DPS: A Dynamic Path Shortening Scheme for Mobile Ad Hoc Networks

Masato Saito[†]

[†]Graduate School of Media and Governance Keio University

Fujisawa, Kanagawa, Japan

masato@ht.sfc.keio.ac.jp

Abstract

This paper describes the design, implementation, and evaluation of a proximity-based dynamic path shortening scheme, called DPS. In DPS, active route paths adapt dynamically to node mobility based on the local link quality without exchanging periodic control packets. Most conventional on-demand routing protocols accommodate the change of network topology only when the link fails. Unless the movement of intermediate nodes leads to any link failures, they cannot adapt to the network topology even if other routes with less hop count become available. DPS skips the upstream node in a *proximity* area that indicates the nearness of two communicating nodes and continues to shorten an active route as possible.

Simulation studies for several scenarios of node mobility and traffic flows reveal that adding dynamic path shortening to DSR and AODV (conventional prominent on-demand ad hoc routing protocols) significantly reduces the packet latency and the number of routing packets, particularly in heavy traffic cases. We also show some simulation results obtained by introducing the two novel models to generate the realistic node mobility: the *random oriented model* and the *random escape model*. Finally, simple performance experiments using our implementation on FreeBSD OS demonstrate that DPS shortens active routes in the order of milliseconds (about 5 ms), so DPS is effective in enhancing TCP throughput and reducing end-to-end delay for all relevant flows.

1 Introduction

As popularity for mobile computing increases, cooperative communications with wireless devices are becoming an attractive technology. A key challenge to succeed in such communications is adapting to node 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 require any stationary infrastructure such as base stations. Each node in the network can act as both a host and a router forwarding data packets to other nodes.

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 research for mobile ad hoc networks has focused on the development of their routing protocols (e.g., DSR [3], AODV [14], SOAR [15], OLSR [9], TBRPF [1], ZRP [8]). These 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 [4], [10]. Both reported results based on simulations show that the reactive protocols perform significantly better than traditional pro-active protocols (DSDV [13] and ZRP) in most situations. The key advantage behind reactive (on-demand) protocols is the reduction of routing overheads so that on-demand routing protocols maintain only active paths to those destinations to which data must be sent. Minimizing the routing overhead is effective in such a dynamic environment of ad hoc networks due to limited available bandwidth, unpredictable nodes mobility, battery outages, interference and high bit error rates.

Above 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. DSR protocol [3] only has the mechanism that shorten an active path which is not driven by link failures but by overhearing packets by operating the network interfaces in promiscuous receive mode. This promiscuous mode, however, requires greater CPU cycles, power consumption and sending delay due to overheard packets. 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. The adaptation to node mobility by DPS leads to the reduction of a hop count and path delay which significantly improves the performance of Transmission Control Protocol (TCP) flows.

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. Thus, we can change the active path while preserving stable link connectivity. 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.

The rest of this paper is organized as follows. Section 2 briefly describes some proposed protocols in mobile ad hoc networks. Section 3 describes a design and a detailed description of DPS, and Section 4 explains the implementation of it. Then, we present the evaluation results and analysis of detailed simulation and some experiments in Section 5. Finally, we present our conclusions and discuss some future works in Section 6.

2 Related Work

This section describes conventional on-demand routing protocols in mobile ad hoc networks.

Dynamic Source Routing (DSR) [3, 17] is an on-demand routing protocol which uses aggressive caching and source routing headers 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. This is especially inefficient in environments where battery power is a scarce resource. Also, because DSR does not take the link quality into account, it possibly leads to inefficient and frequent route change and the great degradation of the link quality.

Roy [15] presents the source-tree on-demand adaptive routing protocol (SOAR) based on link-state information. SOAR incurs much less overhead of control routing packets than DSR under various scenarios, ranging from high mobility to low mobility. SOAR also 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). High control packet overhead is undesirable in low-bandwidth wireless environments. Additionally, periodical exchanging messages could collide with the data streams, thereby may degrading the performance. The mechanism to maintain and synchronize the minimum source tree of its own neighbor nodes with the varying network conditions is fairly complex and computational overhead.

In contrast to the above works, "DPS" does not lead to the weak-connectivity shortened routes or inefficient frequent routes switching since it is based on local link quality, and does not need periodic information advertisements or any overheard packets by making the network interfaces promiscuous receiving mode. DPS adapts effectively to node mobility using local link quality in wireless ad hoc networks which are scarce bandwidth and battery environment.

3 DPS

This section describes the detailed design of DPS. First, we explain some scenarios in which DPS effectively shorten 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 disconnection. 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.

This scenario in particular is likely to occur frequently in the realistic environment with pedestrian or slow vehicle speed in our daily life. However, the most of the previous routing protocols cannot accommodate the change of network topology without any link failures. Thus, there exists the path inefficiency in respect to the hop count, network capacity and power consumption while communicating with other nodes in an ad hoc network. We eliminate the inefficiency by using

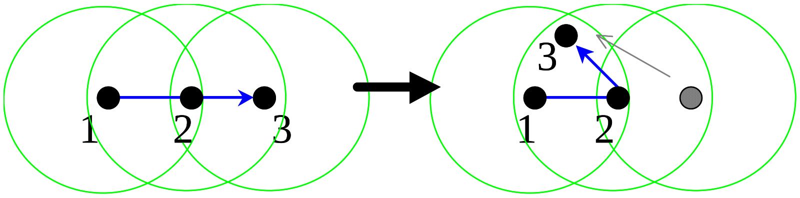


Figure 1: **Node 1 sends packets to Node 3 through Node 2. At the next step Node 3 moves into the cell of Node 1 without link failures. Although Node 1 can directly send packets to Node 3, Node 1 still sends packets to Node 3 through Node 2.**

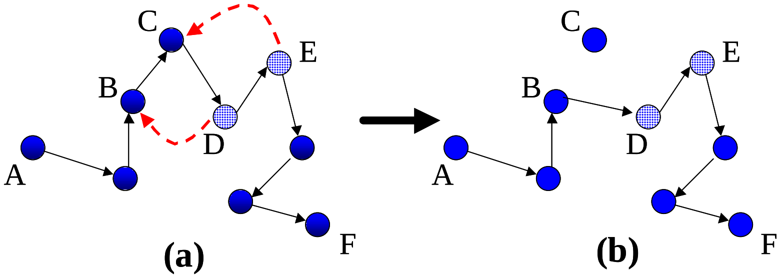


Figure 2: **Node A sends packets to Node F in a multi-hop network. By using DPS's algorithm, Node D and E can shorten the route currently in use preserving the consistency of the active route.**

local link information and the concept of *proximity* in the next section.

On the other hand, DPS also finely accommodates large-scale and dense networks since it is decentralized algorithm using local link quality information. Figure 2 shows a more complicated scenario in which DPS is tuning up the active path adapting to node mobility. In the figure, 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. As a result, the active path from source to destination is failed and the sender node must re-initiate a new route discovery. 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.

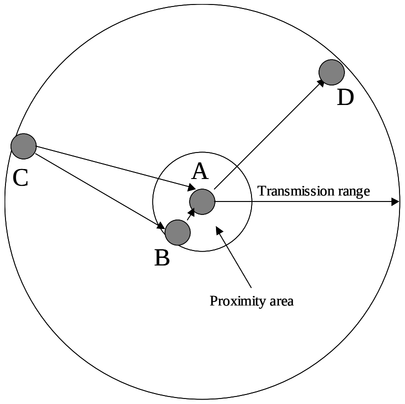


Figure 3: Proximity of node

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 5 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 5, 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. The basic pseudo-code describing our shortening algorithm is shown in Figure 4.

In practice, we need a hysteresis mechanism around the threshold value to avoid oscillation.

```
if ($S_{(K \cdot R_{uf}(K))} \geq S_{(max)}$)
    Node $K$ sends DPS_REQ to $R^2_{uf}(K)$
if Node $K$ received DPS_REQ
    Node $K$ changes route and then sends directly DATA to $R^2_{df}(K)$
```

Figure 4: DPS Algorithm

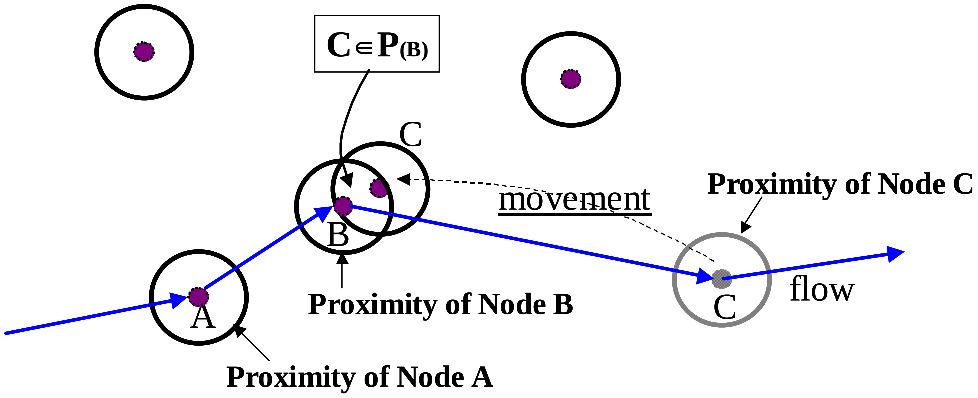


Figure 5: Proximity of node

3.3 Link Quality

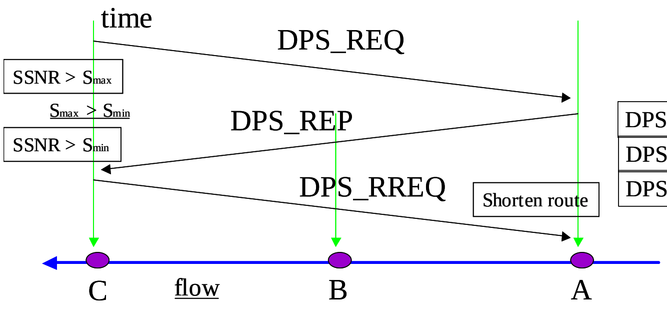
It is desirable for a node on a path to determine whether or not it can shorten the path based on some indicators of the quality of the link between the node and its neighbors on the path. For such an indicator, we use the Signal-to-Noise Ratio (SNR) of the link associated with receiving packets. By definition, SNR represents a channel condition and is expressed as the ratio of signal to noise in electrical power. When the value of SNR becomes higher, the link communication quality will also be relatively higher. However, it should be noted that the SNR could change dynamically with a high frequency due to electro-magnetic effects.

From the point of view of measuring the link quality, DPS uses a smoothed value of SNR in a time domain. 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. In particular, we do not allow transmission of control packets when active flows do not exist. This is an important consideration for an ad hoc network since nodes in the network need to reduce their power consumption.

We design our scheme in which control packets are transmitted only when a node determines that a path should be changed based on the proximity. We call this scheme DPS. In designing DPS, we made an assumption that 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 6: Three DPS control packets

Let us now 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 6. DPS_REQ and DPS_RREQ are newly 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 6.

When Node C determines that it has moved into the proximity of Node B, it sends DPS_REQ to Node A. The intent is to observe whether or not a packet can be directly exchanged between Node A and C. Upon receipt of DPS_REQ, Node A sends DPS_REP to Node C. Unlike DPS_REQ, DPS_REP is rather piggybacked by th data packet of flow f than sending as a single control packet. Therefore 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. The extra packets of DPS_REQ and DPS_RREQ may temporarily interfere with data packets. However, the overhead incurred with the packets is still negligibly small compared with an alternative scheme using HELLO messages.

There is concern about a race condition; simultaneous attempts by each adjacent nodes to shorten the same path may occur as shown in Figure 2. We solve this problem in a way similar to TCP's three-way handshake but in a more delicate way to handle mutual exclusion. The state transition of DPS at node K of flow f is shown in Figure 7.

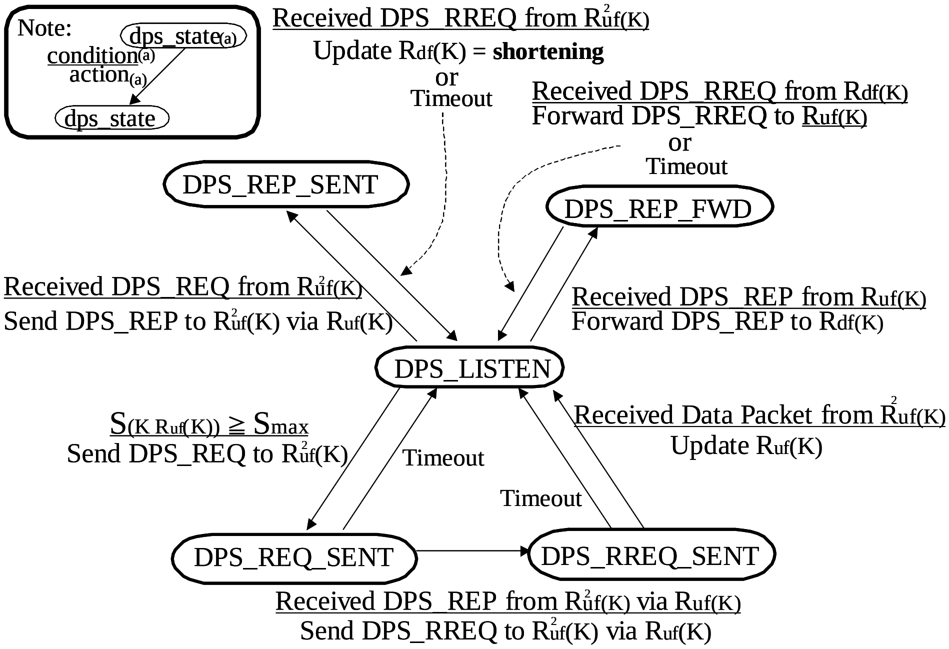


Figure 7: DPS state transition diagram at node K

Let us assume that $A = R_{uf}(B)$, $B = R_{uf}(C)$, and $C = R_{uf}(D)$ for flow f. When $S_{(BC)} \geq S_{max}$, Node C sends DPS_REQ to Node $R_{uf}^2(C)$ (i.e., Node A) to locate the direct hop route. As long as $S_{(BC)} \geq S_{max}$, Node C continues to send DPS_REQ every time DPS's timer expires until Node C receives DPS_REP. Upon successful receipt of DPS_REP, Node C sends DPS_RREQ to Node A to ask for the redirection of the path of flow f. Upon success in the above process, Node A can directly send data packets to Node C.

If we consider a case in which Node D is also attempting to make a short cut between Nodes B and D, Node D sends DPS_REQ to Node B. When Node B receives DPS_REQ, the state of flow f at Node B moves to DPS_REP_SENT. If there is an DPS_REP message from Node A to C, it traverses Node B. When this DPS_REP message reaches Node B and the state is DPS_REP_SENT, the message is discarded since the short cut between Nodes B and C is on-going. Thus the short cut from Node B to C is prioritized. In contrast, if an DPS_REP message from A to C reaches Node B ahead of an DPS_REQ message from B to D, the state of B changes to DPS_REP_FWD and suppresses the short cut from Node B to C.

4 Performance Evaluation

In this section, we show the detailed simulation results of DPS and the experimental results. 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 qualify the scaling behavior of DPS in Network Simulator (ns2) [18]. 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* [11] used in most of the previous simulation research. Finally, we demonstrate building a small wireless ad hoc network testbed and performing a simple preliminary experiment.

4.1 DSR and AODV 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 [3] to the source node when DPS completes shortening of active paths. Furthermore, we implemented the timer routine to reset the above states to compensate such cases as DPS shortening failures or incompletion due to dropping one of the three DPS control packets, etc. 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 the dynamic path shortening 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.

Also, AODV in *ns-2* was implemented as a user-land application daemon, the final destination node cannot initiate the shorten route, unlike the cases in DSR. For comparison with DSR based DPS, we chose to implement DPS on AODV-LL (Link Layer) [4] 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. On *ns-2*, the Monarch research group in CMU developed support for simulating multi-hop wireless networks complete with physical, data link and MAC layer models [4]. The distributed coordination function (DCF) of the IEEE standard 802.11 for wireless LANs is used as the MAC layer. The 802.11 DCF uses Request-to-Send (RTS) and Clear-to-Send (CTS) control packets [2] for "unicast" data transmission to a neighboring node. The radio model uses characteristics similar to a commercial radio interface, Lucent's WaveLAN [12, 19]. WaveLAN is a shared-media radio with a nominal bit-rate of 2 Mb/sec and a nominal radio range of 250 meters.

The routing protocol model handles all data packets transmitted or forwarded, and responds by invoking routing activities as appropriate. The ROUTE REQUEST (RREQ) packets are treated as broadcast packets in the MAC. ROUTE REPLY (RREP), ROUTE ERROR (RERR) and data packets are all unicast packets with a specified neighbor as the MAC destination. 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

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

additional network layer mechanism such as *HELLO Messages* [14] 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.

4.2.1 Traffic and mobility models

Traffic and mobility models use similar to previous published results using *ns-2* ([4], [10], [6]) for appropriate performance comparisons. Traffic sources are CBR (constant bit rate). 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* [11] 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*, etc) 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*, etc). In our proposed models, mobile nodes are classified into two types: Core Nodes and Oriented or Escape Nodes. Core Nodes move around the simulation field based on the random way-point models accurately. The other side, Oriented Node selects one destination from the destinations of Core Nodes instead of a random destination and pursue one Core Node at a speed distributed uniformly between *0* and some maximum speed (*not all people require money*). If one Oriented Node reaches the selecting destination, then it selects another destination among that of Core Nodes. In the random escape model, Escape Nodes desperately leave from one particular Core Node. Escape Nodes select the exact opposite side destination to the particular destination of one Core Node. 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 Oriented Nodes or Escape Nodes to Core Nodes 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. One field configurations are used - (i) $1500m \times 300m$ field with 50 nodes. 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

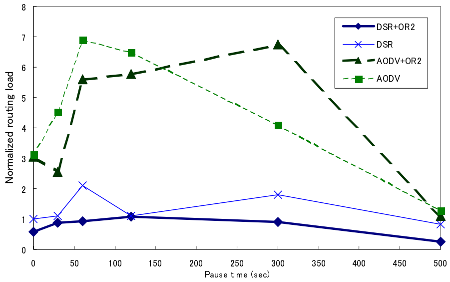
First, we perform experiments using the light traffic loads to study the behavior of DPS added DSR and AODV. For the 50 node experiments we used 10 traffic sources and a packet rate 4 packet/sec. We found that DPS improves the end-to-end delay as expected and reduces the routing control packet overhead effectively (see Figure 4.2.2, 4.2.2 and 4.2.2). However, in the packet delivery ration (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 4.2.2, 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

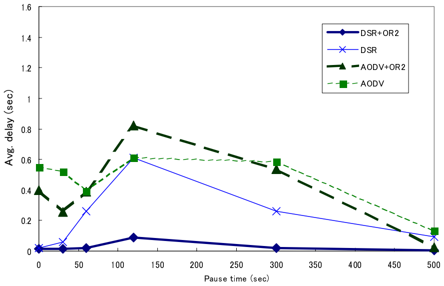
To stress the traffic loads to DPS, we used 30 traffic sources. The other configuration parameters are the same as the above the light traffic load experiments. In Figure 4.2.2, 4.2.2 and 4.2.2, we can see that DSR with DPS achieves the significantly reduction of the packet delay and routing overhead. 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 [4, 6].

Random Oriented Mobility Model

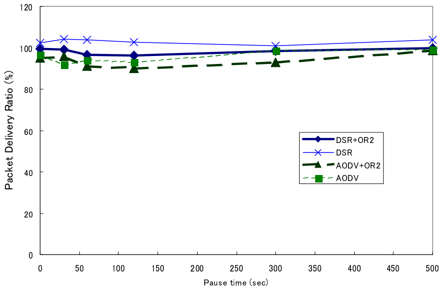
This model typically makes several network and node congestion points. Thus, we can assume the effectiveness of the active shortening in such a area. In Figure 4.2.2, 4.2.2 and 4.2.2, we can see that the improved delay reduction is significant. To generate heavy mobility loads, we have set the ratio of oriented nodes to core nodes to *0.8* (i.e., in 50 mobile nodes case, the number of oriented nodes is *40*) .



Normalized routing load

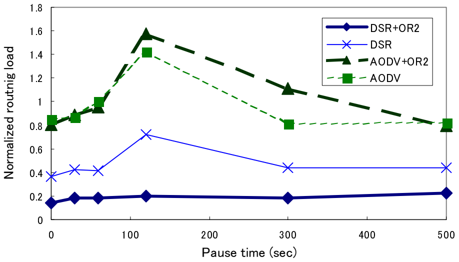


Average data packet delay

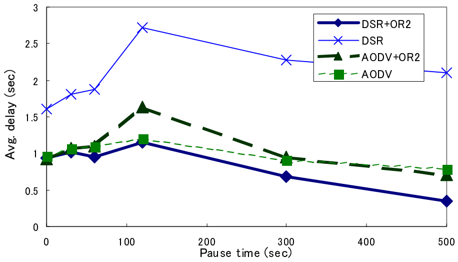


Packet delivery fraction

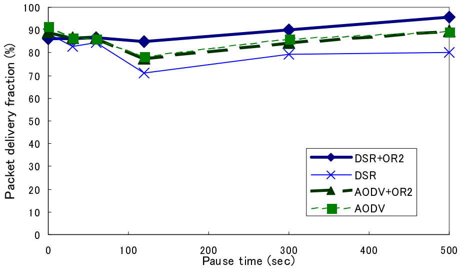
Figure 8: 10 sources model



Normalized routing load



Average data packet delay

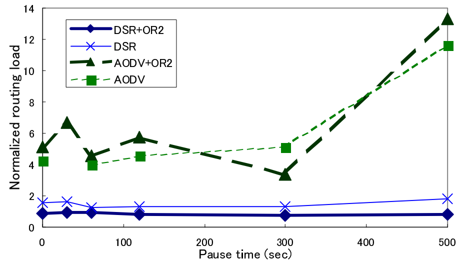


Packet delivery fraction

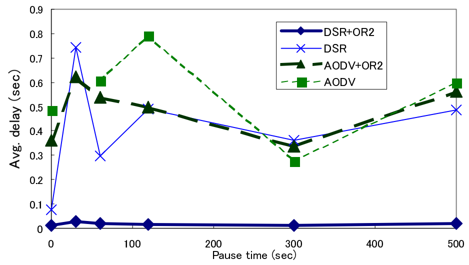
Figure 9: 30 sources model

Random Escape Mobility Model

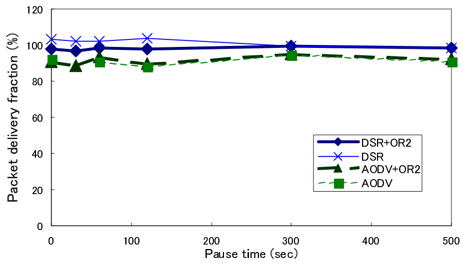
This model makes some network partition areas intentionally. Thus, mobile ad hoc nodes suffer from frequently link failure and relatively speedy node mobility. As Figure 4.2.2, 4.2.2 and 4.2.2 show, DPS improved the performance of DSR. However, AODV protocol created long-lived routing loops in the simulation, 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*.



Normalized routing load

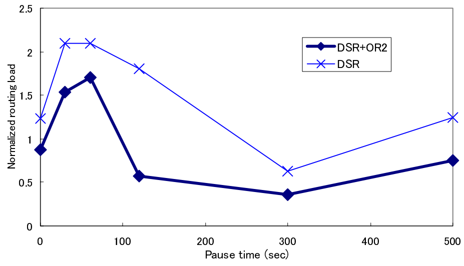


Average data packet delay

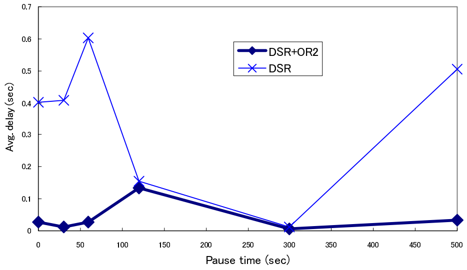


Packet delivery fraction

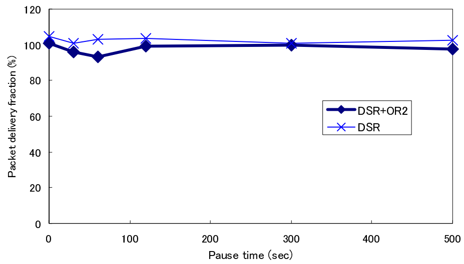
Figure 10: our oriented mobility model



Normalized routing load



Average data packet delay



Packet delivery fraction

Figure 11: our escape mobility model

4.3 Experimental Results

Finally, we show a simple experimental evaluation of our DPS. In our experiment, mobile nodes are Pentium-based laptop computers running FreeBSD 4.4 and equipped with a MELCO IEEE 802.11b wireless network card. We installed DSR and DPS in these nodes and conducted a preliminary experiment: measurement of the latency in re-routing paths.

To observe the overhead associated with shortening of a path, we conducted ten trials of shortening a path with three nodes, Nodes A, B, and C. Node A sends UDP packets continuously to Node C via Node B. For comparison, we measured the round-trip time (RTT) from C to A. The result of measuring the RTT is shown as “Ping” in Figure 12. To create a situation of shortening, we moved Node C near Node B. In this situation, the threshold S_{max} is sufficiently high value so that shortening the active route does not fail. We measured the duration from the time at which Node C sends the first DPS_REQ message to the time at which Node C receives data packets directly from Node A. As observed in Figure 12, the overhead incurred with the exchange of DPS messages is sufficiently small; it is less than 5 ms. This experimental result is sufficiently applicable in the realistic environment with pedestrian or slow vehicle speed in our daily life.

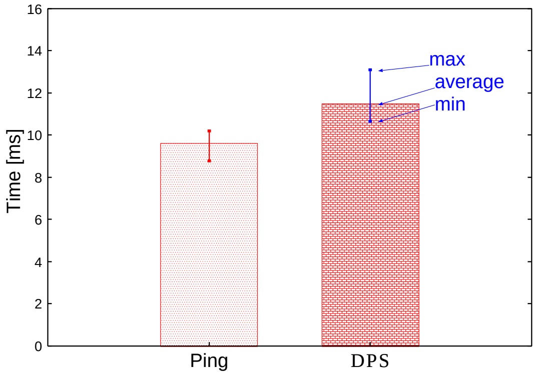


Figure 12: Network delay (ping) and latency time of DPS to shorten an active route over two-hop route

5 Discussion

In this work, we use SSNR as a metric of proximity. However, there are other alternatives. We can determine whether or not two nodes are moving apart by monitoring the differential values of SSNR and rate of received packets. If these differential values decrease, we can know that the distance between these nodes is increasing. These values can be obtained even with TCP ACK packets from a downstream. When we hypothesis $S_{(R_{uf}(K)K)} = S_{(KR_{uf}(K))}$, we can estimate the downstream link quality of the route sending TCP data packets by the SSNR of TCP ACK packets from $R_{df}(k)$.

For a reliable transport protocol like TCP, it is a smarter to control the data transmission rate adaptively to the degradation of link quality and the possibility of link disconnection. Thus, by using the history of SSNR, we presumably identify and estimate the decreasing of link quality and link disconnection so that we avoid making the link wastefully congested and against numerous packet loss. Additionally, we can change the more stable route in advance. However, to extract the information about such link quality, an extra overhead of processing is posed at each node.

6 Conclusion and Future Work

We have proposed DPS, an adaptive route path tuning algorithms for mobile ad hoc networks. Since most conventional routing protocols accommodate topology changes only when an active route is disconnected, it is not adaptive to node mobility. Our approach is more adaptive to our conventional node mobility (e.g., pedestrian and slow vehicle in campus computing) by using the wireless link quality value: SSNR. DPS realizes to automatically shorten the active route by using the notion of *proximity*. This mechanism achieves the significant reduction of the route path delay. As a result, it is highly effective for network capacity and power consumption in such a limited resource environment as multi-hop wireless networks. Also, reducing a route path delay is highly important for in particular TCP flows. In DPS, each node individually monitors local link quality only when receiving packets and make local decisions in a decentralized manner. There is no need for exchanging periodic control information such as HELLO messages.

As a case study of DPS scheme, we extended conventional DSR and AODV to accommodate the dynamic path shortening mechanism. Since these two protocols are on-demand routing approaches, DPS efficiently enhances the performance of them due to negligible routing overhead of DPS. The experimental results have shown that DPS is effective in enhancing TCP throughput and reducing end-to-end delay for all relevant flows. This scheme achieved a significant reduction of a path delay while the links are still active. Similarly, simulation studies for several scenarios of high node mobility and traffic flows have revealed that adding dynamic path shortening to DSR and AODV significantly reduces the number of routing packets and the packet latency in most cases. Most of ad hoc routing protocol proposals are evaluated through simulation, although a few of them through real environment. We have implemented DPS as an extension to DSR on FreeBSD OS and experimented in some simple scenarios. We believe that evaluations of ad hoc routing protocols are incomplete without real-world experiments.

In most previous simulation research of ad hoc networks, simulated nodes move according to the random way-point model. This model generates the random movement of nodes based on a random destination and a speed distributed uniformly between 0 and some maximum speed. In order to investigate how DPS scheme perform in more *realistic* and intensive movement pattern, we proposed the two node mobility models: the “random oriented model” and “random escape model”. These model generate the movement patterns assuming people pursuing something (e.g., *dreams*, *love*, *power*, etc) and escaping from something (e.g., *a fire*, *disaster*, *due date*, etc).

We are currently working on examining much more effects of our multi-hop shortening schemes with tapping network interfaces while taking into account its influences to power consumption. Ideally, we believe that dynamic multi-hop shortening schemes without promiscuous listening is the better approach for power saving.

We also need to analyze the decision of the SSNR threshold S_{max} value and the comparing frequency because these factors have important impact on its effectiveness of DPS. Although we consider devices operating in the ISM bands (such as Lucent WaveLANs or IEEE 802.11b wireless LAN cards) in this thesis, we hope to choose a flexible signal threshold value since signal power of received packets is dependent on what wireless devices are used. In addition, we are now working to extend the ns2 network simulator to accurately model the physical layer behavior of the IEEE 802.11b and 802.11a wireless LAN standard [16], so that we can simulate DPS in the environments where the wireless raw bandwidth is from 11 Mbps (802.11b) to 54 Mbps (802.11a).

We also plan to make the novel experimental routing protocol based on signal power for the blending network. While signal stability based adaptive routing (SSA) protocol already was proposed [7], our planning routing protocol is based on signal strength and supplemental location information using the Global Positioning System (GPS) in the real world.

References

[1] B. Bellur, R. Ogier, F. Templin, and M. Lewis. Topology Broadcast based on Reverse Path Forwarding (TBRPF). IETF Internet-Draft [Work in Progress], January 2002.

[2] Vaduvur Bharghavan, Alan Demers, Scott Schenker, and Lixia Zhang. MACAW: A media access protocol for wireless LAN's. In *Proceedings of ACM SIGCOMM'94*, pages 212–225, August 1994.

[3] Josh Broch, David B. Johnson, and David A. Maltz. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. IETF Internet-Draft [Work in Progress], March 2001.

[4] 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.

[5] Compaq, Inc. iPAQ Handhelds. http://www.compaq.com/products/iPAQ/, 2002.

[6] 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.

[7] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi. Signal Stability-Based Adaptive Routing (SSA) for Ad Hoc Mobile Networks. *IEEE Personal Communications Magazine*, 4(1):36–45, February 1997.

[8] Z. J. Hass and M. R. Pearlman. The Performance of Query Control Schemes for the Zone Routing Protocol. In *Proceedings of ACM SIGCOMM'98 Conference*, pages 167–177, September 1998.

[9] Philippe Jacquet, Paul Muhlethaler, Amir Qayyum, et al. Optimized Link State Routing Protocol (OLSR). IETF Internet-Draft [Work in Progress], March 2001.

[10] 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.

[11] 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.

[12] Orinoco, Inc. The WaveLAN Home Page. http://www.wavelan.com, 1998.

[13] 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.

[14] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das. Ad Hoc On Demand Distance Vector (AODV) Routing. IETF Internet-Draft [Work in Progress], January 2002.

[15] 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.

[16] Mineo Takai, Jay Martin, and Rajive Bagrodia. Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks. In *Proceedings of ACM Mobihoc 2001*, October 2001.

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

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

[19] Bruce Tuch. Development of WaveLAN, an ISM band wireless LAN. *AT&T Technical Journal*, 72(4):27–37, 1993.