アドホックネットワークにおける動的経路短縮機構の評価

斉藤 匡人[†] 間 博人[†] 徳田 英幸[†,‡]

[†] 慶應義塾大学大学院 政策・メディア研究科, [‡] 慶應義塾大学 環境情報学部
{masato, haru, hxt}@ht.sfc.keio.ac.jp

本論文では、アドホックネットワークにおける動的経路短縮機構である、“DPS” に関して述べる。従来のアドホックネットワークの経路制御は、ノードの移動に起因するネットワーク・トポロジの変化に動的に対応できないため、最短ホップ数の経路を常に使用できるとは限らない。DPS では、データ通信中における自ホスト内の通信リンク品質を基にして、ノードの移動に適応的な経路制御を行うことで、動的経路短縮を実現する。既存のアドホック経路制御プロトコルである DSR と AODV に提案する DPS 機構を組み込み、*ns-2* シミュレータと実環境の実装において性能評価を行った。既存プロトコル比較して、DPS 機構が通信遅延と経路制御情報オーバヘッドの大幅な縮小を実現することを詳細に示す。

On the Effectiveness of Dynamic Path Shortening for Ubiquitous Ad Hoc Networks

Masato Saito[†] Hiroto Aida[†] Hideyuki Tokuda[†,‡]

[†]Graduate School of Media and Governance, Keio University
[‡]Faculty of Environmental Information, Keio University

This paper describes the design, implementation, and evaluation of a proximity-based dynamic path shortening protocol, called DPS. In DPS, active route paths adapt dynamically to node mobility based on the “local” link quality estimation without exchanging periodic control packets. Simulation evaluations of DPS for several scenarios of node mobility and traffic flows reveal that adding DPS to DSR and AODV (conventional prominent on-demand ad hoc routing protocols) significantly reduces the end-to-end packet latency by 50-percent and the number of routing packets by 70-percent, particularly in heavy traffic cases. We also demonstrate some more simulation results obtained by using our novel realistic node mobility models: the *random oriented model* and the *random escape model*. Finally, simple performance experiments using DPS implementation on FreeBSD OS demonstrate that DPS shortens active routes in the order of milliseconds (about 5 ms).

1 Introduction

Wireless ad hoc networks is expected to play a significant role in ubiquitous computing and communications in the home, office, and many other places. A key challenge to succeed in such communications is adapting to mobility. A mobile ad hoc network is a group of mobile computing devices (nodes) which communicates with each other using multi-hop wireless links. It does not necessarily require any stationary infrastructure such as base stations. In such a network, one important issue for achieving efficient network resource utilization is to update route information reactively depending on a change of network topology and connectivity. Since node mobility in an ad hoc network causes frequent, unpredictable and drastic changes to the network topology, it is especially important for communicating nodes to grasp the change of the network topology and find an efficient route between two communicating nodes.

A number of proposed mobile ad hoc routing protocols can be classified into main two types: *pro-active* and *reactive*. Pro-active protocols attempt to continuously evaluate the routes within the network, so that when a packet needs to be forwarded, the route is already known and can be immediately used. On the other hand, reactive protocols invoke a route determination procedure on an *on-demand* basis. Some comparisons between these different protocols have been published [1], [3]. Both reported results based on simulations show that the reactive protocols perform significantly better than traditional pro-active protocols (DSDV [6]) in most situations.

The previous on-demand routing protocols accommodate route changes only when an active path is disconnected. They cannot adapt to the change of network topology even if another route with less hop count becomes available by the movement of intermediate nodes unless any link is disconnected. In contrast to the conventional protocols, we propose Dynamic Path Shortening (DPS) scheme that tunes up an active path adaptive to node mobility without any link disconnection based on Smoothed Signal-to-Noise Ratio (SSNR) as a link quality value indicator.

In order to shorten an active path, we introduce the notion of *proximity* that represents the “nearness” of two communicating nodes. Each node determines to shorten an active path by using ***proximity*** based on the local SSNR value obtained from their own network interfaces. This local SSNR value is soft state using the internal state from their local network interfaces. DPS is particularly suitable for our conventional situation under slow node mobility (e.g., pedestrian and slow vehicle in campus computing) or dense mobile ad hoc network. In addition, since DPS operates only when forwarding or receiving data packets, it does not require periodic HELLO messages or advertisements when there are no link connectivity changes in the data path.

2 Related Work

Dynamic Source Routing (DSR) [5, 9] is an on-demand routing protocol which uses aggressive caching and source routing to obtain the topology information. A DSR node is able to learn routes by overhearing packets not addressed to it by operating its network interfaces in promiscuous receive mode. This scheme also automatically shortens the active paths as well as our DPS scheme while sending data packets. The feature can achieve the dynamic multi-hop path shortening, thus it leads the drastic improvement of the packet latency. However, this scheme requires an always-active transceiver mode of the network interfaces and more CPU cycles to process overheard packets, which may be significantly power consuming.

Roy [8] presents the source-tree on-demand adaptive routing protocol (SOAR) based on link-state information. SOAR has the mechanism to shorten the active paths, but it achieves that by ***periodically*** exchanging link-state information in which a wireless router communicates to its neighbors the link states of only those links in its source tree that belong to the path it chooses to advertise for reaching destinations with which it has active flows. As this partial topology broadcast algorithms exchange control packets of relatively larger size including the minimal source tree, total byte overhead due to control packets has been found to be 2–3 times more in SOAR compared to the previous ad hoc routing protocols (e.g., even DSR).

3 Dynamic Path Shortening

This section describes the detailed design of DPS. First, we explain some scenarios in which DPS effectively shortens active paths. Second, we introduce the notion of proximity to identify two near nodes by using link quality and discuss deeply the link quality. Finally, we explain DPS protocol to perform active one-hop shortening in detail and outline the simple multi-hop shortening scheme.

3.1 Path Inefficiency

In a mobile ad hoc network, due to node mobility, we encounter a situation shown in Figure 1. In this case, we pay attention to node mobility without link disconnections. For such node mobility, we possibly find the less hop route (i.e., direct hop route shown in Figure 1) than the current route in use.

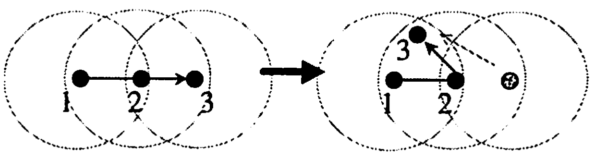


Figure 1: *Node 1 sends packets to Node 3 through Node 2. At the next step, Node 3 moves into the cell of Node 1 without link failures. Although Node 1 can directly send packets to Node 3. Node 1 still sends packets to Node 3 through Node 2.*

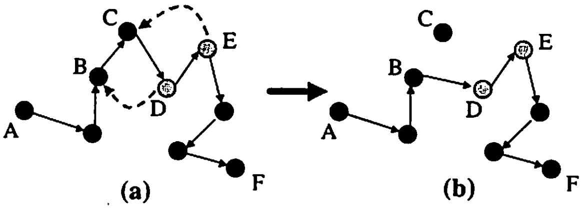


Figure 2: *Node A sends packets to Node F in a multi-hop network.*

Figure 2 shows a more complicated scenario. Some less hop routes are available in the active path from source to destination. If each neighbor node simultaneously shortens the active path (in Figure 2, $D \rightarrow B$ and $E \rightarrow C$), it leads to the isolated routes and deadlocking. We describe how to overcome this problem later.

3.2 The Proximity Area

To argue the "nearness" of two nodes more formally, we introduce the notion of proximity based on the observation of the relationship between the distance and the SSNR between two nodes. Figure 3 demonstrates the proximity area around an active intermediate node.

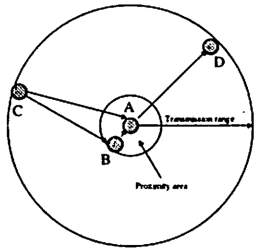


Figure 3: A Proximity Area

To discuss the proximity area logically, we define the following symbols

- $S_{(AB)}$: **The SSNR value observed at Node B for received data packets from Node A.**
- S_{max}: **A threshold value of SSNR.**
- $P_{(A)}$: **The proximity of Node A.**
- $R_{uf}(A)$: **The upstream adjacent node of Node A for flow f.**
- $R_{df}(A)$: **The downstream adjacent node of Node A for flow f.**

We hypothesize that $S_{(AB)} = S_{(BA)}$. This is not impractical since homogeneous nodes are assumed in many mobile ad hoc networks. We will discuss a case in which this assumption does not hold in the future. If $S_{(AB)} \geq S_{max}$, Node B is said to be in the proximity of Node A, or $B \in P_{(A)}$. Based on the above hypothesis, if $B \in P_{(A)}$, then $A \in P_{(B)}$. Figure 4 shows a flow traverses Node A, B, and C in this order. This can be written as $A = R_{uf}(B) = R_{uf}(R_{uf}(C)) = R^2_{uf}(C)$. If $C \in P_{(B)}$, there is a possibility that the path of the flow can be changed: $A = R_{uf}(C)$. As shown in Figure 4, each node is associated with its own proximity. When Node C moves to the proximity of Node B, Node A can directly send data packets to C. In practice, we need a hysteresis mechanism around the threshold value to avoid oscillation.

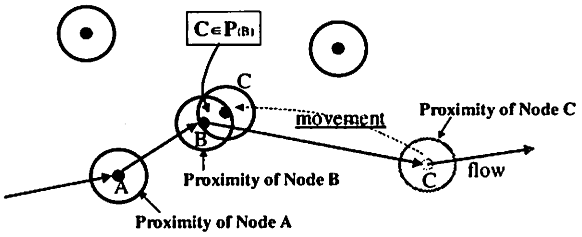


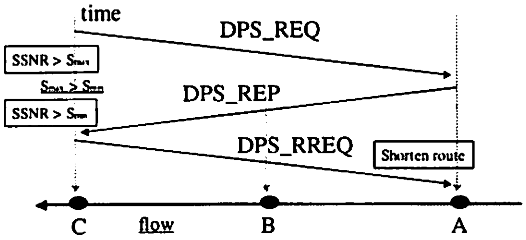
Figure 4: Proximity of node

3.3 Link Quality

In measuring link quality, DPS uses a smoothed value of Signal-to-Noise Ratio (SNR) in a time domain since the SNR could change dynamically with a high frequency due to electro-magnetic effects. This value, Smoothed SNR (SSNR), can be computed using a weighted moving average technique as follows: $ssnr = (1 - \alpha) * old\_ssnr + \alpha * cur\_snr$, where cur_snr and old_ssnr represent the value of SNR on receipt of a packet and the previously computed SSNR, respectively. The constant value of α is a filtering factor and is set to $1/8$ in this paper. It is because we could adapt to the large fluctuation of SNR and use a shift operation in our experimental implementation. In DPS, the filter calculates SSNR whenever a node receives the frames.

3.4 Design of DPS

We set two design goals to DPS: reducing the hop count of a path, and minimizing the number of additional control packets. The first goal is obvious in the context of the problem aforementioned. In addition to the first goal, we aim at a scheme not producing periodic control packets. This is an important consideration for an ad hoc network since nodes in the network need to save their power consumption. We design our scheme so that control packets are transmitted only when a node determines that a path should be changed based on the proximity. In DPS, each node in ad hoc networks has the original routing information concerning upstream two-hop-away nodes. Since a node attempts to transmit the control packet to the upstream two-hop-away node, the node needs to retain the route information of its upstream two-hop-away nodes of the active flows as well as its neighbors.



DPS_REQ	Request to $R^2_{uf}(k)$ as a packet to probe direct route
DPS_REP	Reply to $R^2_{df}(k)$ via $R_{df}(k)$ for DPS_REQ (piggybacked)
DPS_RREQ	Redirect Request to $R^2_{uf}(k)$ via $R_{uf}(k)$ to shorten route

Figure 5: Three DPS control packets

We explain the fundamental messages passed among three nodes. DPS uses three kinds of messages: *DPS_REQ*, *DPS_REP*, and *DPS_RREQ*; they are shown in Figure 5. *DPS_REQ* and *DPS_RREQ* are separately generated control packets, while *DPS_REP* can be piggybacked on a data packet. Let us assume that $A = R_{uf}(B)$ and $B = R_{uf}(C)$ for flow f as shown in Figure 5. When Node C finds it has moved into the proximity of Node B, it sends *DPS_REQ* to Node A. This is to observe whether or not a packet can be directly exchanged between Node A and C. If receiving *DPS_REQ*, Node A sends *DPS_REP* to Node C. Unlike *DPS_REQ*, *DPS_REP* is piggybacked by th data packet of flow f. Thus, *DPS_REP* reaches Node A via Node B. By receiving *DPS_REP*, Node C knows that Node A can send packets directly to Node C; Node C sends *DPS_RREQ* to Node A to initiate a change of route.

4 Performance Evaluation

In this section, we show the detailed simulation and experimental results of DPS. First, we describe the simulator implementation and performance results of DPS adding to DSR and AODV in the *ns-2* network simulator environment. We simulate DPS on several large mobile topologies to quantify the scaling behavior of DPS in Network Simulator (*ns-2*) [10]. In addition, to study how DPS scheme perform in *realistic* node mobility patterns, we measure the effectiveness of DPS using our two practical mobility generation models which are based on the the *random way-point model* [4] which is used in almost the previous simulation research. Finally, we demonstrate building a small wireless ad hoc network testbed and performing a simple preliminary experiment.

4.1 Implementation Decisions

We implemented two DPS schemes in our simulation: the fast shortening mode and robust shortening mode. In the robust mode, *DPS_REQ* sender obtains IP address of the two-hop upstream neighbor from the source routing header of receiving data packets. Since DSR is the source routing protocol, we should report the changes of the active route to send GRATUITOUS_REPLY [5] to the source node when DPS completes shortening of active paths. Furthermore, we implemented the timer routine to reset the above state to compensate such cases as DPS shortening failures or incompletion due to dropping one of the three DPS control packets. This mode is the same as DPS implementation in our real environments.

In the former fast shortening mode, DPS realizes faster shortening at the cost of the completion probability of the shortening operation. This scheme operates as follows. When a node enters the proximity area of his upstream neighbor node, it sends *DPS_REQ* to the two-hop upstream neighbor node as the same way in real implementation. However, the node which received the *DPS_REQ* promptly switch the active route to directly forward data packets to the node which sent the *DPS_REQ*. Since this scheme operates based on the optimistic policy, it cannot assure the consistency of the shortening for the race condition problems as described in the previous chapter. Although the performance of the fast shortening scheme were better than that of the robust mode in preliminary some simulation scenarios, in the end, we chose the robust shortening scheme policy as a conservative solution.

To illustrate that DPS is not specific to DSR protocol, we incorporated DPS mechanism into AODV. The modifications we had to make for AODV were somewhat different than those incorporated for DSR. Specifically, data packets do not carry the full source route in their header since AODV is the distance vector routing algorithm. Thus, the two-hop upstream neighbor is not available from their header. However, we can easily cope with this problem to be forwarded *DPS_REQ* packets by the one-hop upstream neighbor node. In other words, *DPS_REQ* packets travel two-hop journey via the upstream neighbor node as the intermediate node, instead of on-hop direct communication. In the intermediate node, to distinguish own particular route, we needed the matching scheme based on the final destination and the next hop node.

For comparison with DSR based DPS, we chose to implement DPS on AODV-LL (Link Layer) [1] using only link layer feedback from 802.11 as in DSR, completely eliminating the standard AODV Hello mechanism.

4.2 Detailed Simulation

We use a detailed simulation model based on *ns-2* in our evaluation. *Ns-2* has the enormous features for simulating multi-hop wireless networks complete with physical, data link and MAC layer models [1]. The distributed coordination function (DCF) of the IEEE standard 802.11 for wireless LANs is used as the MAC layer. The radio model uses a shared-media radio with a nominal bit-rate of 2 Mb/sec and a nominal radio range of 250 meters.

DSR and AODV protocols detect link breakage using feedback from the MAC layer. A signal is sent to the routing layer when the MAC layer fails to deliver a unicast packet to the next hop. In this evaluation, no additional network layer mechanism such as *HELLO Messages* [7] is used. Table 1 and 2 provide all the simulation parameters of both protocol extended by DPS. These parameters are remained default parameters of *ns-2* current distribution except DPS parameters.

Table 1: DSR Simulation Parameters

Time between retransmitted Route Requests (exponentially backed off)	500 ms
Size of source route header carrying n addresses	$4n + 4$ bytes
Timeout for non-propagating search	30 ms
Time to hold packets awaiting routes	30 s
Max rate for sending gratuitous Replys for a route	1/s
Max rate for sending DPS Request for a route	3/s

Table 2: AODV-LL Simulation Parameters

Time for which a route is considered active	50 sec
Lifetime on a Route Reply send by destination node	1 sec
Number of times a Route Request is retried	3
Time before a Route Request is retried	10 s
Time for which the broadcast id for a forwarded Route Request is kept	6 sec
Time for which reverse route information for a Route Reply is kept	10 sec
Time before broken link is deleted from routing table	3 sec
MAC layer link breakage detection (Hello Packets OFF)	yes
Max rate for sending DPS Request for a route	3/s

4.2.1 Traffic and mobility models

Traffic and mobility models use similar to previous published results using *ns-2* ([1], [3], [2]) for appropriate performance comparisons. Traffic sources are Constant Bit Rate (CBR). The source and destination pairs are spread randomly over the network. Only 512 byte date packets are used. The number of source-destination pairs and the packet sending rate in each pair is varied to change the offered load in the network.

To investigate how DPS scheme perform in the *realistic* node mobility pattern, we proposed the two node mobility models: the "random oriented model" and "random escape model". These models are based on the the *random way-point model* [4] used in most of the previous simulation research.

In the *random way-point model*, each node begins the simulation by remaining stationary for *pause time* seconds. It then selects a random destination in the specified field space and moves to the destination at a speed distributed uniformly between *0* and some maximum speed. On reaching the destination, the node pauses again for *pause time* seconds, selects another destination, and proceeds there as previously described, repeating this behavior for the duration of the simulation.

In contrast, our two node mobility model generate more realistic movement patterns. The random oriented node mobility is assuming people pursuing something (e.g., *peace*, *money*, *hope*) or attracted something(e.g., *gravity*, *power*). On the other hand, the random escape model is literally assuming people are escaping from something (e.g., *disaster*, *ghost*). In our proposed models, mobile nodes are classified into two types: CORE_NODES (CN) and ORIENTED_NODES (ON) or ESCAPE_NODES (EN). CN move around the simulation field based on the random way-point models accurately. The other side, ON selects one destination from the destinations of CN instead of a random destination and pursue one CN at a speed distributed uniformly between *0* and some maximum speed (*not all people require money*). If one ON reaches the selecting destination, then it selects another destination among that of CN. In the random escape model, EN desperately leave from one particular CN. EN select the exact opposite side destination to the particular destination of one CN. By the random escape model, we consider human mobility in the situations as disaster to where ad hoc networks expect to apply. Note that, when the node mobility file is generate, we specify the ratio of ON or EN to CN as one argument. If the stated ratio is 0.0, the generated node mobility pattern is accurately based on the random way-point model.

We use the above-mentioned three mobility model in a rectangular area. $1500m \times 300m$ field configuration with 50 nodes is used. Thus, each node starts its travel from a random location with a randomly chosen speed (uniformly distributed between 0 – 20 m/sec except in the random escape model). We vary the pause time, which affects the relative speeds of the mobile nodes; in this thesis, we used the following pause times (0, 30, 60, 120, 300, 500 [sec]). Simulation are run for 500 simulated seconds for 50 nodes. Each data point represents an average of ten runs with identical traffic models, but different randomly generated mobility scenarios.

In all the below experiments, we assumed that the useful range of proximity should be restricted below S_{max} = 0.000008 obtained from our preliminary analysis and experiments of SNR.

4.2.2 Performance Results

Light Traffic Loads

First, we perform experiments using the light traffic loads to study the behavior of DPS added DSR and AODV. For the 50 node experiments we used 10 traffic sources and a packet rate 4 packet/sec. We found that DPS improves the end-to-end delay as expected and reduces the routing control packet overhead effectively (see Figure 6.(a), 6.(b) and 6.(c)). However, in the packet delivery ratio (PDR), DPS loses about 5 - 10% packets. While we are currently working the accurate reason of lost packets, we think that the reason is by the failures of the path shortening. In Figure 6.(c), note that the packet delivery fractions for DSR are more than 100 %, we think that it may be caused by the retransmissions of data packets performed by DSR protocol.

Heavy Traffic Loads

To stress the traffic loads to DPS, we used 30 traffic sources. The other configuration parameters are the same as the above the light traffic load experiments. In Figure 7.(a), 7.(b) and 7.(c), we can see that DSR with DPS achieves the significantly reduction of the packet delay and routing overhead. Additionally, in contrast to the first simulation experiment, DSR with DPS has the high performance of PDR. However, AODV with DPS does not show the improved performance as salient as DSR with DPS. We think that one of the reasons is the potential feature of AODV protocol; AODV node holds many state information and uses the timer-based routine frequently. Hence, under high node mobility, AODV may not operate normally. In fact, such the simulation results were pointed out in the previous research work [1, 2].

Random Oriented Mobility Model

This model typically makes several network and node congestion points. Thus, we can assume the effectiveness of the active shortening in such a area. In Figure 8.(a), 8.(b) and 8.(c), we can see that the improved delay reduction is significant. To generate heavy mobility loads, we have set the ratio of oriented nodes to core nodes to *0.8* (i.e., in 50 mobile nodes case, the number of oriented nodes is *40*) .

Random Escape Mobility Model

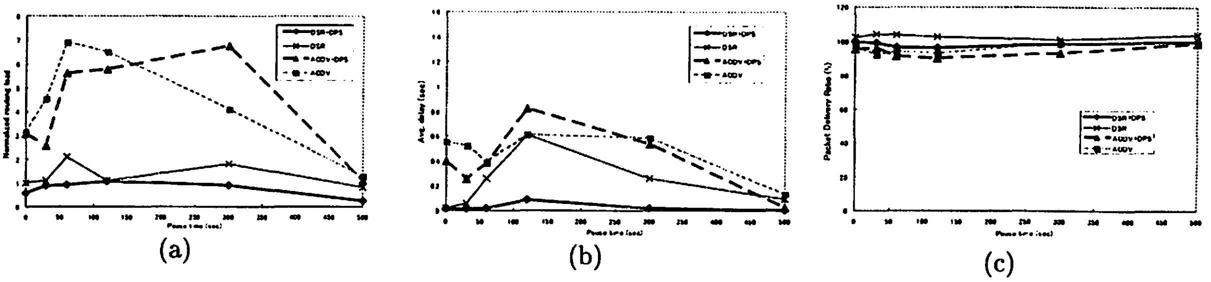
This model makes some network partition areas intentionally. Thus, mobile ad hoc nodes suffer from frequently link failure and relatively speedy node mobility. As Figure 9.(a), 9.(b) and 9.(c) show, DPS improved the performance of DSR. However, AODV protocol created long-lived routing loops in the simulation, so we could not perform the evaluation of AODV and AODV with DPS. It seems that AODV does not perform well in the situations where several network partitions exist frequently. Now, we are trying to improve the ns-2 AODV code. In this case, we used the ratio of escape nodes to core nodes to *0.8*.

4.3 Simple Experimental Study

We have implemented DPS as an extension to DSR developed by the Monarch project [9]. To observe the overhead associated with path shortening, we have conducted five trials of path shortening among three nodes. The overhead latency (including packets processing time) incurred with the exchange of DPS's control messages is sufficiently negligible; it is the order of milliseconds (about 5 ms).

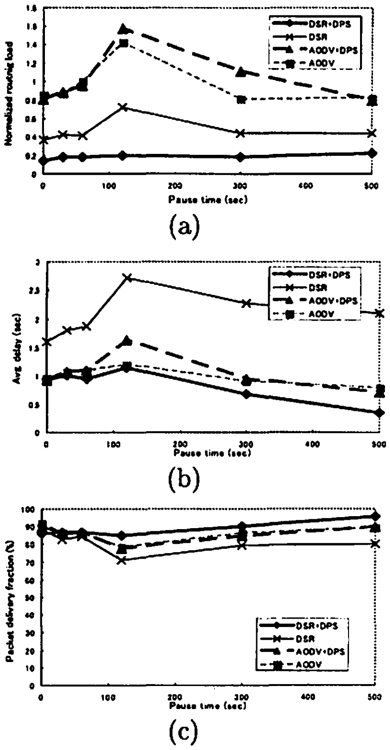
5 Conclusion

We have proposed DPS, an adaptive route path tuning algorithm for mobile ad hoc networks. Our approach is highly responsive to the conventional human mobility (e.g., pedestrian and slow vehicle in our daily life) by using the wireless link quality value: SSNR. In DPS, each node individually monitors local link quality only when receiving packets and makes local decisions in a decentralized manner. As a case study of DPS scheme, we embedded DPS in the prominent DSR and AODV protocols. Performance analysis by means of simulation demonstrates a significant improvement with regard to the number of routing packets and end-to-end data packet latency in almost scenarios. We have also proposed two novel mobility models



(a) (b) (c)

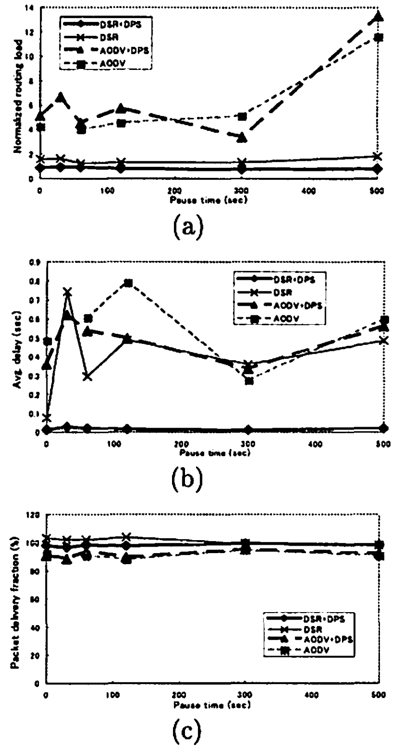
Figure 6: 10 Sources Model



(a)

(b)

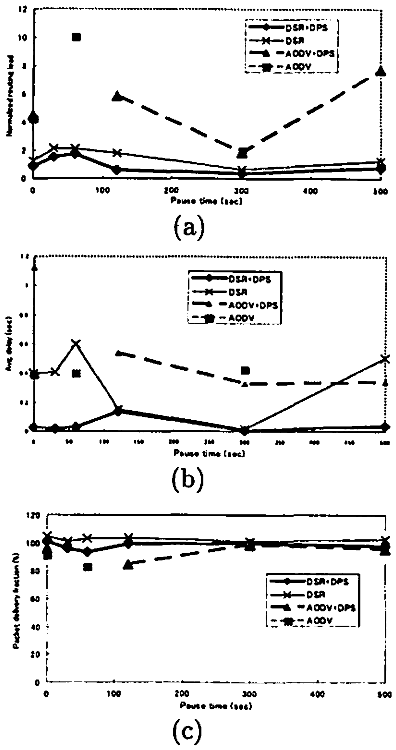
(c)



(a)

(b)

(c)



(a)

(b)

(c)

Figure 7: 30 Sources Model

Figure 8: Random Oriented Model

Figure 9: Random Escape Model

which generate more realistic node mobility than the random-waypoint model. Additionally, DPS shortens active routes in the order of milliseconds (about 5 ms) in our real DPS implementation.

We are currently working on studying the effects of DPS in case of using the promiscuous listening mode of network interfaces. We also need to analyze more the appropriate value of the SSNR threshold S_{max} and the comparison frequency since these factors have significant impact on its effectiveness of DPS.

References

[1] Josh Broch, David A. Maltz, David B. Johnson, et al. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking*, October 1998.

[2] Samir R. Das, Charles E. Perkins, and Elizabeth M. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. In *Proceedings of IEEE INFOCOM'00 (1)*, pages 3-12, March 2000.

[3] Per Johansson, Tony Larsson, Nicklas Hedman, and Bartosz Mielczarek. Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks. In *Proceedings of ACM MOBICOM'99*, August 1999.

[4] David B. Johnson and David A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. In Mobile Computing, edited by Tomasz Imeilinski and Hank Korth, chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[5] David B. Johnson, David A. Maltz, and Yin-Chun Hu. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. IETF Internet-Draft [Work in Progress], April 2003.

[6] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proceedings of ACM SIGCOMM'94*, August 1994.

[7] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das. Ad Hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, July 2003.

[8] Soumya Roy and J. J. Garcia-Luna-Aceves. Using Minimal Source Trees for On-Demand Routing in Ad Hoc Networks. In *Proceedings of IEEE INFOCOM'01*, August 2001.

[9] The Rice University Monarch Project. The MONARCH Project at Rice University. http://www.monarch.cs.rice.edu/, 2002.

[10] The VINT Project. Network simulator - ns2. http://www.isi.edu/nsnam/ns, 2001.