multiservice ATM Networks," *Proceedings of IEEE INFOCOM*, 1996.

[7] N.F. Maxemchuk, "Dispersity Routing in High-speed Networks," *Computer Networks and ISDN system*, 1993, Page 645-661.

[8] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: a Library for Parallel Simulation of Large-Scale Wireless Networks," *Proceedings of PADS'98*, Banff, Canada, May 26-29, 1998.

[9] K. Wu and J. Harms, "On-demand Multipath Routing for Mobile Ad Hoc Networks," *Proceedings of 4th European Personal Mobile Communication Conference (EPMCC)*, Vienna, Austria, Feb. 2001.

[10] Lucent Technologies, "*WaveLan product specification*", http://www.wavelan.com, 2000.

[11] S.R. Das, C.E. Perkins, and E.M. Royer, "Performance Comparison of Two On-Demand Routing Protocols for Ad-Hoc Networks, " *Proceedings of IEEE INFOCOM'00*, March 2000.

[12] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols, " *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Dallas, TX, October 1998.