Adaptive Multipath Source Routing in Ad Hoc Networks

Lei Wang, Yantai Shu, Miao Dong, and Lianfang Zhang
Department of Computer Science
Tianjin University
Tianjin 300072, China
w-lei@excite.com

Oliver W.W. Yang
School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5
yang@site.uottawa.ca

Abstract —**In this paper, we propose a new multipath routing protocol for ad hoc wireless networks — Multipath Source Routing (MSR), which is an extension of DSR (Dynamic Source Routing). Based on the measurement of RTT, we propose a scheme to distribute load among multiple paths. The simulation results show that our approach improves the packet delivery ratio and the throughput of TCP and UDP, and reduces the end-to-end delay and the average queue size, while adding little overhead. As a result, MSR decreases the network congestion and increases the path fault tolerance quite well.**

I. INTRODUCTION

Congestion at the links and in the routers is the main cause of large delays in the Internet; the same is true in ad hoc networks where bandwidths are always very limited. Routing protocols used in conventional wired networks (e.g., Bellman-Ford and link state) are not well suited for the mobile environment due to the considerable overhead produced by periodic route update messages, and to their slow convergence to topological changes. In addition, all the Internet routing protocols in use today rely on single-path routing algorithms, which not only under-utilize resources, but also cannot cope with congestion and link breakage. This can be attributed to the fact that all traffic for a destination is required to be routed through a single successor. So when a link becomes congested or broken, its entire carried traffic has to be rerouted; this becomes more time consuming in mobile networks. If link costs are made functions of congestion or delays, routing table entries can become unstable in single-path routing protocols.

Multipath routing can overcome the above problem. In addition, it can provide load balancing and route failure protection by distributing traffic among a set of diverse paths. These benefits make multipath routing a good candidate for bandwidth-limited and mobile ad hoc networks. However, maintaining alternative paths requires much more routing table space and computation, especially for table-driven routing algorithms in large networks. Many ad hoc routing protocols have been proposed recently, such as AODV, DSDV, DSR, and TORA [9]. However, they are all single-path based[1]. DSR (Dynamic Source Routing) is capable of reducing communication and computation overhead. Nasipuri et al [10, 11] had proposed an extension to DSR independently. In their work, they developed analytical models to demonstrate how the frequency of query floods is reduced with their multipath extensions, and presented some numerical data obtained from the analytic models. They also did some simple and preliminary simulations; there are still a lot of work regarding multipath routing in wireline networks (see [3] and references therein for detail).

In this paper, we propose a new approach called the Multipath Source Routing (MSR) for multipath routing in ad hoc wireless networks. This is an extension of DSR (Dynamic Source Routing).[2] Our work focuses on the adaptively distributing load among several paths, according to the measurement of the round-trip time of every path, whereby a heuristic load-balancing equation is given. To the best of our knowledge, this is the first trial to introduce probing-based feedback control to multipath routing. An RTT measurement tool for DSR and MSR in simulation, `SRping` is developed to get the RTT between two arbitrary nodes. It is also the first attempt to analyze the TCP dynamics in ad hoc networks and multipath routing context. Our results show that MSR can reduce the traffic burst ness seen by individual paths, adapt to frequently topology changes and consequently achieve inherent robustness to channel errors and link failures. We also analyze and compare the packet dynamics of MSR and DSR in depth.

II. DYNAMIC SOURCE ROUTING

DSR [6, 1] uses source routing instead of hop-by-hop packet routing. Each data packet carries the complete path from source to destination as a sequence of IP addresses. The main benefit of source routing is that intermediate nodes need not keep route information because the path is explicitly specified in the data packet. DSR is on-demand based; that is, it does not require any kind of periodical message to be sent. The source routing mechanism, coupled with the on-demand nature of this protocol, eliminates the need for the periodic route advertisement and neighbor detection packets in other protocols.

The DSR protocol consists of two mechanisms: Route Discovery and Route Maintenance. Route discovery is initiated by a source whenever a source has a data packet to send but does not have any routing information to the destination. To establish a route, the source floods the network with request messages carrying a unique request ID. When a request message reaches the destination or a node that has route information to the destination, the node sends a route reply message containing path information back to the source. In order to reduce overhead generated during a route discovery phase, the "route cache" maintained at each node records routes the node has learned and overheard over that time frame.

Route Maintenance is the mechanism by which the sender S of a packet detects network topology changes that render useless its route to the destination D (because two nodes listed in

[1] DSR and TORA [13] can provide multiple paths, but only one path is used.

[2] A preliminary work was [16]

0-7803-7097-1/01/$10.00 ©2001 IEEE

the route have moved out of range of each other). When Route Maintenance indicates a source route is broken, S is notified with a ROUTE ERROR packet. The sender S can then attempt to use any other route to D already in its cache or can invoke Route Discovery again to find a new route.

III. MSR (MULTIPLE SOURCE ROUTING)

By using source routing, MSR can improve performance by giving applications the freedom to use multiple paths within the same path service. However, maintaining alternative paths requires more routing table space and computational overhead. Fortunately, some DSR's characteristics can suppress these disadvantages. First, Source Routing is so flexible that messages can be forwarded on arbitrary paths, which makes it very easy to dispatch messages to multiple paths without demanding path calculation in the intermediate hops. Second, the on-demand nature of DSR reduces the routing storage and routing computation greatly.

A. Path finding

MSR retains the route discovery mechanism of DSR whereby multiple paths can be returned. Each route discovered is stored in the route cache with a unique route index. So it is easy for us to pick multiple paths from the cache. In multipath routing, path independence is an important property, because a more independent path set offers more aggregate physical resources between a node pair (because when those resources are not shared, the less likely the performance of one path affects the performances of other paths). To achieve high path independence, the disjoint paths are preferred in MSR[3]. There is no looping problem in MSR, as the route information is contained inside the packet itself; routing loops, either short- or long-lived, cannot be formed as they can be immediately detected and eliminated.

B. Packet forwarding and load balancing

Since MSR uses source routing, intermediate nodes do nothing but forwarding the packet as indicated by the route in its header, thus adding no more processing complexity than that in DSR. All the work for path calculation is done in the source hosts. In MSR, source nodes are responsible for load balancing. In our protocol, we implement a special table containing multiple path information to the specific destination, as illustrated as follows.

```
struct mul_dest
{
    int index ;
    ID Dest;
    float Delay;
    float Weight;
              ...
};
```

Dest is the destination of a route. *Index* is the current index of the route in DSR's route cache that has a destination to *Dest*. *Delay* is the current estimate of the round-trip time. *Weight* is a per-destination based load distribution weight between all the routes that have the same destination. *Weight* is in terms of the number of packets to be sent consecutively on the same route every time. The choice of *Weight* is an interesting and challenging task, and we make the following observation.

Within an ad hoc network, which is always an autonomous system acting as a stub network, there is less heterogeneity in some sense when compared to WAN and MAN. For instance, in WAN or MAN the maximal bandwidths that every node can obtain vary little, so do the round-trip delays. Therefore, we assume the bandwidth-delay product is a constant. Then the available bandwidth is inversely proportional to the RTT. So the traffic can be distributed among multiple paths proportional to the available bandwidth. The principle is inherently simple but reasonable in wireless networks. In wireline networks, due to the very different bandwidths, delay cannot be a definite indicator of the available bandwidth.

From our above observation, we propose to choose the weight W_i^j (i is the index of the route to j) according to a heuristic equation (1):

$$W_i^j = \mathrm{Min}_j\left(\left\lceil \frac{d_{\max}^j}{d_i^j} \right\rceil, U\right) \times R \qquad (1)$$

where $d_{\max}^j$ is the maximum delay of all the routes to the same destination, d_i^j is the delay of route with index i, and U is a bound to insure that W_i^j should not to be too large. R is a factor to control the switching frequency between routes. The larger the R's value, the less frequently the switching happens and the less processing overload of searching and positioning an entry in the `mul_dest` table. When choosing R, the IFQ[4] buffer's size should also be taken into considerations. Unlike the work done in [5, 10, 11], we have done extensive experiments beyond the $R = 1$ work in [16] and we found $R = 3$ to be better in reducing the out-of-order deliveries in TCP. So in our experiment, R is set to three for IFQ size of 50. When distributing the load, the weighted-round-robin scheduling strategy is used.

To aid the load balancing and to decouple the interlayer dependence of delay measurement, a network layer probing mechanism is employed. Probing is also an enhancement to the DSR Route maintenance mechanism. Normally, in DSR, a link breakage can be notified only when a Route Error message is returned. However, in wireless mobile environment, it has a nontrivial chance that the Route Error message cannot reach the original sender successfully. Although, "as a last resort, a bit in the packet header could be included to allow a host transmitting a packet to request an explicit acknowledgement from the next-hop receiver [1]", probing one path constantly only to test its validity is not cost effective. Therefore, the function of probing in MSR is twofold: to get the path delay status and to test the validity of active paths.

C. Toward gratuitous mode

We should explain gratuitous packet here. In DSR, when a data packet is received as the result of operating in promiscuous receive mode, the node checks if the Routing Header packet contains its address in the unprocessed portion of the source route. If so, the node knows that packet could bypass the unprocessed hops preceding it in the source route. The node then sends what is called a gratuitous Route Reply message to the packet's source, giving it the shorter route without these hops [1]. Since in MSR, there are always routes that are not the shortest ones,

---

3 In this paper we address the multipath routing problem in the context of single channel, for the disjoint path problem in multiple channel environment refer to [14].

4 The network stack for a mobilenode consists of a link layer (LL), an ARP module connected to LL, an interface priority queue (IFQ), a MAC layer (MAC), a network interface (netIF), all connected to the channel [12].

the GRAT (GRATuitous) packets increase greatly, which take too much IFQ and ARP buffer space. Thus, we turn off the gratuitous options in our simulations.

IV. PERFORMANCE EVALUATION

A. Simulation environment

We use *ns* to conduct the simulation. CMU has extended *ns* with some wireless supports, including new elements at the physical, link, and routing layers of the simulation environment. Using these elements, it is possible to construct detailed and accurate simulations of wireless subnets, LANs, or multi-hop ad hoc networks. For scenario creation, two kinds of scenario files are used. The first is a movement pattern file that describes the movement that all nodes should undergo during the simulation. The second is a communication pattern file that describes the packet workload that is offered to the network layer during the simulation.

To get the performance of MSR under different moving speeds environment, we use two simulation sets with speed 1m/s and 20m/s respectively. Our simulations model a network of 50 mobile hosts placed randomly within a 1500m×300m area, both with zero pause time. To evaluate the performance of MSR, we experimented with different application traffic, including CBR and FTP. CBR uses UDP as its transport protocol, and FTP uses TCP. The channel is assumed error-free except for the presence of collision. For other simulation detail, please refer to [2].

B. Performance Metrics

In performance evaluation, we choose the following metrics:

- Queue size: The queue size of an IFQ object at each node;
- Packet delivery ratio: The ratio between the number of packets originated by the "application layer" CBR sources and the number of packets received by the CBR sink at the final destination;
- Data throughput: The total number of packets received during a measuring interval divided by the measurement interval;
- End-to-end delay;
- Packet drop probability.

For TCP, another issue concerned is the out-of-order problem described in [5]. To present the packet dynamics clearly, the ack time-sequence plot is given.

C. Simulation Results

1) *UDP traffic (max moving speed of 20m/s)*

We first look at CBR traffic implemented with UDP agents. A scenario with 20 CBR connections is adopted. Since UDP has no feedback control mechanism, all the CBR traffic generated is constant no matter how the network runs. So it can act as a good testbed for comparing routing protocols. We shall use it as a reference point. Fig. 1 shows that fewer CBR packets are dropped in MSR than that in DSR. Table I shows drop summery in detail; the main reason of dropping is "No Route" and "IFQ Full." Fig. 2 provides the end-to-end packet delivery ratio of every connection, and the comparison shows MSR is better than DSR.

TABLE I
CBR PACKET DROPS SUMMERY.

CBR Drops Summery	DSR	MSR
Total Packets:	217	114
Total Bytes:	119976	63096
No Route:	129	90
TTL Expired:	0	0
RTR Queue Full:	0	0
Timeout:	0	0
Routing Loop:	0	0
IFQ Full:	30	9
ARP Full:	53	14
MAC Callback:	0	0
SIM End:	5	1

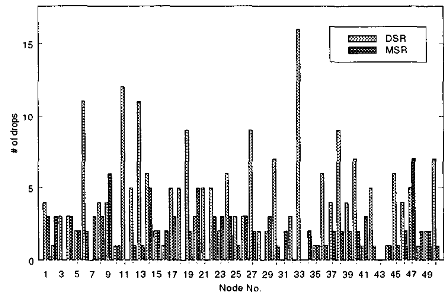


Fig. 1. CBR packets dropped at each node

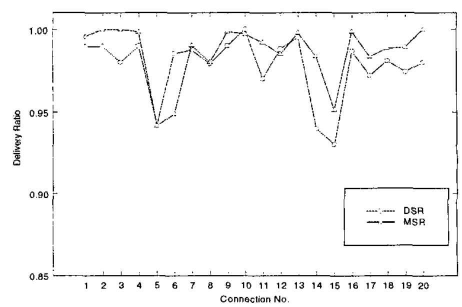


Fig. 2. Packet delivery ratio of every connection.

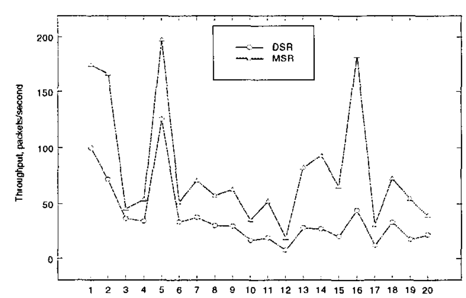


Fig. 3. End-to-end throughput.

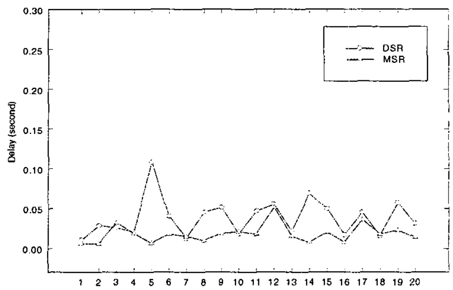


Fig. 4. End-to-end delay

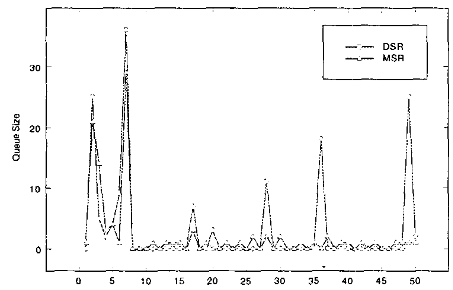


Fig. 5. IFQ queue size at each node

From Fig. 3 we can see that MSR achieves higher throughput than DSR almost on every connection, just as we expected. This can be attributed to the fact that the multipath routing effectively utilizes currently unallocated network resources. Fig. 4 shows the end-to-end delay of every connection. Fig. 5 presents the average queue size for all 50 hosts. From Fig. 5, we can see that, in MSR, the packets that should have been queued in the IFQ have been redistributed to other nodes that have light load, through which the traffic is balanced. Balancing the route load in MSR shortens the delay as the chance of congestion is reduced.

Table II shows the routing overheads in DSR and MSR respectively. We can see the routing messages in DSR are only little more than that of DSR. However, the packet drops probability is lower than that of DSR. The main drop reason is still "No Route" and "IFQ Full."

TABLE II.
ROUTING LEVEL (IN PACKETS) STATISTICS FOR CBR

Type	DSR	MSR
DSR TOTALS Transmitted	2200	2313
DSR TOTALS Received	21786	25970
DSR TOTALS Forwards	2740	3392
DSR TOTALS Drops	56	58
IFQ len above 25	100	73
DSR REQUEST Transmitted	199	189
DSR REQUEST Received	18255	19389
DSR REQUEST Forwards	1145	1232
DSR REQUEST Drops	0	1
DSR REPLY Transmitted	72	89
DSR REPLY Received	202	318
DSR REPLY Forwards	137	233
DSR REPLY Drops	5	4
DSR ERROR Transmitted	596	676
DSR ERROR Received	1085	1261
DSR ERROR Forwards	492	593
DSR ERROR Drops	6	2

2) *TCP traffic (max moving speed 20m/s)*

For TCP traffic, we take a scenario with 30 FTP connections, with network rather heavily loaded. Since TCP has an AIMD (Additive Increase Multiplicative Decrease) feedback control mechanism, the statistics at every node has less meaning than that of UDP, we focus on the end-to-end packet dynamics instead. Figs. 6 and 7 show that the multipath routing can also be used to reduce the end-to-end network latency and message drop probability, or increase the likelihood of message delivery for TCP connections.

From Figs. 8 and 9, we can see there are not many out-of-order deliveries in MSR. On the contrary, the end-to-end throughput of TCP in MSR has increased a lot due to the smooth increase of sequence number. Fig. 9 also implies that MSR recovers more quickly than DSR does when the connection meets severe packets droppings (e.g. at time 90s). It illustrates that our load-balancing method achieves a good switching granularity.

D. Discussions

In our initial experiments, we found that the major statistics of Routing Packets of MSR is comparable to that of DSR, except that the GRAT packets count in MSR is too large compared to DSR. Thus, we turn off the gratuitous options, and the results become better. The simulation results show that the main reason for packet drops in DSR is "No Route" and "IFQ Full", while these two factors improve a lot in MSR.

Under max speed 1m/s, the throughput and end-to-end delay of MSR are also better than those of DSR. There is no significant difference of packet drops between DSR and MSR. Therefore, we can conclude that one of the mail gains we get from MSR is attributed to less "No Route" drops. In other words, multipath routing compensates for route failures efficiently in high-speed movement. It is consistent with the results in Tables I and II. Due to space limitations, we do not give the results here.

When evaluating a network routing protocol, control load should also be considered. There is no more control load in MSR than that in DSR, except for probing packets transmitted in networks. Since we use `SRping` (which is unicast), rather than flooding, to test the validity of paths currently used, and the probing interval we choose is very conservative, there is little overload added.

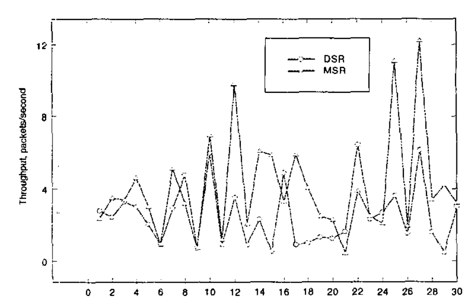


Fig. 6. End-to-end throughput of each connection

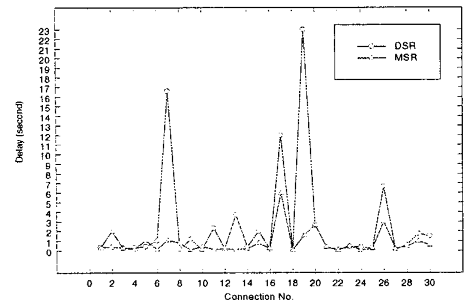
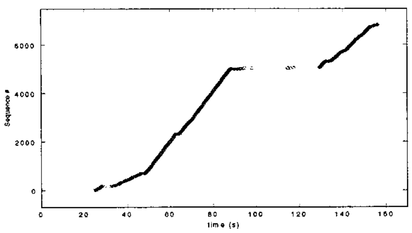
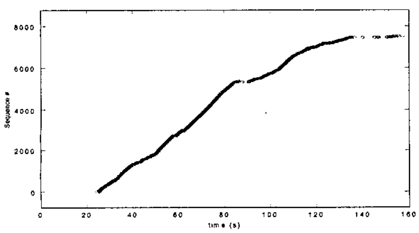


Fig. 7. End-to-end delay of each connection.



DSR



MSR

Fig. 8. TCP segments time-sequence plot of a heavy connection, node 49 to node 50.

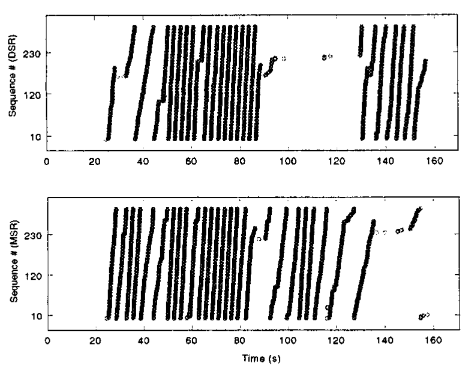


Fig. 9. Time-Sequence plot with sequence # mod 300.

V. Conclusions

In this paper, a new multipath routing protocol in ad hoc networks, MSR is presented. Our protocol is a direct descendant of DSR. By incorporating the multipath mechanism into DSR and employing a probing based load-balancing mechanism, the throughput, end-to-end delay, and drop probability have been improved significantly. The drawback of MSR may be the processing overhead of originating the packets. Fortunately the computer is becoming more powerful and cheaper. So it may not be the obstacle to the deployment of MSR. The future work might include QoS support in MSR and MSR under multi-channel environment.

Acknowledgement

Thanks to *ns* and its CMU wireless extensions, without them this work is impossible. This research was supported in part by the National Natural Science Foundation of China (NSFC) under grant No. 69872025, a Cisco grant, and the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant No. OGP0042878.

References

[1]. J. Broch, D. B. Johnson, and D. A. Maltz. "The dynamic source routing protocol for mobile ad hoc networks." http://www.ietf.org /internetdrafts/draft-ietf-manet-dsr-03.txt, June 1999.

[2]. 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." In *ACM MOBICOM'98*, pp. 85–97, October 1998.

[3]. J. Chen, *New Approaches to Routing for Large-Scale Data Networks*, PhD Thesis, 1998.

[4]. D. Kourkouzelis, *Multipath Routing Using Diffusing Computations*, Master Thesis, 1997.

[5]. N. Gogate and S.S. Panwar, "Supporting Applications in a Mobile Multihop Radio Environment Using Route Diversity," I. Non-Real Time Data, *IEEE ICC'98*, Atlanta, GA, June 1998.

[6]. D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, 1996, pp. 153-181.

[7]. P. Karn and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", In *ACM SIGCOMM*, pp. 2–7, August 1987.

[8]. R. Krishnan and J. Silvester, "Choice of Allocation Granularity in Multipath Source Routing Schemes," *IEEE INFOCOM'93*, pp. 322-329, March 1993.

[9]. MANET. http://www.ietf.org/html.charters/manet-charter.html.

[10]. A. Nasipuri and S. R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," *IEEE ICCCN*, Boston, MA, Oct. 1999.

[11]. A. Nasipuri, Robert Castaneda and S. R Das, "Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks," *Mobile Networks and Applications*, 2000.

[12]. ns, http://mash.cs.berkeley.edu/ns.

[13]. V. D. Park and M. S. Corson. "Temporally-Ordered Routing Algorithm (TORA) version 1: Functional specification." Internet-Draft, draft-ietf-manet-tora-spec-00.txt, November 1997.

[14]. M. Pearlman, Z. Haas, P. Sholander, and S. S. Tabrizi, "On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad-Hoc Networks," *MobiHoc'2000*, Boston, USA, August 11, 2000.

[15]. C. E. Perkins. "Ad Hoc On Demand Distance Vector (AODV) routing," Internet-Draft, draft-ietf-manet-aodv-03.txt, June 1999.

[16]. L. Wang, L. F. Zhang, Y. T. Shu, M. Dong, and O. W. W. Yang, "Multipath Source Routing in Wireless Ad Hoc Networks," in Proceedings of the *IEEE CCECE*, pp. 479-483, Halifax, Nova Scotia, Canada, May 7-10, 2000.