Timely Backup Source Routing for Multimedia Transport in Wireless Ad-hoc Network

Ming-Ho Hsiao, Yi-Wen Chen, Hua-Tsung Chen, Chao-Ching Ma, Suh-Yin Lee
College of Computer Science, National Chiao Tung University
{ mhhsiao, ewchen, huatsung, majc, sylee}@csie.nctu.edu.tw

ABSTRACT

Routing protocols for wireless ad-hoc networks consider the path with the minimum number of hops as the optimal path to any given destination. However, this strategy does not balance the traffic load among the mobile network nodes, and may create congested areas. These congested areas greatly degrade the performance of the routing protocols. Prior research in routing for wireless ad hoc networks has shown that multi-path routing can enhance data delivery reliability and provide load balancing. Nevertheless, they obtain the backup route at the earlier moment based on the rough route load metric. In this paper, we propose a novel routing scheme that balances the load over the network by selecting a path based on traffic load at the opportune moment. We present a simulation study to demonstrate the significant improvements on robustness, responsiveness, and effectiveness of the proposed scheme.

1: INTRODUCTIONS

MANET (Mobile Ad hoc network) [1] has been mostly utilized in situations where infrastructure is unavailable (e.g., rescue, search and other emergency services), or where infrastructure is difficult to deploy (e.g., battlefields). The motivation of mobile ad-hoc networks is to support a robust and efficient operation in wireless networks by incorporating routing functionality into mobile nodes. Each node in the network not only acts as a sender/receiver but also as a router, forwarding data for other nodes. Thus, wireless ad-hoc networks are one of the most attractive products in case a fixed communication infrastructure, wired or wireless, does not exist or has been destroyed.

With the enhancement both in the bandwidth of wireless channels and in the computing power of mobile devices, multimedia transmission service is expected to be provided over MANET maturely. It is an instantly deployable wireless network characterized by multi-hop wireless links, the absence of any cellular infrastructure, and frequent host mobility. The topology may change frequently due to node mobility. It is highly error prone because of frequent changing topology, interference, channel fading, and the lack of infrastructure. Hence the established end-to-end connection routes between senders and receivers are possibly to be broken during multimedia transmission service, causing error interruptions, freezes or jerkiness in the received signal.

Most existing ad-hoc routing protocols build and utilize only one single route for each pair of source and destination nodes, such as Dynamic Source Routing (DSR) [2]. Due to node mobility, node failures, and the dynamic characteristics of the radio channel, links in a route may become temporarily unavailable, making the route invalid. The overhead of finding alternative routes may be high and extra delay in packet delivery may be introduced. Various multipath routing protocols address this problem by providing a strategy using more than one route to transmit data to the destination node [3][4]. Source and intermediate nodes can use these routes as primary and backup routes. Alternatively, they can distribute traffic among multiple routes to enhance transmission reliability, provide load balancing, and secure data transmission.

The multipath extension to DSR (MDSR) presented in [5] maintains alternate disjoint routes to the destination, which are used when the primary one fails. It is demonstrated by simulations that the multipath extension scheme can reduce the frequency of route request query floods. Ad hoc on-demand distance vector routing (AODV) adopts both a modified on-demand broadcast route discovery approach used in DSR and the concept of destination sequence number adopted from destination-sequenced distance vector routing [6]. Ad-hoc on-demand multipath distance vector routing (AOMDV) computes multiple loop-free and link-disjoint paths during the route discovery process [7]. Multipath source routing (MSR) proposes a weighted round-robin heuristic-based scheduling strategy to distribute load among multiple paths [8]. Another multipath extension to DSR, proposed in [9], uses node coloring techniques to find two disjoint paths during the query phase of the route discovery process. However, the load is distributed across only two routes per session.

While most multipath extensions to DSR and AODV prefer disjoint paths, new path selection criteria have recently been adopted in several multipath schemes. Load-Aware On-Demand Routing Protocol (LAOR) [10] that computes and uses the total path delay as the load metric to select a route. Load-Sensitive Routing Protocol (LSR) [11] defines the load metric of a node as the total number of packets buffered in the node interface and its neighbors. This technique does not take into account the different sizes of the buffered packets. (NTBMR) scheme builds non-disjoint paths by maintaining a two-hop neighbor table and a route cache in every node [12]. The authors of NTBMR introduced an analysis model to compute the reliability of non-disjoint and disjoint multipath routing and argued that the non-disjoint paths perform better and provide more redundancy Meshed multipath routing (M-MPR) [13] uses meshed paths and selective forwarding on all intermediate nodes to achieve better load distribution in sensor networks.

In this paper, we propose a new routing scheme with load balance to support multi-route multimedia transmission with robustness and efficiency over wireless ad hoc networks. We develop an efficient multipath routing algorithm for ad-hoc networks. Due to time constraints, it is not feasible to investigate all aspects of an ad hoc routing protocol or to design a complete protocol. The emphasis in this research is on improving the efficiency of data transmission and the throughput of an ad hoc routing protocol. This work focuses on the performance of multipath routing in the wireless ad-hoc networks.

The rest of this thesis is organized as follows. In section 2, we introduce our proposed schema called Timely Backup Source Routing protocol (TBSR). TBSR is divided into two phases including on-demand route discovery and dynamic route maintenance. The experimental results and discussion will be presented in section 4. and the conclusion is made in section 5.

2: Timely Backup Source Routing

Based on our observation on those design concepts and simulation results of prior researches about multipath on-demand routing protocol, most of these algorithms acquire multiple routes by retrieving multiple subsequent RREP messages which are originated from the same RREQ query message. Upon receiving the first RREQ packet in the final destination, the node will assume that the routing path stored in the header of RREQ is the shortest-delay primary path. After saving the primary path in the cache of the destination, it waits a time-window to receive the following multiple RREQ packets. From these received multiple RREQ packets, the destination will choose the most suitable backup route according to self-defined routing metrics such as the node disjoint level between the primary path. But a problem will occur if we simulate in wireless mobile scenario, and we explain it in the following sentences. Owing to dynamic topology, the probability that the primary route becomes broken is high. If the event of broken primary link happens, in the previous solutions, the node would use the backup route to perform the route recovery process. But the chance to use a backup route being broken is also large, because the mobility may make the backup route become invalid. Once if the source node uses the staled backup route to distribute packet transmission for load balance or to salvage packets, we will have a bigger cost of control overhead and delay caused by error notification and route rediscovery. On the whole, if we apply the previous multiple path routing algorithms in a mobile scenario, it would cause much control overhead and degrade the performance.

In order to solve this problem, we propose a Timely Backup Source Routing (TBSR) algorithm. In the mobile wireless network, we need to accommodate the communications to the constrained bandwidth and mobility, and we believe that utilizing backup route is still useful for the purpose of the load-balance and salvaging packet transmission. We have defined a routing metric, called route load, which is the largest current transmitting node load on the T-RREQ passing path. Like DSR, the set of routes are discovered on-demand in TBSR. There are two phases in TBSR：1) Route Discovery and 2) Route Maintenance. We extend DSR by modifying some detailed operations in these two mechanisms and the data structure of control packet and cache to implement our TBSR. This section describes the detailed operation of the "Timely Backup Source Routing" algorithm. The functions of TBSR include on-demand route discovery, data transmission, load balance and dynamic route maintenance.

2.1: ON-DEMAN ROUTE DISCOVERY

TBSR creates the shortest-delay primary route and the timely backup route using request-reply cycles

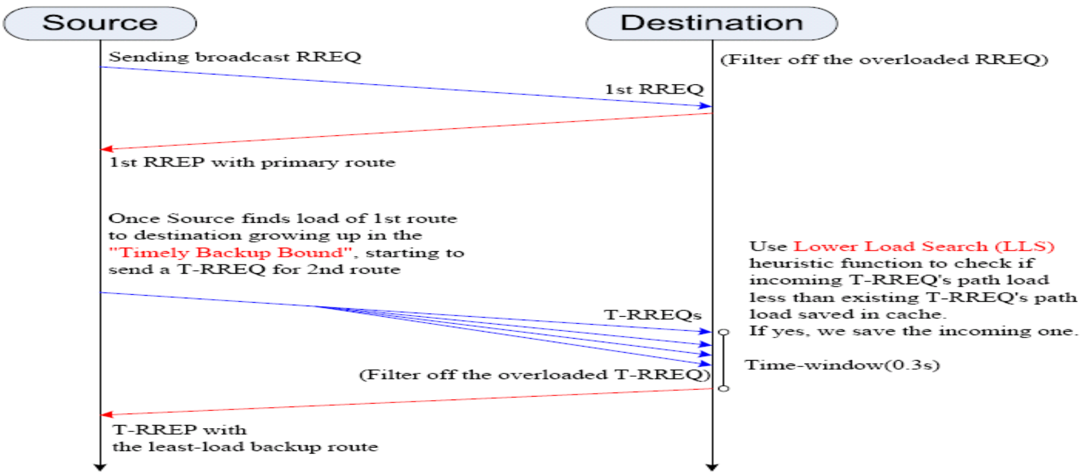

Fig. 1. Illustration of On-demand Route Discovery

based on the on-demand concept. These two routes construct a route set referring to the same destination node. Later, we will describe the process of the construction of a route set as illustrated in Fig. 1.

At first, when the source needs a route to the destination but no route information is known, it propagates the RREQ to the ad-hoc network to find the shortest-delay route. Upon receiving RREQ in the destination, we will extract the route load information from header and check if its value is over the bandwidth. If the value is larger, the destination will filter off the RREQ because using this route will cause more packet dropping. Destination assumes that the first incoming RREQ brings the shortest-delay route, so it replies to the source using this first arriving route in the reverse direction and skips the subsequent RREQ messages. Source would receive the RREP and saves its route information as the shortest-delay primary route.

Second, when the source detects that the current delivering load of the primary route entry is increasing to the level of fully loaded, it indicates that the source needs to discover another backup route to distribute the delivering load of this route set for load balance (shown in Fig. 2). Let $\lambda_p(i)$ denote the estimated transmitting load on the primary route in the ith route set. It is defined as follows：

$$\lambda_p(i) = RL_p(i) + SL_p(i) \qquad (1)$$

where $RL_p(i)$ is the queried route load of the primary route in the *i*th route set, and $SL_p(i)$ is the current transmitting node load on the primary route in the *i*th route set. We have defined a range of delivering load as the level of fully loaded, and called "Timely Backup Bound". Let β denote the bandwidth of wireless link, and α is the size of CBR packet. The range is denoted as ω, and $\omega = (\beta - 2\alpha, \beta - \alpha]$. Once $\lambda_p(i)$ enters ω, it goes to process "Timely Backup Route Search (TBRS)" algorithm described as follows：

1) The source propagates the T-RREQ
2) The destination still filters off the overloaded T-RREQ. When the destination receives the first T-RREQ, it will wait for a short time-window (0.3s) so as to receive those subsequent T-RREQ messages, so that it can select the backup route before returning the T-RREP to the source.
3) During the time-window, the destination passes the received T-RREQ to TBSR-LLS (Lower Load Search) heuristic function to get a lower-load route.

```
TBSR-LLS($\varphi_1$, $\varphi_2$) {
  if($\varphi_1$.Route_Load > $\varphi_2$.Route_Load) {
    $\varphi_1 = \varphi_2$; }
  else if($\varphi_1$.Route_Load == $\varphi_2$.Route_Load && $\varphi_1$.Route_Length > $\varphi_2$.Route_Length) {
    $\varphi_1 = \varphi_2$; }}
Note: $\varphi_1$ is the recorded T-RREQ, $\varphi_2$ is the incoming T-RREQ.
```

4) At the end of time-window, we have obtained the least-load backup route from the compared result of TBSR-LLS function. The destination replies this route to the source.
5) Once source receives the T-RREP, it inserts the timely backup route information to the corresponding route set.

Note that when the RREP or the T-RREP is forwarded by the intermediate node, the node will snoop the route in the header. First, it divides the route into two new routes from itself, one is toward the source, and the other is toward the destination. Second, it checks if any target node of route set is the same as the target node of the new route, that is, it needs to check twice. Third, if a route set is matched and it has an empty entry for the backup route, the node will put the new route into this entry. Else, if no route set is found, the node uses the new route to build a new route set and employs it as the primary route, but not for the case of T-RREP because the route in the T-RREP is not suitable to be treated as the primary route.

2.2: DATA TRANSMISSION AND LOAD BALANC

The main idea of TBSR is to distribute network load on reliable multiple routes to improve robustness and efficiency. We use the shortest-delay primary route to minimize the route acquisition latency required by on-demand protocols. However, it would induce a trend that most shortest-delay route might become congested because we want each connection has as little transmission delay as possible. To avoid the appearance of congested area in the ad-hoc network, we propose the concept of "Timely Backup Route" discussed as above. After the second time of Route Discovery phases, data packets can be delivered jointly using the primary route and the timely backup route. Let $\lambda_B(i)$ denote the estimated transmitting load on the backup route in the *i*th route set. It is calculated as follows：

$$\lambda_B(i) = RL_B(i) + SL_B(i) \qquad (2)$$

where $RL_B(i)$ is the queried route load of the backup route in the *i*th route set, and $SL_B(i)$ is the current transmitting node load on the backup route in the *i*th route set. We also propose an algorithm to be responsible for load balance.

TBSR-LB (Load Balance) Algorithm：

1) Each time the source sends a packet, it checks the following conditions.
   (C1) $\lambda_p(i) \geq \beta - \alpha$
   (C2) $\lambda_p(i) > \lambda_B(i)$
   (The definitions of $\lambda_p(i)$ and $\lambda_B(i)$ are included in the equations of Eq. (1) and Eq. (2))
2) Only if both conditions are met, the source chooses the backup route to deliver the packet.

In the first condition, we want to guarantee that the ratio of the shortest-delay primary route utilized is maximized. In the second condition, the backup route would not be chosen to deliver packets until the primary route is fully loaded and its load is less than the load of the primary route.

2.3: DYNAMIC ROUTE MAINTENAINCE

2.3.1: SALVAGING A BROKEN LINK

TBSR utilizes multiple routes to provide robustness and resilience to mobility and congestion. It is important to detect link failures and recover broken routes immediately to maintain effective routing. In our TBSR algorithm, when an intermediate node fails to forward the data packets to the next hop using its primary route, it uses "salvaging by backup" procedure to try to use its backup route to deliver the data to the destination and prolong the connection.

If there is an available backup route, the node uses it to salvage the transmission of the data packets. Even if the backup route is unavailable or broken in the corresponding route set, it will explore other route sets in its route cache for other routes to the destination. If a route exists, the intermediate node would rebuild the backup route using this route and use it to forward data packets to continue the connection. Otherwise, it will report a RERR to the source of data packet and drop the data packet. Since we need to ensure that RERR can arrive at the source safely, we employ the route recorded in the header of data packets and send a RERR in the reverse upstream direction. When the source node and the passing upstream intermediate nodes receive RERR, it will update the route set. If partial primary route is workable, we will re-allocate it to an adequate route set. Else, we will promote the backup route to the primary entry. Only the source node would re-initiate a new route discovery process if no backup route exists for the destination.

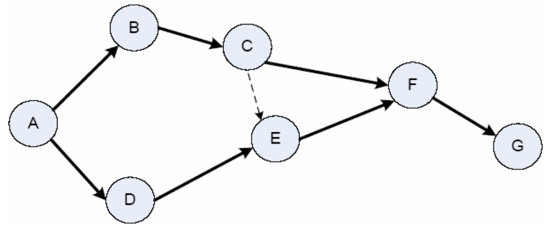

Fig 2. An example of salvaging a broken link.

In Fig. 2, when the link C→ F is broken, the node C will find the backup route C→ E→ F→ G for data transmission. If no available route exists in node C, node C will send a RERR to node A via C→ B→ A. Node A and B will update the route set.

2.3.2: UPDATING A NON-OPTIMAL ROUTE

In the mobile scenario, the primary route and timely backup route in the route cache may not be optimal due to dynamic topology. When a node snoops a data packet and finds that its address is on the unprocessed part of route in the packet's header, it will elide the extra portion to build a new route and send a Gratuitous RREP (G-RREP) with this new shorten route back to the source of the data packet. Upon receiving G-RREP, the source will only shorten the primary routes according to the route information in G-RREP because the routing metric for the backup route is the route capacity, not the delay. For example, if node E in Fig. 2 can snoop data packets directly from node A, and node E finds the route in the header is A→ B→ C→ E. Node E would shorten the route into A→ E and send a G-RREP with this new route back to node A.

2.3.3: REFRESHING THE CACHE

When the RERR arrivals the source node, the node would re-initiate a new route discovery process if the backup route also fails to transmit data. In this case, the source node propagates new RREQ piggybacked with the RERR. Once the node around the source node receives this kind of RREQ, it would remove the ineffective link from the route cache according to the information of RERR. After the refreshing cache procedure, the backup route would be promoted to the primary entry if possible.

The key improvement of TBSR is the reduction of the frequency of rediscovering backup route. TBSR has a flexible route maintenance mechanism to provide robust link connection under the mobile scenarios. We broadcast T-RREQ when source node detects the load of primary path is growing up in the Timely Backup Bound. Therefore TBSR extends DSR by selecting a timely backup route to achieve more reliable load balance between each communicating mobile pair of nodes.

3: SIMULATION

To evaluate the performance of TBSR, we used a comprehensive simulation model based on *ns-2* to compare with Dynamic Source Routing (DSR) [2], which is a well-known and rival on-demand routing protocol. Another protocol compared is BSR, which copies TBSR but eliminates the timely characteristic. The source node broadcasts one RREQ to get both primary route and backup route in BSR. *ns-2* includes CMU's Monarch Group's mobility extension for simulating the scenario of multi-hop wireless ad-hoc networks. For the DSR simulation, we used the latest available version from the VINT project with a full implementation of the update and enhanced technique.

The interface uses the IEEE 802.11 Distributed Coordination Function (DCF) as the medium access control (MAC) protocol. The 802.11 DCF uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets for "unicasting" data transmission to a neighbor node. The transmission of data packet is followed by an ACK. "Broadcast" data packets and the RTS control packets are sent using physical carrier sensing. An unslotted carrier sense multiple access (CSMA) technique with collision avoidance (CSMA/CA) is used to transmit these packets. Once link failure happens, MAC layer would send a signal to notify the routing layer. The parameter values of TBSR and DSR are summarized in Table 1.

Table 1. Parameters of routing protocols.

Parameter	TBSR	DSR
Cache Size	94	94
Interface Queue Size	50	50
Reply to Requests From the Cache	Off	On
Backup Route Reply Wait Time	0.3 sec	N/A
Bandwidth	1 Mbps	

3.1: TRAFFIC AND MOBILITY MODELS

The mobile ad-hoc network experiments were performed with 50 mobile nodes moving in two physical areas of different size. One is 1,500 meters × 300 meters, the other is 1,000 meters × 1,000 meters. The rectangular scenario is to increase the average number of hops. The smaller pause time value stands for the more mobile status. Ten runs of different traffic and mobility scenarios are averaged to generate each statistical data. The details of simulated model are summarized in Table 2

Table 2. Parameters with respect to pause time.

Area (m^2)	1500 x 300	1000 x 1000
Simulation time (s)	500	
Mobile nodes	50	
Sessions	32	
Packet type	CBR (Constant Bit Rate)	
sending rate (packets/s)	5	
Packet size (bytes)	1024 / 512	
Node max. speed (m/s)	20 (uniform distribution)	
Pause time (s)	0, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500	

3.2: RESULTS

We used three key performance metrics to evaluate TBSR and DSR as follows：

1) *Packet delivery ratio* ***(PDS)***：the percentage of data packets delivered to their destination nodes of those originated by the sources..
2) *Average end-to-end delay of the data packets* ***(AVD)***：the interval from the time a packet is sent from the source until the time it is received at the destination.
3) *Control message overhead ratio(**CMO**)*：the ratio of routing control packets transmitted to data packets delivered.

The performance results in this simulation have shown that TBSR provides a significant improvement for each key metric. Table 4 (a) and Table 4 (b) show the simulation results of 1024 bytes CBR packet scenario.

Table 4. Results of 1024 bytes CBR packet scenario

Area Size *(m^2)*	**1500 x 300**			**1000 x 1000**		
	TBSR	**BSR**	**DSR**	**TBSR**	**BSR**	**DSR**
PDS (%)	12.6	12.5	11.1	7.74	6.38	5.12
AVS (s)	3.09	3.19	3.43	3.06	3.22	3.2
CMO (%)	5.72	5.76	8.65	7.8	8.39	13.6

Table 5. Results of 512 bytes CBR packet scenario

Area Size *(m^2)*	**1500 x 300**			**1000 x 1000**		
	TBSR	**BSR**	**DSR**	**TBSR**	**BSR**	**DSR**
PDS (%)	19.29	18.43	18.5	9.21	8.3	8.63
AVS (s)	2.46	2.49	2.5	2.93	3.3	3.08
CMO (%)	8.54	8	13.5	11.15	11.6	21.4

TBSR outperforms DSR and BSR, and TBSR generally has significant improvement in each mobility rate. These results are due to fact that TBSR attempted to utilize the timely backup route when two conditions happen：1. distributing the traffic when the primary route is going to be congested. 2. packet recovery in the presence of the primary link failure. In the former case, TBSR can enhance the utilization of bandwidth and reduce the congested regions. Although the backup route may costs more delay slightly, it has the largest available bandwidth to share the load. To fit the requirement of on-demand time constraint, we have used the stringent reply waiting time-window (0.3 s). In the latter case, TBSR can salvage more packets and reduce more control overhead because the intermediate nodes only send RERR back to the source when both the primary route and the backup route break. In the higher mobility, the improvement is more prominent because TBSR discovers a backup route in a timely manner and can utilize the usable backup route to salvage packets and to continue the data session.

TBSR also surpasses DSR and BSR, and the improvement is especially outstanding when the mobile nodes are moving faster. As the mobility increases, TBSR will have a more reliable data session and less end-to-end delay because it can use the backup route to keep on packet transmission and reduce the probability of yielding a much longer delay in route re-discoveries.

As expected, TBSR also outperformed DSR and BSR in the control message overhead ratio. The reason is that more route re-discovery is invoked by DSR through its salvaging procedure because DSR has no backup route to use. In the more stable condition, TBSR would broadcast slightly more routing control messages to construct the backup route for preventing the occurrence of congested areas.

In all cases of mobile scenarios, TBSR outperforms BSR and DSR. This achieves our goal and confirms that the intuition of utilizing backup route is advantageous for applications transmitting CBR data flow in the mobile wireless ad-hoc networks. The system consumes less battery power in the network device hardware because TBSR reduces the amount of control message overhead.

4: CONCLUSION

Timely Backup Source Routing (TBSR) is a novel on-demand routing protocol which explores the least-load backup route in a timely manner in the wireless ad-hoc network. Through simulation, it is proved that TBSR considerably gains better performance in packet delivery ratio, average delay and control overhead compared with BSR and DSR. In order to implement this idea practically, we propose an estimation algorithm to measure the rout load metric approximately and to select the least-load route by a heuristic function. Measuring the traffic in bytes makes the measurement of route load more accurately. We validate this algorithm via simulation under diverse mobile scenarios.

More sophisticated Timely Backup Route Search (TBRS) algorithm can be deployed to improve the performance. We can combine multiple types of packets in the simulation of the mobile scenario to see how the performance changes. These are interesting and open research issues that are worth further investigating in the future.

REFERENCES

[1] J. Macker and M. S. Corson, “Mobile Ad Hoc Networks (MANET): Routing Protocol Performance Issues and Evaluation Considerations”, http://www.ietf.org/rfc/rfc2501.txt, Jan.1999.

[2] D. B. Johnson, D. A. Maltz, and Y.-C. Hu “The dynamic source routing protocol for mobile Ad Hoc networks”, IETF Internet Draft (http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt), July. 2004.

[3] Song Guo; Yang, O.; Yantai Shu; “Improving source routing reliability in mobile ad hoc networks”, Parallel and Distributed Systems, IEEE Transactions on vol. 16, no. 4, pp. 362 - 373, Apr. 2005.

[4] Shiwen Mao; Shunan Lin; Panwar, S.S.; Yao Wang; Celebi, E.; “Video transport over ad hoc networks: multistream coding with multipath transport”, Selected Areas in Communications, IEEE Journal on vol. 21, no. 10, pp. 1721 – 1737, Dec. 2003.

[5] A. Nasipuri, R. Castañeda, and S. R. Das, “Performance of multipath routing for on-demand protocols in mobile ad hoc networks,” Mobile Networks and Applications, vol. 6, no. 4, pp.339–349, Aug. 2001.

[6] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir Das. "Ad Hoc On Demand Distance Vector (AODV) Routing." IETF RFC 3561.

[7] M. K. Marina and S. R. Das, “Ad hoc on-demand multipath distance vector routing,” ACM SIGMOBILE Mobile Computing and Communications Review, vol. 6, no. 3, pp. 92–93, July 2002.

[8] L.Wang, Y. Shu,M. Dong, L. Zhang, and O.W.W. Yang, “Adaptive multipath source routing in ad hoc networks,” in Proceedings of IEEE International Conference on Communications (ICC), vol. 3, pp. 867–871, 2001.

[9] J. Wu, “An extended dynamic source routing scheme in ad hoc wireless networks,” in Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS) , pp. 3832–3838, Jan. 2002.

[10] J-H. Song, V. Wong and V. Leung, “Load-Aware On-Demand Routing (LAOR) Protcol for Mobile Ad hoc Networks”, in Proceedings of IEEE Vehicular Technology Conference (VTC-Spring), Jeju, Korea, Apr. 2003.

[11] K. Wu and J. Harms, “Load-Sensitive Routing for Mobile Ad Hoc Networks”, Proceedings of IEEE ICCCN’01, Scottsdale, AZ, Oct. 2001.

[12] Z. Yao, Z. Ma, and Z. Cao, “A multipath routing scheme combating with frequent topology changes in wireless ad hoc networks,” in Proceedings of International Conference on Communication Technology Proceedings (ICCT), vol. 2, pp. 1250–1253, Apr. 2003.

[13] S. Kim, W. Noh, and S. An, “Multi-path ad hoc routing considering path redundancy,” in Proceedings of Eighth IEEE International Symposium on Computers and Communication (ISCC) , pp. 45–50, Mar. 2003.