Associativity-Based On-Demand Multi-Path Routing In Mobile Ad Hoc Networks

Shafqat Ur Rehman[1], Wang-Cheol Song[2] and Gyung-Leen Park[3]
[1] INRIA, Sophia Antipolis, France
[e-mail: shafqat.rehman@gmail.com]
[2] The Department of Computer Engineering, Jeju National University, Jeju, Korea
[e-mail: philo@jejunu.ac.kr]
[3] The Department of Computer Science and Statistics, Jeju National University, Jeju, Korea
[e-mail: glpark@cheju.ac.kr]
*Corresponding author: Wang-Cheol Song

Received August 5, 2009; revised September 13, 2009; accepted September 20, 2009; published October 30, 2009

Abstract

This paper is primarily concerned with multi-path routing in Mobile Ad hoc Networks (MANETs). We propose a novel associativity-based on-demand source routing protocol for MANETs that attempts to establish relatively stable path(s) between the source and the destination. We introduce a new notion for gauging the temporal and spatial stability of nodes, and hence the paths interconnecting them. The proposed protocol is compared with other unipath (DSDV and AODV) and multi-path (AOMDV) routing protocols. We investigate the performance in terms of throughput, normalized routing overhead, packet delivery ratio etc. All on-demand protocols show good performance in mobile environments with less traffic overhead compared to proactive approaches, but they are prone to longer end-to-end delays due to route discovery and maintenance.

Keywords: Associativity, multi-path routing, node-disjoint, mobile ad hoc networks

This research was supported by the MKE (The Ministry of Knowledge Economy), the Korean government, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency ((NIPA-2009-C1090-0902-0040)).

DOI: 10.3837/tiis.2009.05.004

1. Introduction

Mobile ad hoc networks (MANETs) have gained worldwide popularity as a result of the continous reduction of the sizes of personal computing devices, proliferation of those devices and with the advancements made in wireless communication technologies. Wireless networks are currently popular because of their "3 Anys"–Any person, anywhere and anytime. MANETs are self organizing and self configuring multi-hop networks wherein nodes act co-operatively to establish the network "on-the-fly". MANETs bear great application potential where wired infrastructure is not viable and where temporary wireless networks are needed for instant communication such as disaster and emergency situations, battlefield communications, mobile conferencing, law enforcement operations and so on [1].

Routing in MANETs has received tremendous interest from the networking research community [2]. Routing protocols can be classified as either unipath or multipath based on the number of routes between the source and destination. Intuitively, multipath routing can better utilize network resources and it can offer performance improvements over unipath routing. Multipath routing is also more promising for QoS provisioning in ad hoc networks. The reason is multipath routing can provide load-balancing, fault-tolerance and higher throughput.

In this paper, we propose a multipath routing protocol that builds multiple nod-disjoint paths between the source and the destination. The path discovery process is initiated by the source. The protocol derives its motivation from on-demand source routing, multipath routing and associativity-based routing. We modify DSR [3] to incorporate periodic beacons for associativity measurements, incorporate associativity metrics in the path discovery process and morph its route maintenance according to our mechanism.

On-demand routing protocols are well suited for large, random and dynamic multihop ad hoc networks because they only maintain the active routes. Dynamic Source Routing (DSR) and Ad hoc On-demand Distance Vector (AODV) [4] are two of the most popular on-demand routing protocols for ad hoc networks. However, both are designed to discover and maintain only a single path between a pair of end nodes at any given time. Thus, when route failures occur, the latency of route repair can disrupt communication for an extended period of time [5].

Multipath routing can be broadly classified into two categories: node-disjoint and link-disjoint. Both approaches to multipath routing provide improved fault-tolerance, load balancing, throughput and QoS provisioning. Contemporary research shows that using multipath routing in high-density ad hoc networks results in better throughput than using unipath routing [6]. Node-disjoint multipath routing provides additional benefits in terms of enhanced fault-tolerance and congestion control, and the enhanced capability for load balancing [5]. In this paper, we focus on node-disjoint paths, which are particularly beneficial in ad hoc networks for public safety and emergency applications.

The pioneering work on associativity-based routing in MANETs was done by C.K. Toh by proposing the Associativity-based Routing Protocol (ABR) [7]. ABR is a beacon-based on-demand source routing protocol. Each node broadcasts periodic Hello messages to signify its presence to its neighbors. These beacons are used to update the associativity table of each node. With the temporal stability and the associativity table, nodes are able to classify each neighbor link as being stable or unstable [8]. By selecting the nodes with high associativity counts/ticks, the route is expected to be long-lived. This may not result in the shortest path, but

route failures are less, and hence route maintenance overhead is lower due to the relative stability of the paths. The protocol offers better performance as compared to DSR in terms of throughput and end-to-end delays, but DSR beats ABR in terms of storage overhead by a small fraction and in terms of simplicity [9].

We propose an on-demand multipath source routing algorithm that guarantees the discovery of node-disjoint paths in wireless multihop ad hoc networks [5]. The constituent nodes are decided on the basis of their relative temporal stability. Please, note that the concept of associativity employed in our protocol is distinct from the one employed in ABR. The Associativity-based routing protocol that's proposed here decides the path(s) between the source and the destination based on the spatial and temporal stability of the nodes. The calculated temporal and spatial stability are not based on the usual tic-count approach, instead the stability of a node is based on the node's temporal connectivity, mobility of the node's 1-hop neighborhood and its health. We expect this approach to find optimal paths and we expect it to help reduce route maintenance cost.

The rest of the paper is organized as follows: Section 2 proposes an associativity-based on-demand multipath source routing protocol for mobile ad hoc networks. Section 3 shows the analysis and comparison of the proposed protocol with the other routing mechanisms. Section 4 presents the summary of the paper, draws the conclusion of the research and sheds light on possible enhancements and our future work.

2. Associativity-based Dynamic Source Routing in MANETs

In this section, we propose a distributed on-demand multi-path routing protocol that's called Associativity-based Dynamic Source Routing (ADSR) for MANETs. ADSR uses the temporal information of nodes to calculate the fitness of the candidate paths. Routes are selected based on the relative stability of the intermediate nodes. A node is classified as stable based on its temporal associativity and health. Associativity is determined by time-averaged nodal connectivity and nodal mobility; and health is the combination of the number of factors like residual battery life, signal stability, buffer occupancy rate, storage capacity, processing power and etc. However, in this paper, we consider only the residual battery power for calculating nodal health. The link corresponding to a stable neighbor is considered to be a stable link, while a link to an unstable neighbor is called an unstable link. A path that has comparatively more stable nodes is considered to be optimal in terms of stability.

2.1 Neighbor Discovery

Each node periodically relays Hello messages to its 1-hop neighbors in order to make its presence known to them. Each node maintains a neighbor history table that is used to keep track of the neighbor nodes that were discovered through Hello messages. The interval of the periodic Hello broadcasts is configured during network setup. The node keeps a record of the neighbor nodes that were discovered only during the last h periodic intervals. This set of records is used to calculate the temporal connectivity and the degree of movement in the 1-hop neighborhood of a node. This measurement is referred to as *nodal associativity*. *Nodal associativity* together with *nodal health* determines the overall stability of a node. This stability is termed as the *nodal weight* whose calculation is elaborated in the next section.

2.2 Nodal Weight Calculation

Each node, after h NA broadcasts, calculates its weight based on the associativity index and the residual power capacity.

$$W_v = |(P_b \times \rho) + I_v \times (1 - \rho)|, \rho \in \mathbb{R} : 0 \leq \rho \leq 1 \quad (1)$$

Here, ρ represents the level of weightage assigned to P_b or I_v. P_b is the residual battery life and I_v is the associativity index of the node. I_v is calculated as:

$$I_v = N_c - N_{topo} \quad (2)$$

In equation (2), N_c is the average nodal connectivity of a node and N_{topo} is the estimation of the relative mobility of nodes in the 1-hop neighbor of a node. Both of these measurements are explained in the next section.

2.3 Calculation of Associativity Index

Our calculation of I_v is based on both the connectivity and the relative mobility of the nodes. Connectivity of a node is measured by the periodic exchange of NA messages. Each node maintains a topology cache of size h. h is a pre-defined constant. After h broadcast of NA messages, the average nodal connectivity is calculated as follows:

$$N_c = \frac{\sum_{i=0}^{h-1} c_i |X_i|}{h} \quad (3)$$

X_i is the set of neighbors figured out by ith NA broadcast. c_i is the weightage assigned to ith neighbor set and is decided such that $c_0 + c_1 + \cdots + c_{h-1} = 1$ and $c_0 < c_1 < \cdots < c_{h-1}$. Using arithmetic series [10], c_0 is calculated as $\frac{h}{2}(2c_0 + (h-1)d_c) = 1$. We assume that common difference $d_c < \frac{1}{2h}$. Successive weight terms are found by the arithmetic sequence $c_i = c_0 + (i)d_c$. By summing over the cardinalities of h X_i's and dividing them by h, we get the average connectivity of the node. Also we measure the change in the 1-hop neighbors of a node that gives a good estimation of the relative mobility of the neighboring nodes. Again, we utilize the h -sized local cache of neighbor sets:

$$N_{topo} = \frac{\sum_{i=0}^{h-1} w_i |(X_i \cup X_{i-1}) - (X_i \cap X_{i-1})|}{h-1} \quad (4)$$

Here, $w_1 + w_2 + \cdots + w_{h-1} = 1$ such that $w_1 < w_2 < \cdots < w_{h-1} = 1$. w_1 is calculated as $\frac{h-1}{2}(2w_1 + (h-2)d_t) = 1$ [10]. It is recommended that common difference $d_t < \frac{1}{2(h-1)}$. Using arithmetic series successive weight terms that are found by $w_i = w_1 + (i-1)d_t$.

We combine equations (3) and (4) to get the final associativity index for a node. Therefore, from equation (2), we have

$$I_v = \frac{-(\sum_{i=1}^{h-1} w_i |(X_i \cup X_{i-1}) - (X_i \cap X_{i-1})|)}{h-1} + \frac{\sum_{i=1}^{h-1} c_i |X_i|}{h} \quad (5)$$

Suppose 1-hop neighbor sets of a node discovered against 3 NA broadcasts are $X_i = \{1,2,4,5,6\}$, $X_{ii} = \{1,2,4,10\}$ and $X_{iii} = \{4,6,7,10,11,12,14\}$, as is shown in **Fig. 1**.

Let $d_c = 0.16 < \frac{1}{2h}$, then $c_0 = 0.17$, $c_1 = 0.33$ and $c_2 = 0.49$. Therefore, $N_c = \frac{c_i|X_i|+c_{ii}|X_{ii}|+c_{iii}|X_{iii}|}{3} = \frac{0.17(5)+0.33(4)+0.49(7)}{3} = 1.87$.

For N_{topo}, assume $d_t = 0.25 < \frac{1}{2(h-1)}$, then $w_i = 0.375$ and $w_{ii} = 0.625$. From equation (4), we get $N_{topo} = \frac{0.375(3)+0.625(7)}{2} = 2.75$. Hence, $I_v = |1.87 - 2.75| = 0.88$.

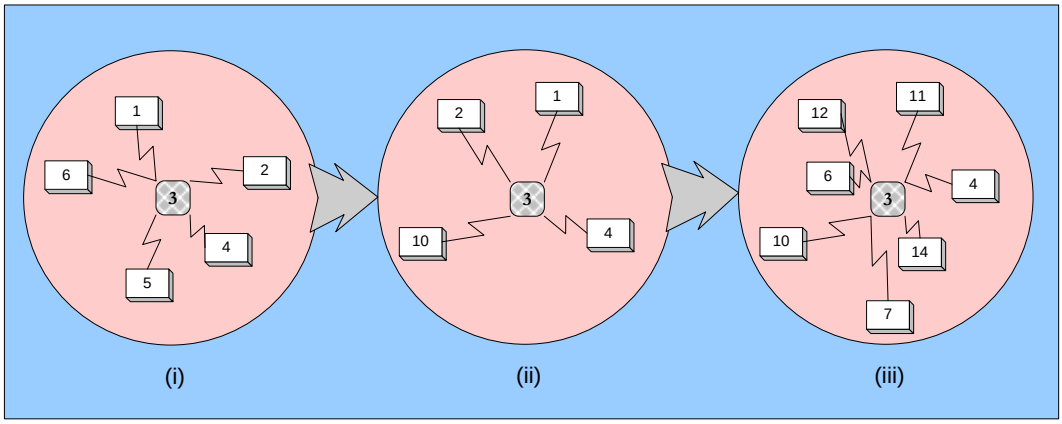

Fig. 1. The gradual change in 1-hop neighbors of node 3

2.4 Route Discovery

ADSR is an on-demand source routing protocol wherein routes are built through RREQ/RREP cycles. The source node broadcasts a Route Request (RREQ) if it does not already have a route to the destination. The destination node receives multiple RREQ packets through multiple routes. The destination node then determines the most suitable node-disjoint paths according to the fitness function and sends a Route Reply (RREP) to the destination through the specified routes.

2.4.1 Route Request (RREQ) Propagation

When the source node does not know the route to the destination, it floods the network with Route Requests (RREQs). A RREQ carries along two metrics. The first metric is the source path in which a record of the sequence of hops that were visited by the RREQ packet is accumulated, as it is propagated through the network. Each subsequent node appends its address to the source path as RREQ coninues to travel towards the destination. Path is also pre-appended with hop count, which is the number of legs that the RREQ packet has traversed between the source and the current node. The second metric is the weight of a node that represents the node's relative stability. Each intermediate node adds its weight to the path weight. When a RREQ packet arrives at the destination, it contains the summation of all the nodal weights along the traversed path. We modify the DSR Route Request (RREQ) packet in order to incorporate the nodal weight.

The aim of the algorithm is to construct on-demand multiple node-disjoint paths. In order to achieve this purpose, the destination node must know all the alternating paths so that it can

choose the optimal paths. As we are using source routing, the route is included in the RREQ packets. Additionally, Route Replies (RREPs) from the intermediate nodes are disabled. This facilitates the centralized decision making about the selection of the optimal routes at the destination. A RREQ packet also contains the source ID (set by the source), the destination or target host's ID and the sequence number that uniquely identifies the RREQ packet. When a node receives a duplicate packet, it calculates the fitness of the partial path that's traversed by the packet so far. The hop count and nodal weight are the two parameters used by the fitness function. If the path fitness is greater than the previous partial paths via this node, then the duplicate packet is allowed to be forwarded.

The destination node receives multiple packets through multiple paths during a pre-defined time window. The destination node calculates the fitness of each path and selects at least two optimal routes that are the best fit according to the fitness function.

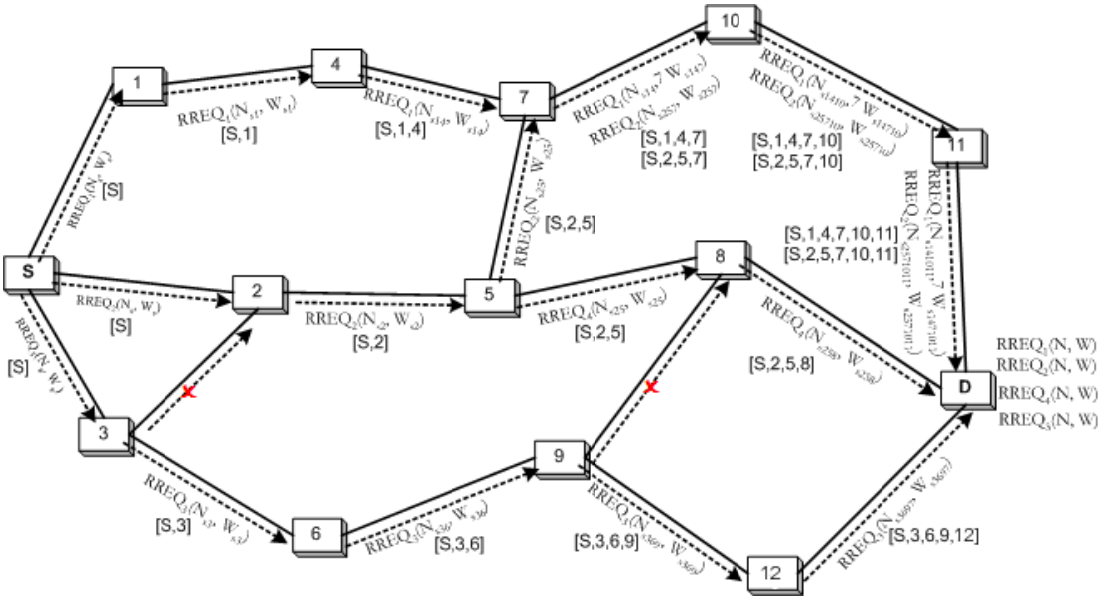

Fig. 2. Route Request (RREQ) propagation

Fig. 2 explains a typical propagation of route request (RREQ) through an example network. The request that's broadcasted by source node S is received by three nodes identified as 1, 2 and 3. The request packets traveling through different paths are identified as $RREQ_1$, $RREQ_2$, $RREQ_3$,… $RREQ_N$. The propagation of duplicate packets is allowed under certain conditions. Node 3 broadcasts the request to its neighbor nodes 2 and 6. Since node 6 recieves a new packet, it updates the hop count, source path and weight fields of the packet, and it then simply forwards the packets by broadcasting. Since node 2 recieves a duplicate packet, it has to decide whether or not this duplicate packet is fit enough to be allowed to further propagate. It compares the fitness of route [S, 3, 2] with that of route [S, 2]. The request packet is discarded because its fitness is not greater than the previous route's (in this case [S, 2]). Similar is the case of the route request packet that's broadcasted by node 9. Node 12 allows the packet to be broadcasted further, but node 8 discards it for not being sufficiently fit. Now consider another case, where a duplicate packet is allowed to be transmitted by a node. Node 7 receives a duplicate RREQ packet from node 5. Node 7 allows the duplicate RREQ packet to be forwarded through it because this packet's fitness is greater than that of the previous routes. As

is obvious from figure 2, only four RREQ packets are able to make it to the destination. They are identified as $RREQ_1$, $RREQ_2$, $RREQ_3$ and $RREQ_4$. We are interested in the most optimal two node-disjoint routes. The algorithm can easily be generalized for selecting only one best route or more than two optimal routes. This means that the algorithm can be easily adapted to unipath and multipath scenarios.

2.4.2 Route Selection

In our algorithm, the destination selects two optimal node-disjoint routes. Which routes are the best, is decided by evaluating the fitness function:

$$W_p \times \rho + N_p \times (1 - \rho) = P_p, where \ \ \rho \in \mathbb{R} : 0 \leq \rho \leq 1 \tag{6}$$

The fitness function takes the hop count and route weight as parameters. The effect of both of these parameters on the path fitness can be controlled by ρ. ρ simply provides a way to assign priority to one of the two parameters. The solution of the function is the measure of the route fitness. Two routes that have higher fitness value than the rest are selected and transmitted back to the source on their respective reverse paths. In order to reduce the route acquisition latency, a variation of the above route selection mechanism is also proposed. We can choose the route corresponding to the RREQ packet that arrived first at the destination as the first route. This route is the shortest delay route. The destination immediately sends the route to the source using a Route Reply (RREP) packet without waiting for the time window to expire. The fitness function is used to measure the fitness of the remaining routes received during the time window. The route that has the highest fitness value and is node-disjoint with the shortest delay route is selected and transmitted to the source using RREP. Both of the above mentioned options are further explained in the upcoming sections.

In **Fig. 2**, destination node D receives four route request packets during the time window. P_1 = {S, 1, 4, 7, 10, 11, D}, P_2 = [S, 2, 5, 7, 10, 11, D], P_3 = {S, 3, 6, 9, 12, D} and P_4 = {S, 2, 5, 8, D} are the candidate paths that the destination has to decide between to select the top two. Given below are the sample fitness values for the above mentioned paths:

S, 2, 5, 8, D	P_4	0.8
S, 3, 6, 9, 12, D	P_3	0.7
S, 2, 5, 7, 10, 11, D	P_2	0.6
S, 1, 4, 7, 10, 11, D	P_1	0.3

We consider the following two cases:

2.4.2.1 Case 1

The destination node delays the route selection until the time window expires. The received routes are then evaluated using the fitness function. Two routes that are most optimal among the candidate routes according to their fitness values are selected as the final source routes. These routes are transmitted back to the source using the their reverse. According to the fitness values that are calculated above for the candidate paths in the example network, P_3 and P_4 are the best. In this case source node can't start the transmission of data packets until it receives both the routes that are approved by the destination. The source can either delay the transmission in which case the buffer capacity is of greater importance. Normally, nodes in an ad hoc network have limited memory capacity. Hence, increased memory consumption

resulting from the route acquisition delay may not be feasible. We, therefore, propose a little modification to the route selection process that's explained in Case 2.

2.4.2.2 Case 2

The first route selected is the shortest delay route. This is the route taken by the RREQ packet that arrives at the destination first. The route is sent back to the source using the Route Reply (RREP) packet via reverse route. This minimizes the route acquisition delay. The destination node waits for a certain period of time that's also referred to as the time window, in order to learn all the remaining possible candidate paths. The destination node then selects the route that has the highest fitness value according to the fitness function and that's node-disjoint with the shortest delay route.

In **Fig. 2**, P_4 = {S, 2, 5, 8, D} is determined to be the shortest delay route so it is sent back without the need to wait for the time window to expire and for the selection process to begin. The second route is then selected after the expiration of the time window and is selected such that it meets two conditions of being both the fittest and node-disjoint with the first selected route. In **Fig. 2**, P_3 has the best fitness value among the remaining candidate paths, and it is node-disjoint as well. P_3 is, therefore, selected as the second route and is sent to the source using RREP packet on the reverse path.

2.4.2.3 Conflict Resolution

If there are more equally fit routes than one, then the one with the highest weight is chosen. If the routes happen to have the same weight, then the hop-count is considered. If the routes still can't be selected on the basis of these two conditions, then their respective arrival times at the destination are taken in to account and the one with earliest arrival time assumes priority.

2.4.3 Route Reply (RREQ)

When the destination selects a path and it needs to send it to the source, it prepares a RREP packet. Node IDs of the entire path are recorded in the RREP packet. Intermediate nodes use this information to forward the packet towards the source. Back-propagation of RREP is demonstrated in **Fig. 3**. When the source node receives a RREP packet, it retrieves the embedded route from this packet and stores it locally. The source node now has sufficient information about the destination to start transmitting data packets.

Our protocol employs source routing in which the entire route is stored in data packets resulting in an increase in the packets' sizes. Because the route replies that are from intermediate nodes are not allowed, only the source node keeps track of the routing information to the destination. This puts less demand on memory requirements at the intermediate node.

2.5 Route Maintenance

Wireless links are more error prone. They can be broken due to mobility, congestion and packet collisions at any time. This renders the route unfit for carrying the data to the destination. For effective and reliable routing, it is imperative to recover or rediscover the broken route. When a node fails to deliver the data to the next hop of the route, the route is considered to be broken and the node sends a Route Error (RERR) packet to its immediate upstream node that forwards it towards the source.

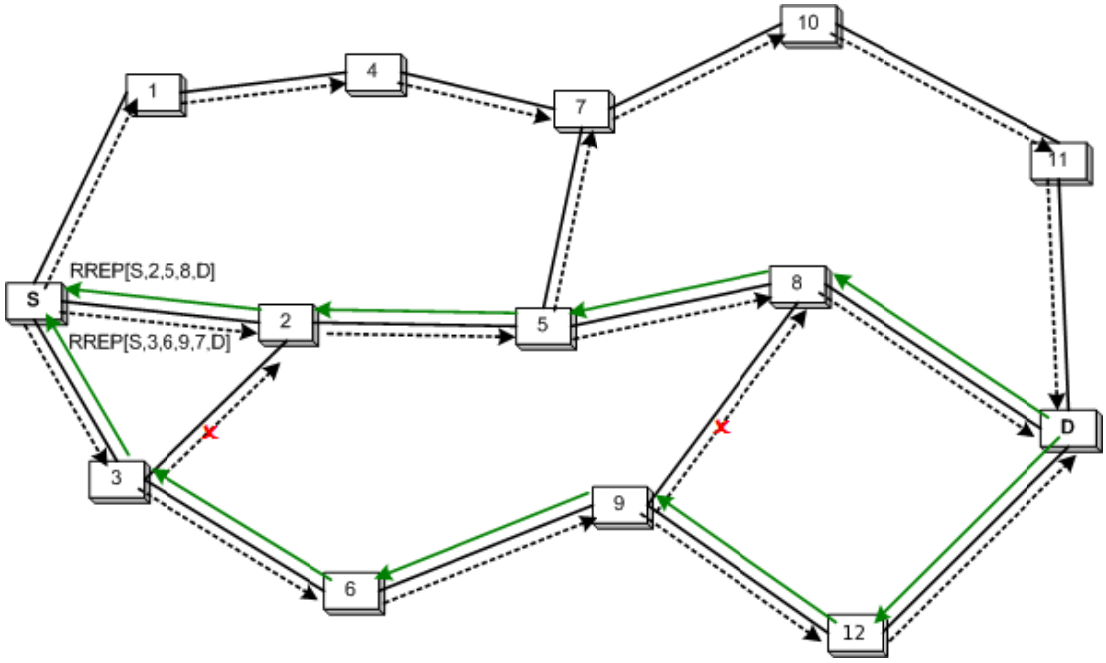

Fig. 3. Route Reply (RREP) propagation

2.5.1 Route Failure Detection

The following mechanisms are used to identify link failure on an active route:-

2.5.1.1 Hello Messages

We employ periodic Hello messages for neighbor acquisition. Periodic Hello messages can also fullfil the purpose of link failure detection. Each node maintains a neighbor table for keeping track of its immediate neighbors. If an intermediate node on an active route does not receive a Hello message from its downstream node during h time intervals (as explained in section 2.3), then it assumes the link with the upstream node to be broken and sends the route failure notification to the source node in the form of RERR packet. Two approaches are suggested to convey RERR messages to the source.

- When forwarding a data packet, a node checks the status of the next hop in the neighbor table. If the next hop is present in the neighbor table, then this means that the link connection between the two nodes is intact; otherwise, the link has failed and the source needs to be informed about the link failure.
- The intermediate node only logs the link failure, and when it receives a destination-bound message, it sends the route error (RERR) message using the embedded route info in the data packet.

2.5.1.2 Passive Acknowledgements

The previous node overhears the radio transmission of a data packet by an intermediate node and uses it as an acknowledgement. The previous node then sends RERR message to the source using the reverse of the route that's embedded in the data packet that was overheard.

2.5.1.3 Link Layer Feedback from the IEEE 802.11

The IEEE 802.11 DCF employs a contention avoidance mechanism wherein a node sends RTS (Ready To Send) messages and forwards the data frame only if it receives CTS (Clear To Send) from the next hop. After seven failed RTS retransmissions, the link is considered to have failed. The MAC layer conveys the feedback to the routing layer where it is taken as route failure and the route maintenance procedure is invoked.

2.5.2 Route Recovery and Reconstruction

The Route Error (RERR) message contains the source route and upstream and downstream nodes of the broken link. Upon receiving the RERR message, the source invalidates the route containing the broken link. If the remaining route is still valid, then the data can be immediately rerouted to this route without incurring any route rediscovery latency. If the failed route is not in use by an active session, then there is no need to rediscover the route. In the case the route is servicing an active session, it is intuitive to reconstruct the failed route. For the time during which a new route is being discovered, the whole traffic is handled by the existing valid route. If the source still has data to transmit but both the routes have been invalidated, then the source either drops or buffers the packets while route acquisition is in process. In a nutshell, we allow the reconstruction of route(s) only when the data session is active, i.e., if there is no demand for data transmission, then no route reconstruction is warranted.

2.6 Bandwidth Allocation Granularity

The source node immediately starts transmitting the data packets when it receives the first RREP. When it receives the second RREP, it will have two routes available. Efficient utilization of both routes is desirable. There are two well known bandwidth allocation schemes, namely per-connection allocation and per-packet allocation. With the per-connection allocation, it is difficult to ensure even distribution of traffic over multiple paths. The Per-packet allocation scheme has proved to be more graceful when route failures take place though it results in out-of-order packet delivery and the destination is burdened with the task of re-sequencing. However, there are efficient schemes available for re-sequencing. We employ a simple per-packet allocation scheme in which the packets are alternatingly routed on to the two paths. In case a route gets disconnected, the allocation scheme is disabled and the packets use the still valid path. Whenever multiple paths are available to the source node, the per-packet allocation scheme kicks in, and the load is almost evenly distributed across the two paths.

3. Results and Interpretations

3.1 Simulation Environment

In order to demonstrate the effectiveness of ADSR that's proposed in this paper, we evaluate the ADSR protocol and compare its performance with those of well-known reactive, proactive and multipath routing protocols. At the moment, we have performed comparative analysis with AODV, DSDV and a reactive multipath routing protocol AOMDV [11]. We have considered throughput, normalized routing overhead and packet delivery ratio as performance metrics. Measurements on the basis of some other metrics, e.g., jitter, end-to-end delay etc are also under consideration.

We have implemented ADSR using the ns-2.33 simulator (USC ISI). The topology was generated using ***setdest*** (USC ISI) utility of NS2. Three mobility scenarios were considered.

The measurements for all the protocols were taken against three mobility levels, i.e., 5m/s, 10m/s and 20 m/s. We employed both TCP and CBR traffic sources. CBRGEN.TCL (USC ISI) was used to create 10 random connections for both CBR and TCP traffic models. In the case of CBR, 7 traffic sources remain active throughout the simulation run. In the case of TCP, 6 traffic sources generate TCP traffic according to the default TCP settings in NS2. Mobile nodes are employed in an area of 670m* 670m. Each node is reachable from every other node on the network throughout the simulation run.

As explained in section 2, the Hello messages are employed to discover the temporal associativity of mobile nodes. All the nodes are pre-configured with this parameter. In our simulations, we use a Hello interval of 1000 ms. Network administrators can choose a value for it according to the network conditions and their past experience. For priority attribute ρ in the path fitness function, we use a value of 0.5 that means both hop count and path weight, assuming equal priority during the simulation analysis.

The environment settings are explained in **table 1** below.

Table1. The NS2 environment settings

Antenna type	Omnidirectional
Propagation model	TwoRayGround
Transmission range	250m
MAC protocol	802.11 with RTS/CTS
MAC bandwidth	1 Mbit
Interface queue type	CMUPriQueue for ADSR Drop-tail priority queue for the rest
Max. IFQ length	50
Propagation delay	1 ms
Node count	50
Network size	670m × 670m
Simulation time	1000s

3.2 Packet Delivery Ratio

Packet delivery ratio is achieved by dividing the total number of packets that are sent by the total number of packets that are received. This provides a good measure of the reliability and robustness of the routing protocol. **Fig. 4** and **Fig. 5** compare the packet delivery ratios of ADSR, AOMDV, AODV and DSDV under three different mobility scenarios. **Fig. 5** demonstrates the case wherein all traffic sources are CBR, and **Fig. 4** shows the case wherein traffic sources are TCP. ADSR is more fault-tolerant and stable as it achieves greater packet delivery ratios.

3.3 Throughput

Throughput is the amount of data transferred from one place to another in a specified amount of time. We measure the throughput of the protocol under consideration of time granularity of 2000 ms. Three different mobility scenarios, namely, 5m/s, 10m/s and 15m/s are taken into consideration, and each scenario is analyzed under several TCP and CBR traffic sources.

3.3.1 The throughput of CBR Traffic

In **Fig. 6**, **Fig. 7** and **Fig. 8**, we compare the throughput of ADSR with those of AODV, DSDV and AOMDV under CBR traffic. Overall, ADSR is shown to have achieved better throughput

under the aforementioned conditions. Especially, as the maxium speed increases, we can see that ADSR shows more stable throughput than other protocols.

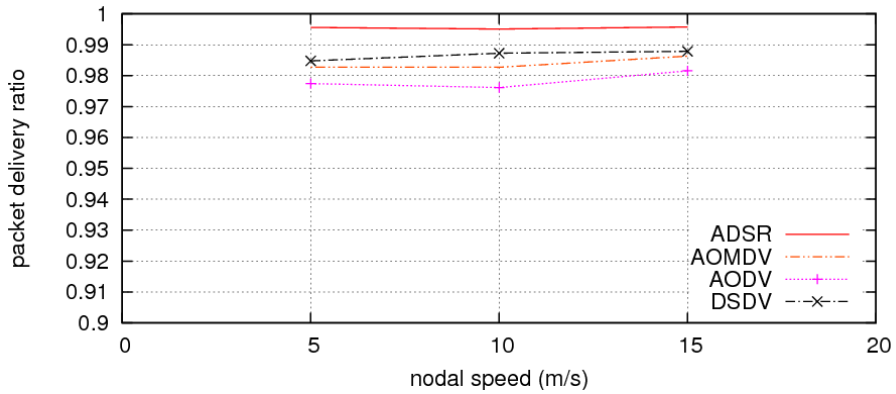

Fig. 4. Packet delivery ratio of TCP traffic against three mobility levels, namely, 5m/s, 10m/s and 15m/s

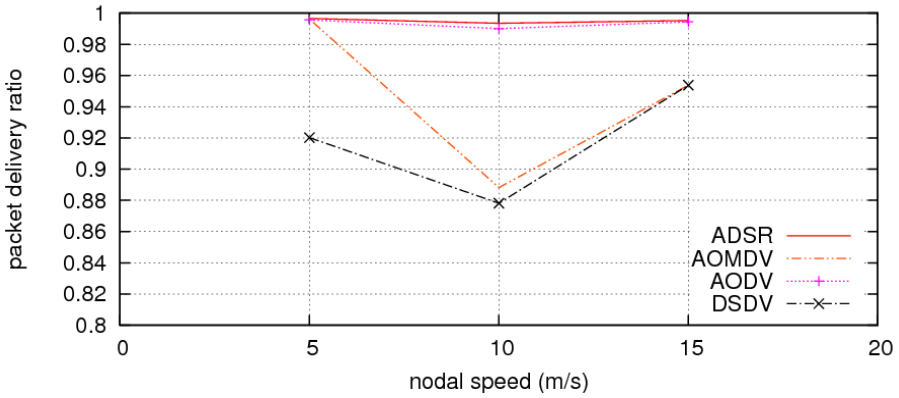

Fig. 5. Packet delivery ratio for CBR traffic against three mobility levels, namely, 5m/s, 10m/s and 15m/s.

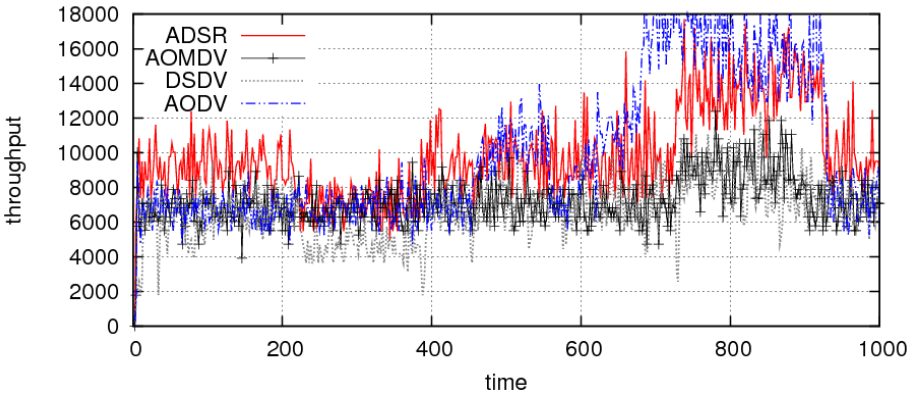

Fig. 6. ADSR vs. AOMDV, DSDV and AODV. The throughput is measured under CBR traffic against the maximum nodal speed of 5m/s.

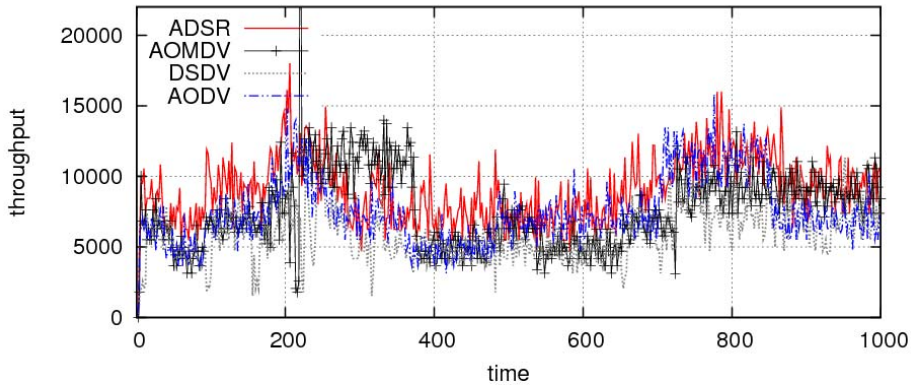

Fig. 7. ADSR vs. AOMDV, DSDV and AODV. The throughput is measured under CBR traffic against the maximum nodal speed of 10m/s.

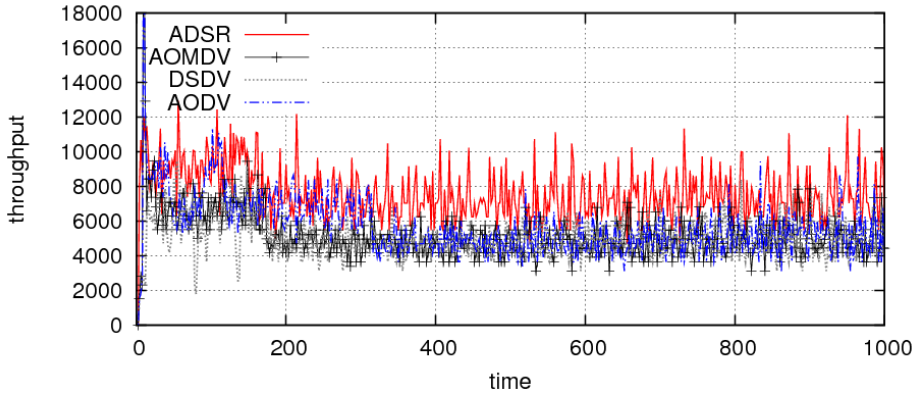

Fig. 8. ADSR vs. AOMDV, DSDV and AODV. The throughput is measured under CBR traffic against the maximum nodal speed of 15m/s.

3.3.2 Throughput for TCP Traffic

Fig. 9, **Fig. 10** and **Fig. 11** illustrate the throughput performance of ADSR with respect to AOMDV, ADOV and DSDV under TCP traffic. ADSR clearly offers better throughput under the given network conditions. Two cases of TCP and CBR can be compared. The throughput of TCP traffic shows larger enhancement against other protocols than that of CBR. As CBR uses UDP, errored CBR packets are just dropped. However, in TCP traffic, they must be retransmitted. TCP can create redundant packets in an unstable route. It can be known that ADSR provides a stable route.

3.4 Normalized Routing Overhead

Normalized routing overhead is the number of routing packets that are transmitted per data packet that's sent to the destination. This measurement is closely associated with the number of route changes on the network. **Fig. 12** shows the comparative normalized routing overhead of ADSR, AOMDV, DSDV and AODV when FTP application is the traffic source. ADSR performs better in situations where mobility is higher, but it usually shows similar overhead to other uni-path routing algorithms. The same measurement is performed in **Fig. 13** with the

traffic source of CBR. ADSR shows much better performance than AOMDV, AODV and DSDV.

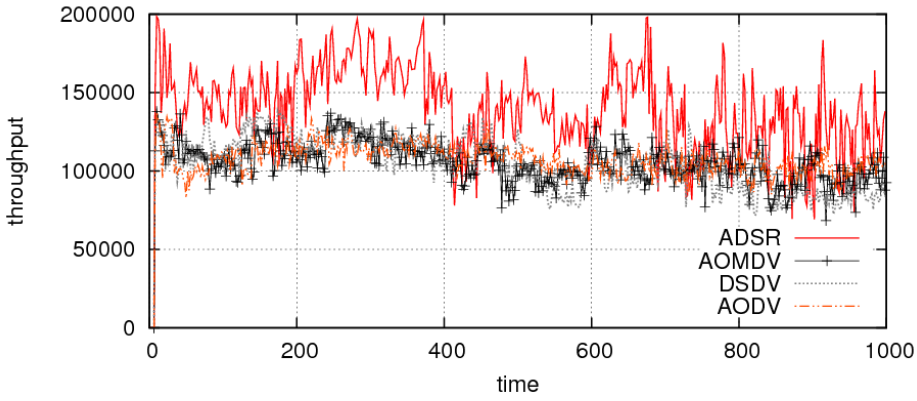

Fig. 9. ADSR vs. AOMDV, DSDV and AODV. The throughput is measured under TCP traffic against the maximum nodal speed of 5m/s.

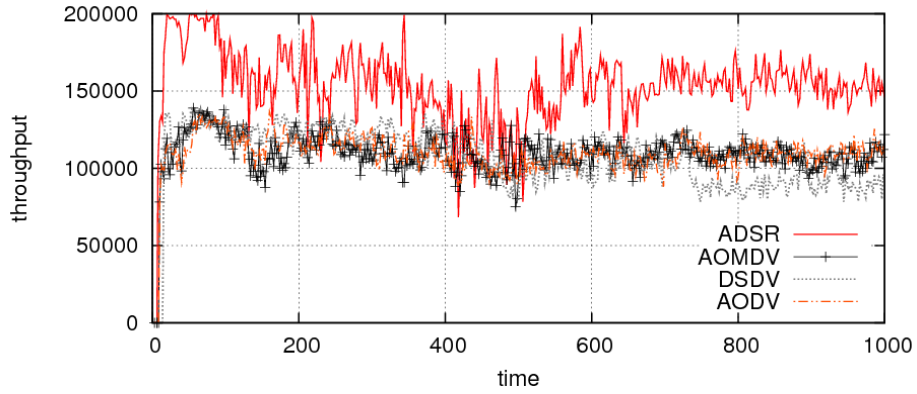

Fig. 10. ADSR vs. AOMDV, DSDV and AODV. The throughput is measured under TCP traffic against the maximum nodal speed of 10m/s.

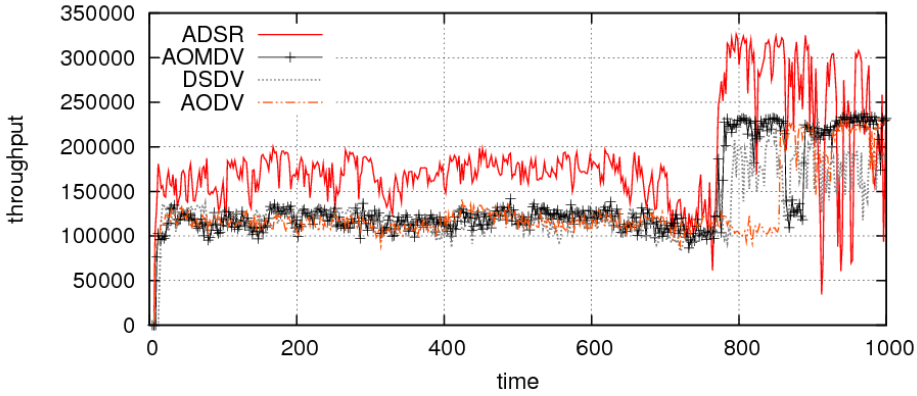

Fig. 11. ADSR vs. AOMDV, DSDV and AODV. The throughput measured under TCP traffic against the maximum nodal speed of 15m/s.

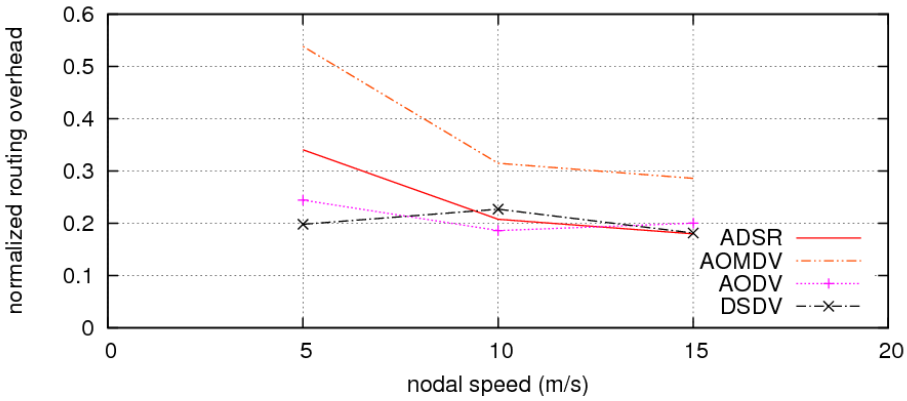

Fig. 12. Normalized Routing Overhead of TCP traffic.

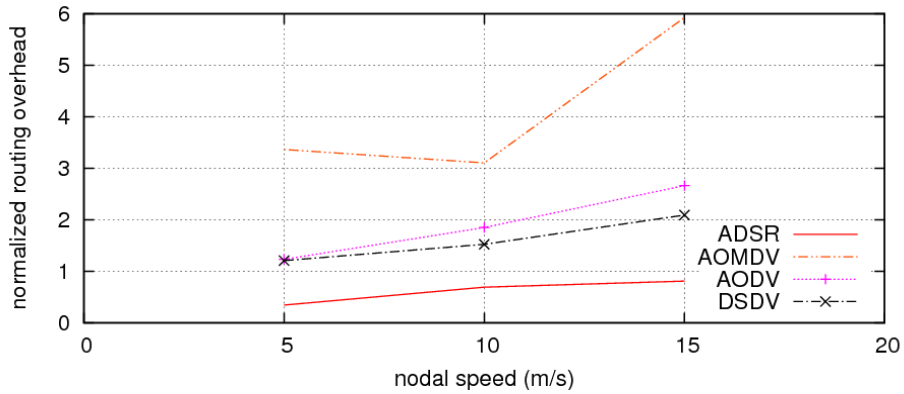

Fig. 13. Normalized routing overhead of CBR traffic.

4. Conclusion

Routing is a core function of any network, and it enables communication among reachable nodes within the network. Designing and developing efficient and robust routing protocols for ad hoc networks is a challenging task due to mobility and resource constraints. In this paper, we have proposed and analyzed simulations of the dynamic source routing protocol, ADSR, through NS2. The protocol was tested under diverse traffic and mobility scenarios. ADSR is compared with two unipath routing protocols, namely, AODV and DSDV and with one multipath routing protocol, AOMDV. The simulation results demonstrate that ADSR is more efficient and robust because it offers better packet delivery ratio and reduces normalized routing traffic overhead, but it significantly improves throughput. Therefore, ADSR is capable of discovering stable node-disjoint multiple routes. The stable route results in less route discoveries and improved fault-tolerance, and hence decreasing the normalized routing overhead. We can conclude that the multiple paths based on temporal and spatial associativity improves reliability by routing traffic over to valid route(s) when route failure occurs.

References

[1] C. E. Perkins, “Ad Hoc Networking,” Addison-Wesley, 2001.

[2] M. Abolhasan, T. Wysocki, E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks, Ad Hoc Networks," Vol. 2, No. 1, pp.1-22, January 2004.
[3] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing , Kluwer Academic Publishers, pp.153-181, 1996.
[4] C. E. Perkins, E. M. Belding-Royer, and S. Das, "Ad Hoc On-Demand Distance Vector (AODV) Routing," IETF RFC 3561, 2003.
[5] C. Liu, M. Yarvis, W. S. Connera, and X. Guo, "Guaranteed on-demand discovery of node-disjoint paths in ad hoc networks," *Computer Communications*, Vol. 30 No. 14-15, pp. 2917-2930, 2007.
[6] P. P. Pham and S. Perreau, "Increasing the network performance using multi-path routing mechanism with load balance," *Ad Hoc Networks* (*ADHOC*), Vol. 2, No. 4, pp. 433-459, 2004.
[7] C. K. Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks," *Wireless Personal Communications Journal*, Vol. 4, No. 2, pp. 103-139, 1997.
[8] C. K. Toh, "A Novel Distributed Routing Protocol to Support Ad Hoc Mobile Computing," *Proc. of the IEEE International Phoenix Conference on Computers and Communications* (*IPCCC*), Scottsdale, AZ, pp. 480-486, 1996.
[9] S. J. Lee, "Routing and Multicasting Strategies in Wireless Mobile Ad hoc Networks," PhD thesis, University of California, 2000.
[10] Wolfram, "Arithmetic Series," Wolfrom MathWorld, http://mathworld.wolfram.com/ ArithmeticSeries.html.
[11] M. K. Marina and S. R. Das, "On-Demand Multipath Distance Vector Routing in Ad Hoc Networks," *Proc. of IEEE International Conference on Network Protocols* (*ICNP*), pp. 14-23, 2001.

Shafqat Ur Rehman received his B.S in computer science from NUCES–FAST, Islamabad in 2003. He received MS from Dept. of computer engineering, Jeju National University, South Korea, majoring in 2008. He joined INRIA, Sophia Antipolis, France as a doctoral student in 2009. His interests mainly include wireless networks. He is currently working on developing and enhancing wireless experimentation platforms and evaluation of network protocols, applications and services in real world wireless scenarios.

Wang-Cheol Song received his B.S in Dept. of Food Engineering from Yonsei University, Korea in 1986. He received B.S, M.S. and Ph.D. from Dept. of Electronics, Yonsei University in 1989, 1991, and 1995 respectively. In 1996, he joined the Dept. of Computer Engineering, Jeju National University, Korea, where he is currently a professor. His research interests include Mobile ad hoc networks, Network Management, and structured P2P technology.

Gyung-Leen Park received his B.S. in computer science from Chung-Ang University. He received M.S. and Ph.D. from Dept. of computer science and engineering, University of Texas at Arlington. In 1998, he joined the Dept. of computer science and statistics, Jeju National University, South Korea, where he is currently a professor and the director of the ITRC. His research interests include parallel and distributed systems, mobile computing, and telematics.