モバイルコンピューティングと
ワイヤレス通信 18－28
高度交通システム 6－28
（2001．9．7）

アドホックネットワークにおけるキャッシュ情報を有効利用した AODV 拡張ルーティングプロトコル

WANG Wooi-Ghee　原 隆浩　塚本 昌彦　西尾 章治郎
大阪大学大学院工学研究科情報システム工学専攻
E-mail: {wooighee,hara,tuka,nishio}@ise.eng.osaka-u.ac.jp

近年，無線通信を用いて，移動体のみで暫定的にネットワークを構築するアドホックネットワークに対する注目が高まっている．アドホックネットワークでは，各移動体がパケットを中継することにより，無線では直接通信できない移動体間の通信を実現している．従来の一般的なアドホックネットワークルーティングプロトコルでは，ネットワーク内のすべての移動体が頻繁に移動することを想定して，通信を行うときにだけルートの発見を行っている．しかし，このようなルーティングプロトコルでは，移動体間のルートが変化していない場合でも，キャッシュされている情報が短時間でタイムアウトするため，ルート発見のためのフラッディングにより大きな遅延が生じる可能性がある．そこで，本研究では，アドホックネットワークにおけるルート発見による遅延の短縮を目的として，既存の AODV ルーティングプロトコルを，安定したルートのキャッシュ情報を長時間的保持するように拡張する．さらに，拡張プロトコルの有効性をシミュレーション実験によって検証する．

An AODV Compatible Routing Protocol Using Cache Information in Ad-hoc Networks

Wooi-Ghee WANG　Takahiro HARA　Masahiko TSUKAMOTO　Shojiro NISHIO
Dept. of Information Systems Eng., Graduate School of Engineering, Osaka University

Recently, there has been an increasing interest in ad-hoc networks, which are dynamically constructed by collections of mobile hosts without using any existing network infrastructure or centralized administration. Each mobile host plays a role of a router and relays packets for multihop network communications. A recent trend in ad-hoc network routing is the reactive on-demand philosophy where routes are established only when required. Most of the protocols in this category, however, show a long latency in route discovery since the cached routing information will be invalidated even if it is still effective. In this paper, we propose a scheme which is compatible to the existing Ad-hoc On-Demand Distance Vector (AODV) protocol and prevents the cached routing information from becoming invalidated without using any extra control message. We also verify the effectiveness of the newly proposed scheme by simulation experiments.

1 Introduction

Ad-hoc networking [4] has emerged as one of the most focused research areas in the field of wireless networking and mobile computing. Ad-hoc networks consist of only mobile hosts and can be constructed without any wired base station or infrastructure support. In ad-hoc networks, routes are mainly multihop because of the limited radio propagation range, and topology may frequently change since each host moves freely. Therefore, routing is an integral part of ad-hoc communications, and has received much interest from researchers. Recently, many new routing protocols have been proposed for ad-hoc networks [3, 5, 6, 7, 8].

Conventional routing protocols developed for traditional wired LANs/WANs may be used for routing in ad-hoc networks by treating each mobile host as a router. Such algorithms broadly come under the category of *proactive* algorithms [6, 7] since routing information is disseminated among all the nodes in the network throughout the network operating time. Thus, the proactive algorithms provide the routing information instantly when a mobile host needs to send data packets. However, the flip side for such protocols is that the excessive routing overhead transmitted is periodic in nature and lacks consideration for the network's mobility. Proactive algorithms perform excellently in packet latency, especially when most of the nodes are in low mobility mode, but in high mobility mode these algorithms perform poorly especially in routing overhead .

Recently, a new style of routing proposed for ad-hoc networks called *reactive* or *on-demand* routing [3, 8] has been gaining wide attention. Unlike conventional proactive routing protocols, each node in on-demand routing does not need periodic route table update exchanges and does not have a full topological view of the network. Network hosts cache route table entries only to destinations that they communicate with. The Ad-hoc On-demand Distance Vector (AODV) protocol [8] is one of the on-demand routing algorithms that is receiving the most attention recently and has been extensively analyzed [1, 9]. One of the biggest advantages of these protocols is that they produce significantly less routing overhead comparing to proactive routing protocols, especially when most of the nodes are in high mobility mode. However, such reactive routing protocols tend to provide a higher packet latency than proactive protocols in low mobility mode because they start attempting to discover a route to the destination only when they finally want to send data packets. Due to the on-demand nature of the protocols, the long delay of end-to-end data transferring can be costly when the network traffic requires real time delivery (voice, for instance). Likewise, if the session

is a best effort, TCP connection, long delay may lead to slow start, timeout, and throughput degradation.

If we inspect the real world mobile model, there is hardly the case where all nodes are in high mobility mode or all nodes are in low mobility mode at the same time. In fact, a dynamic mobile model, in which, parts of its mobile nodes are in high mobility mode and parts of its mobile nodes are in low or static mode, is more likely to correctly reflect a real world mobile model. Taking this into consideration, we believe that a dynamic combination of proactive and reactive techniques which apply adaptively to parts of the mobile model, is more likely to perform better than either approach alone.

Instead of proposing an entirely new protocol which is incompatible with any existing protocols, we analyzed the well-known AODV routing protocol and propose to apply proactive techniques to extensively utilize the effective cached routing information with the reactive protocol in order to make it more suitable for real mobile environments. Our proposed protocol is highly AODV compatible in that it tolerates the existence of mobile hosts which only accept AODV in the network. This enables the coexistence of both protocols at the same time and enhances its robustness of deployment compared to other newly proposed protocols which lack compatibility with AODV. The "compatibility to the existing protocol" concept is a brand new idea that we have proposed in the ad-hoc routing research field. In this paper, we also compare our proposed protocol with the existing AODV protocol by simulation experiments. We show that, compared to AODV, our proposed protocol performs better in packet latencies and is highly compatible to AODV. Both proactive and on-demand techniques are applied dynamically to improve the use of cached routing information extensively.

The rest of the paper is organized as follows. Section 2 describes the AODV protocol. Section 3 describes our proposed protocol. Section 4 provides performance evaluations of the proposed protocol. Section 5 concludes the paper.

2 AODV Protocol

In this section, we give an overview of the well-known on-demand AODV scheme.

When a source needs to initiate a data session to a destination but does not have any cached route information, it searches a route by flooding a route-request (RREQ) packet. To prevent unnecessary broadcasts of RREQs, the source node uses the *expanding ring search* technique as an optimization. In the expanding ring search, increasingly larger neighborhoods are searched to find the destination. The search is controlled by the time-to-live (TTL) field in the IP header of the RREQ packets. Each RREQ packet has a unique identifier so that nodes can detect and drop duplicate packets. An intermediate node, upon receiving a non-duplicate RREQ, records the previous hop and the source node information (source sequence number) in its routing table. Then it rebroadcasts the RREQ packet. Figure 1(a) illustrates the propagation of RREQs across the network from source node S to destination node D and the cached routing table of the corresponding node upon receiving the RREQ packet.

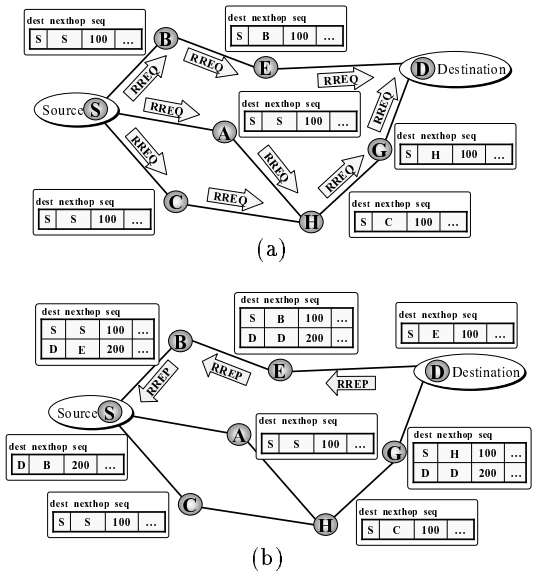

Figure 1: AODV protocol

An intermediate node sends back a route-reply (RREP) packet to the source if it has cached route information to the destination in its routing table. Otherwise, the destination node sends a RREP via the selected route when it receives the first RREQ or subsequent RREQs that traversed a better route (for instance, a fresher or shorter route) than the previously replied route. An intermediate node, upon receiving a non-duplicate RREP, records the previous hop and the destination node information (destination sequence number) in its route table. Then it unicasts the RREP packet to the next hop node leading back towards the source. Figure 1(b) illustrates the unicasting of RREPs from destination node D to source node S. When RREP reaches source node S, the route to destination node D is discovered and data packets from source S will be routed along the route ⟨S-B-E-D⟩.

Every routing table entry maintains a route expiration time which indicates the time until which the route is valid. Each time that route is used to forward a data packet, its expiration time is updated to be the current time plus the *active_route_timeout*. A routing table entry is invalidated if it is not used by the expiration time. AODV uses an active neighbor node list for each routing entry to keep track of the neighbors that are using the entry to route data packets. These nodes are notified with route-error (RERR) packets when the link to the next hop node is broken. Each such neighbor node, in turn, forwards the RERR to its own list of active neighbors, thus invalidating all the routes using the broken link.

3 Protocol Concept

In this section, we present the operation details of our proposed scheme. We name our proposed scheme the *AODV Compatible Stability Based Routing Protocol* (AODV-SB). Since the purpose of our study is to construct a protocol compatible to the existing AODV protocol, our protocol description is based on AODV. Our scheme does not require any modification to the AODV's route discovery and route maintenance mechanisms.

3.1 Link Stability

In this subsection, we introduce the new idea of link stability and utilize this link stability in the AODV protocol. In ad-hoc networks, two mobile hosts can be connected directly to each other by a radio link. This link is disconnected when they move further away from each other, thus making the distance between them longer than the possible communication range. In AODV, a node offers connectivity information by broadcasting local Hello messages. By inspecting the pattern of arriving Hello messages between two connected neighboring mobile hosts, the stability of the link can be estimated. Here, we propose a few functions to decide the stability of a link between two nodes. Using current time as t, we define A_{ij} as a function that represents at least a Hello message from mobile host M_j reaches mobile host M_i within the last $(t-T)$ period of time. T represents the Hello message period.

$$A_{ij}(t) = \begin{cases} 1: & \text{Hello message(s) arrive(s)} \\ & \text{within } (t-T) \\ 0: & \text{Otherwise} \end{cases} \tag{1}$$

Then, $B^n_{ij}(t)$ is defined as a function that represents the total number of Hello messages arrived within $(t-nT)$ period of time. The value n is a predecided value where it indicates the number of allowed losses of Hello message. This is essential because the Hello message may get lost in radio transmission or fail to arrive in time due to collision or contention problems.

$$B^n_{ij}(t) = \sum_{k=0}^{n-1} A_{ij}(t-kT) \tag{2}$$

The link connection state of two mobile hosts M_i and M_j is represented as $C^n_{ij}(t)$. $C^n_{ij}(t)$ is calculated by using $A_{ij}(t)$ and $B^n_{ij}(t)$ as follows:

$$C^n_{ij}(t) = \begin{cases} C^n_{ij}(t-T)+1 & : B^n_{ij}(t) > 0, A_{ij}(t) = 1 \\ C^n_{ij}(t-T) & : B^n_{ij}(t) > 0, A_{ij}(t) = 0 \\ 0 & : B^n_{ij}(t) = 0 \end{cases} \tag{3}$$

If the Hello messages from mobile host M_j arrives continuously, $A_{ij}(t)$ will equal to 1 and B^n_{ij} will have a non-zero positive value, thus $C^n_{ij}(t)$ will increase continuously too. If the Hello message fails to arrive within $(t-T)$ but there is at least one Hello message that arrives within $(t-nT)$, then $C^n_{ij}(t)$ will be sustained as $C^n_{ij}(t-T)$. If no Hello message arrives within $(t-nT)$, $C^n_{ij}(t)$ will be set to the value zero.

The value of $C^n_{ij}(t)$ indicates the continuous arrival of Hello messages which infer how frequently mobile hosts M_i and M_j are connected to each other by a radio link. We infer that a radio link with a larger $C^n_{ij}(t)$ is more stable than one with a smaller $C^n_{ij}(t)$. If $C^n_{ij}(t)$ increases above a threshold value S_{th}, then the link between mobile hosts M_i and M_j is regarded as a *stable link*. The total number of stable links that mobile host M_i has at time t, is calculated as $L_i(t)$. A mobile host with $L_i(t)$ larger than a threshold value K_{th} is considered to be a *stable node*.

Using the concept of these stability functions, we can evaluate the stability of radio links among mobile hosts. By propagating the effective stable link information only among the stable hosts , we may construct *temporal perennial links* adaptively over the static and the low mobility part of the ad-hoc networks. The constructed temporal perennial links will improve the performance of conventional on-demand routing protocol by fully utilizing its cached routing information. At the same time, by ignoring the unstable radio links, unnecessary traffic to propagate the unsustainable routing information is eliminated.

3.2 Temporal Perennial Link Construction

In our AODV-SB scheme, we propose two strategies to construct temporal perennial links without introducing any new control message type. We modify the AODV protocol to achieve our goals.

3.2.1 Strategy 1: AODV-SB-RREQ

The first strategy is the AODV-SB-RREQ strategy. We utilize the AODV's RREQ packet to propagate the effective cached routing information among nodes with stable links. The AODV-SB-RREQ strategy consists of three steps.

1. Calculating the link stability
2. Generating the *pseudo-route-request* (PRREQ) packet
3. Forwarding the *pseudo-route-request* (PRREQ) packet

At the first step, a mobile host calculates the link stability of its neighbors by using the functions that we described in the previous subsection.

After that, we go to the second step where mobile host M_i with $L_i(t) \geq K_{th}$ constructs the temporal perennial link every *propagate_check_interval* period of time. In order to achieve the goal of compatibility to the existing protocol, we introduce a new concept in the on-demand routing research field. We name the concept as the "pseudo-control-packet concept". This is a concept where we modify the existing control packet and disseminate the modified "fake control packet" into the network. Mobile hosts which only accept the existing protocol, will be "deceived" by the fake control packet and process the packet as if it is a real

AODV control packet. In order to accomplish the goal mentioned above, we have made some modifications as described below into the AODV's RREQ packet to make the fake control packet looks real to mobile hosts which only accept AODV packet. We name the modified RREQ packet as the *pseudo-route-request* (PRREQ) packet.

- *Indication Flag*: In order to distinguish between the real RREQ and the fake RREQ, that is PRREQ packet, we need to insert an *indication flag* into the modified RREQ packet where the normal AODV mobile hosts will not notice the indication flag. Since the 13 byte Reserved field in RREQ is always ignored on reception by the AODV mobile hosts, we utilize 1 byte in this field as an *indication flag* of the fake RREQ packet. In this way, only mobile hosts which install our protocol will look for the indication flag and distinguish between the fake and the real RREQ packet.
- *Fictitious_ip_address*: RREQ is originally a packet designed to discover the destination node by constructing reverse paths towards the source node so that the destination node or the intermediate node with valid route information will be able to send back the RREP packet along the reverse paths and setup the forward path toward the destination node. Since the purpose of the fake RREQ is to construct temporal perennial link towards the stable node, we need to find a way to activate the reverse path construction mechanism and at the same time to suppress the forward path setup mechanism. To achieve this goal, we allocate a *fictitious_ip_address* into the Destination IP Address field and fill the stable node's IP address into the Source IP Address field in RREQ. The *fictitious_ip_address* is a made-up IP address that will reach no mobile host in the whole network. This avoids any generation of unnecessary RREP packet to the PRREQ packet. In this way, normal AODV nodes are deceived by the fake RREQ packet and construct only the reverse path towards the stable node after receiving the PRREQ packet which is actually requesting for an unattainable route.
- *Fictitious_sequence_no*: Since the *fictitious_ip_address* will reach no host in the mobile network, the sequence number related to the address will not be copied into any mobile host's cached routing table. In AODV, the Destination Sequence Number field of RREQ is filled with the last sequence number received in the past by the source for any route towards the destination. Thus, a make-up sequence number, *fictitious_sequence_no* is used to fill up this field.
- *Aodv-sb-rreq-ttl*: Due to the reason that the PRREQ packet is not to search for a real destination node, it does not have to follow the expanding ring search technique in deciding the range of the packet propagation. In AODV, the TTL field in the RREQ's IP header is always filled with the value following the expanding ring search mechanism where it is 1 initially, and increases gradually. However, in the PRREQ packet, we set the TTL field in the PRREQ's IP header to the *aodv-sb-rreq-ttl* value. When this *aodv-sb-rreq-ttl* value is larger than 1, then the PRREQ packet will be allowed to be propagated and penetrate its route information a few hops away from the last stable node.

The third step is to forward PRREQ packets. When a mobile host M_i receives a PRREQ packet, it first checks whether it has received a PRREQ packet with the same source IP address and broadcast ID. If such a packet has been received, the node silently discards the newly received packet.

If the received PRREQ packet is not discarded, and $L_i(t) \geq K_{th}$, then the mobile host performs the following procedures:

- Processing the PRREQ packet: Mobile hosts process the PRREQ packet by following the normal AODV procedure. Source sequence number from PRREQ is copied to the corresponding destination sequence number and the next hop in the cached routing table becomes the node that have broadcasted the PRREQ packet.
- Rebroadcasting the PRREQ packet: In AODV, the TTL field in the outgoing IP header is always decreased by one. Since we intend to penetrate the stable node routing information a few hops away from the last stable node, the TTL field is freezed at the *aodv-sb-rreq-ttl* value, before it is rebroadcasted to its neighboring nodes. Then, the PRREQ packet is rebroadcasted from the mobile host using its own IP address in the outgoing PRREQ packet.

On the other hand, if the mobile host M_i has $L_i(t) < K_{th}$, then it rebroadcasts the PRREQ packet following the AODV procedure until its TTL value becomes zero.

Figure 2 is an example showing how AODV-SB-RREQ will improve the delay of route discovery. As shown in Figure 2(a), temporal perennial links are constructed over the stable links (E-D, D-G, G-H) by using the PRREQ packets. Since PRREQ can penetrate its routing information a few hops from last stable node, routing information is disseminated until reaching nodes A, B and C. Figure 2(b) illustrates that source node S initiates a route discovery mechanism towards destination node D by broadcasting the RREQ packets. Since intermediate nodes A, B and C have valid cached routing information, they respond to the RREQ packets by generating RREP packets as shown in Figure2(c). A route to the destination node D is discovered instantly and the data packet will be routed along the route ⟨S-B-E-D⟩.

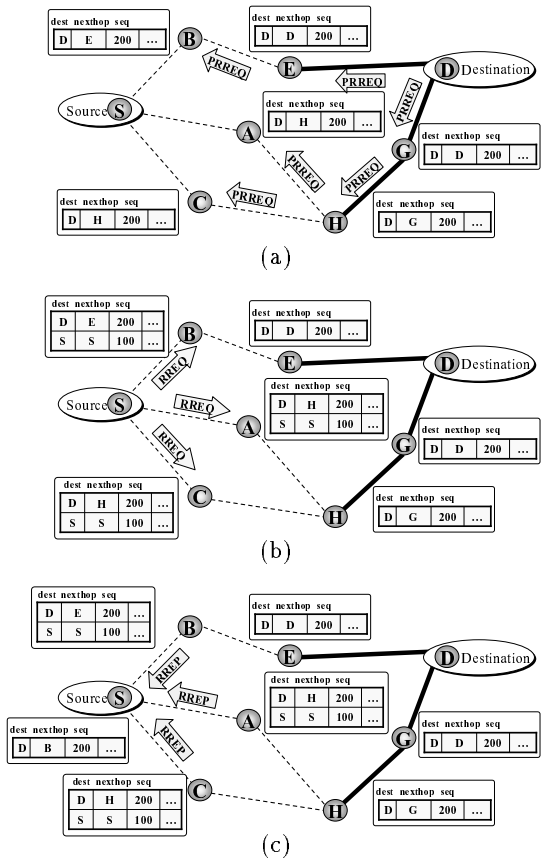

Figure 2: AODV-SB-RREQ Strategy

3.2.2 Strategy 2: AODV-SB-RREP

The second strategy is the AODV-SB-RREP strategy. Since the RREP packet is originally a unicast packet which has features differing from the multicast RREQ packet, the modification will be slightly different. The AODV-SB-RREP strategy consists of three steps.

1. Calculating the link stability
2. Generating the *pseudo-route-reply* (PRREP) packet
3. Forwarding the *pseudo-route-reply* (PRREP) packet

In the first step, the total number of stable links for a mobile host M_i will be calculated as $L_i(t)$. Then, the stable host will generate the PRREP packet in the second step. Mobile host M_i with $L_i(t) \geq K_{th}$ generates a modified RREP packet every *propagate_check_interval* period of time without the arrival of a RREQ packet. We call the modified RREP packet as the *pseudo-route-reply* (PRREP) packet. The PRREP packet is a fake RREP packet with the modifications as described below.

- *Indication Flag*: In order to distinguish between the real RREP and the fake RREP, that is PRREP packet, we need to insert an *indication flag* into the modified RREP packet where the normal AODV mobile hosts will not notice the indication flag. Since the 9 byte Reserved field in RREP is always ignored on reception by the AODV mobile hosts, we utilize 1 byte in this field as an *indication flag* of the fake RREP packet. In this way, only mobile hosts which install our protocol will look for the indication flag and distinguish between the fake and the real RREP packet.
- *Fictitious_ip_address*: The purpose of the PRREP packet is not to answer any RREQ packet, but to activate the cached information towards the destination node. In AODV, the Source IP Address field in RREP is filled with the IP address of the source node which issued the RREQ for which the route is supplied. To achieve our goal, we use the *fictitious_ip_address* in the Source IP Address field instead. This prevents the PRREP packet from terminating at a certain mobile host in the network.
- *Aodv-sb-rrep-lifetime*: Since the attribute of the PRREP packet is to construct the temporal perennial links, we set the Lifetime field of RREP to the *aodv-sb-rrep-lifetime* value. In conventional AODV, the Lifetime field of RREP is copied from the mobile host's default *my_route_timeout* value, to indicate the time for which nodes receiving the RREP consider the route to be valid.

The third step is to forward PRREP packets. When a mobile host M_i receives a non-duplicate PRREP packet, and it has $L_i(t) \geq K_{th}$, then it performs the following procedures:

- Processing the PRREP Packet: The mobile host first compares the destination sequence number in the received PRREP packet with its own copy of the destination sequence number for the destination IP address. The forward route for this destination is created or updated if the destination sequence numbers is greater than the node's copy of the destination sequence number. The next hop to be copied in the routing table is the IP address of the node from which the PRREP is received.
- Rebroadcasting the PRREP Packet: The TTL field will be fixed at the *aodv-sb-rrep-ttl* value. The PRREP packet is rebroadcasted to its neighboring nodes using its own IP address in the outgoing PRREP packet.

On the other hand, if $L_i(t)$ is less than K_{th}, then it stops from rebroadcasting the PRREP packet.

Figure 3 is an example showing how AODV-SB-RREP will improve the delay of route discovery. As shown in Figure 3(a), temporal perennial links are constructed over the stable links (E-D, D-G, G-H) by using the PRREP packets. Since AODV-SB-RREP is a conservative strategy, it always refrains from disseminating its routing information beyond the stable nodes. Routing information is disseminated only until reaching nodes E and H. Figure 3(b) illustrates that source node

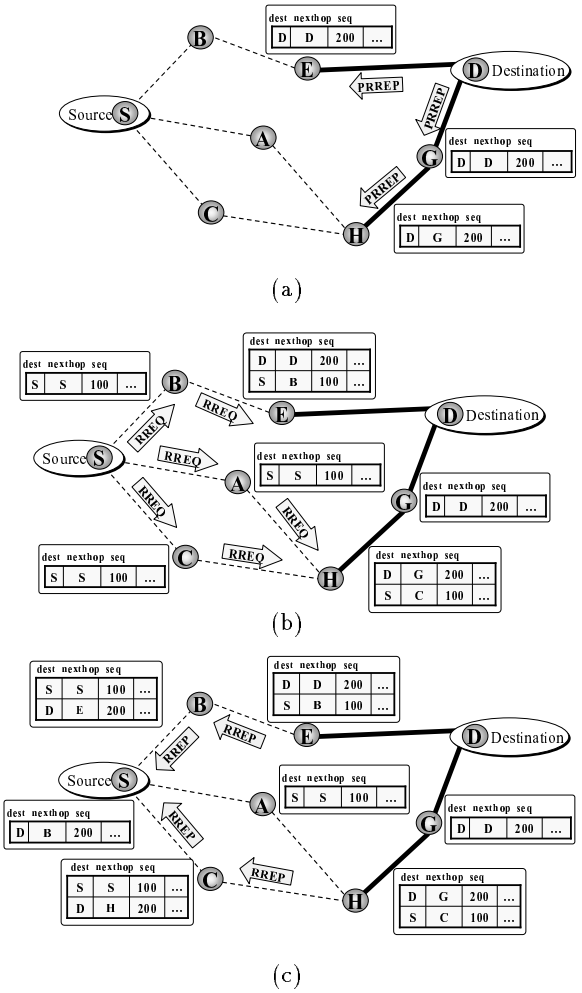

Figure 3: AODV-SB-RREP Strategy

S initiates a route discovery mechanism towards the destination node D by broadcasting the RREQ packets. Since intermediate nodes E and H have valid cached routing information, they respond to the RREQ packets by generating RREP packets as shown in Figure3(c). A route to the destination node D is discovered instantly and the data packet will be routed along the route ⟨S-B-E-D⟩.

4 Performance Evaluation

To evaluate the performance improvements made by our proposed scheme, we compare the simulation results of the AODV protocol with and without applying our protocol.

4.1 Simulation

Our simulation models a network of 50 mobile hosts placed randomly within a 1500 m × 300 m area. The radio propagation range for each node is 250 m. Each run has an execution simulation time of 300 seconds.

To simulate constant bit rate sources, we developed a traffic generator, where 10 traffic sources are maintained throughout the simulation time. The sources and the destinations are randomly

Table 1: Parameter configuration

Parameter	Value
n (allowed Hello loss)	2
S_{th} (stable link threshold)	4
K_{th} (stable node threshold)	1
aodv-sb-rreq-ttl	3
aodv-sb-rrep-lifetime	3 (seconds)

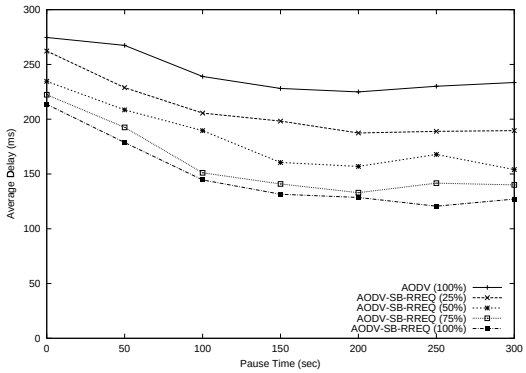

Figure 4: AODV-SB-RREQ End-to-end Delay

selected with uniform probabilities. The size of the data payload is 512 bytes. As for the mobility model, we use the random waypoint mobility model [1]. Each node randomly selects a position, and moves toward that location with a randomly chosen speed uniformly distributed between 0 and 20 m/seconds. Once it reaches that position, it becomes stationary for a predefined pause time. After that pause time, it selects another position and repeats the process. We vary the pause time to simulate different mobility degrees. We assume that the local link connectivity is detected by using a MAC layer beacon message. Each result point in the graph represents an average of at least five runs with identical traffic models, but different randomly generated mobility scenarios.

A send buffer of 64 packets is maintained throughout the simulation time. A node buffers all data packets waiting for a route, e.g., packets for which route discovery has started, but no reply has arrived yet. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more than 30 seconds. The interface queue is FIFO, with a maximum size of 64. Routing packets are given higher priority than data packets in the interface queue.

In order to show the significant differences of features in AODV-SB-RREQ and AODV-SB-RREP, we constructed the parameter configuration as shown in Table 1. The *propagate-check-interval* value for AODV-SB-RREQ is 5 seconds and 10 seconds for AODV-SB-RREP.

4.2 Results

Figure 4 and 5 show the average end-to-end delay of data packets for AODV, AODV-SB-RREQ and AODV-SB-RREP. The end-to-end delay of data packets includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, propagation and transfer

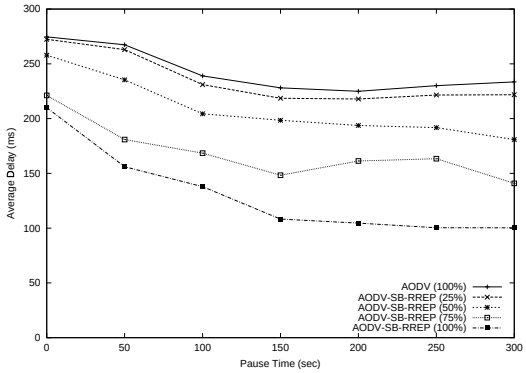

Figure 5: AODV-SB-RREP End-to-end Delay

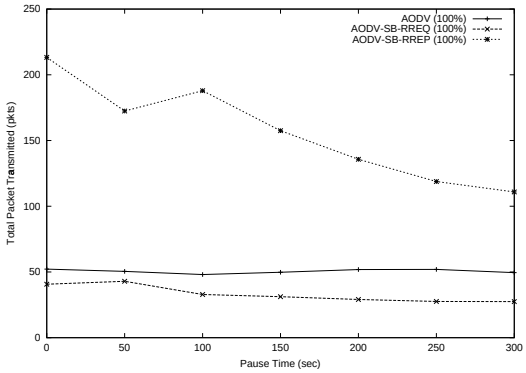

Figure 7: Total RREP Packet Transmitted

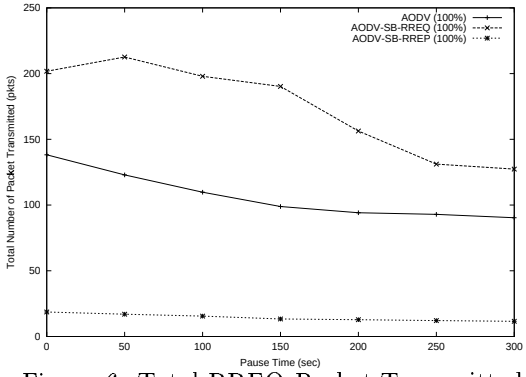

Figure 6: Total RREQ Packet Transmitted

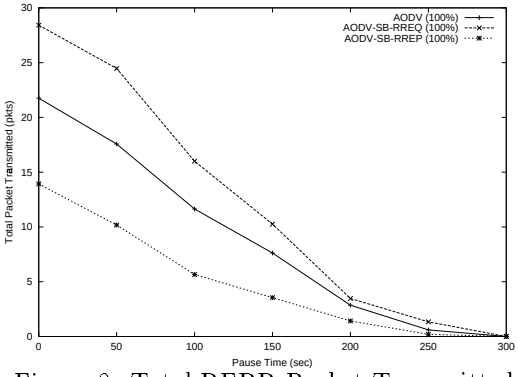

Figure 8: Total RERR Packet Transmitted

times. *AODV(100%)* indicates that all of the mobile hosts in the network apply the AODV protocol. Whereas, *AODV-SB-RREQ/RREP(x%)* specifies that x % of the mobile hosts in the network apply the AODV-SB-RREQ/RREP scheme and $(100 - x)$ % of the the mobile hosts apply the AODV scheme.

We can see that both of our strategies improve the average end-to-end delay of data packets performance compared to AODV. As the mobility decreases (i.e., pause time gets longer), the performance gain becomes more significant. AODV-SB-RREQ(100%) yields less average packet latencies, ranging from 22% to 45% less, compared to AODV(100%) as the mobility decreases. Whereas, AODV-SB-RREP(100%) produces less average packet latencies, ranging from 23% to 57% less, compared to AODV(100%) as the mobility decreases. The reason is that in low mobility model, more stable links are available in the network to be constructed as the temporal perennial links. Thus, the existing of the temporal perennial link speeds up route discovery process. Also, in the presence of route breaks, the AODV-SB protocol is able to deliver data packets to the destination faster than AODV since the local repair mechanism can be carried out swiftly.

Since AODV-SB is highly AODV compatible, e.g. even if mobile hosts which only allow AODV exist in the network, the scheme will still be well operated. We investigate the effect where mobile hosts with AODV and mobile hosts with AODV-SB are mixed in the network. As we can observe from Figure 4 and 5, both strategies produce significantly less average end-to-end delay as the percentage of node with AODV-SB increases from 25% to 75% in the network. The existence of more AODV-SB nodes in the network, enable the construction of temporal perennial links to be carried out more efficiently.

Nevertheless, AODV-SB-RREP(25%) only decreases the average end-to-end delay from 1% to 5% compared to AODV(100%). On the other hand, AODV-SB-RREQ(25%) decreases the average end-to-end delay significantly from 5% to 19%. This shows one of the differences between AODV-SB-RREQ and AODV-SB-RREP. AODV-SB-RREQ allows the propagation and penetration of PRREQ packet information a few hops away from the last stable host whereas AODV-SB-RREP cannot do this since the RREP packet is originally a unicast packet. Thus, we can infer that AODV-SB-RREQ, as an aggressive strategy, performs better where there are a high percentage of AODV nodes in the network. On the other hand, AODV-SB-RREP, as a conservative strategy, performs better where there are a low percentage of AODV nodes in the network.

Average total numbers of RREQ, RREP, and RERR packets transmitted per route in the network are presented in Figure 6, 7 and 8, respectively. Each hop-wise transmission of a routing packet is counted as one transmission. The PRREQ transmission is counted as one RREQ transmission and the PRREP transmission is counted as one RREP transmission. As expected, AODV-SB-RREQ transmits more RREQ control packets compared to AODV as shown in Figure 6. However, interestingly AODV-SB-RREQ reduces the

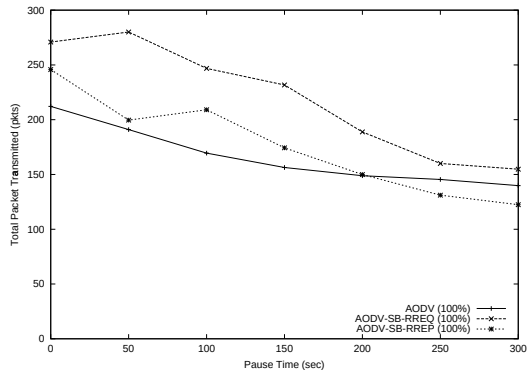

Figure 9: Total Routing Packet Transmitted

transmission of RREP throughout the network as shown in Figure 7. Likewise, AODV-SB-RREP transmits more RREP control packets and less RREQ control packets compared to AODV. The reason is that the construction of temporal perennial links shortens the hop distance for a mobile host to discover its destination host. From Figure 8, AODV-SB-RREQ generates more RERR packets than AODV because of its aggressive features in propagating routing information. Whereas, AODV-SB-RREP generates the least RERR packets since it is conservative in route information propagation.

Figure 9 gives the average total number of RREQ, RREP and RERR packets transmitted per route. AODV-SB-RREQ transmits 27% to 10% more routing packets than AODV. We can learn from this result that we need to sacrifice some routing overhead in order to improve data packet latency and protocol effectiveness. Whereas AODV-SB-RREP produces about 15% more routing packets in high mobility model and 12% less routing packets in low mobility model. The reason of the reduce of total routing packets in AODV-SB-RREP is that we may reduce the dissemination of PRREP by taking a larger value for the propagation period and timeout period of the temporal perennial links. We may infer from these results that, compared to AODV, AODV-SB provides significantly lower latencies while producing a moderate increase of control packets.

5 Conclusions

In this paper, we have proposed a new routing protocol, AODV-SB which combines the proactive and reactive techniques dynamically in order to improve the use of cached routing information extensively. The proactive technique is applied into parts of the network where mobility is relatively low and reactive technique is applied into the network where mobility is relatively high. By propagating the effective stable link information only among the stable hosts, we construct temporal perennial links adaptively over the static and the low mobility part of the networks. By ignoring the unstable radio links, unnecessary traffic to propagate the unsustainable routing information is eliminated.

We have also proposed the compatibility concept which enhances the robustness of deployment and tolerates the existence of mobile nodes which only accept the existing AODV protocol. In order to realize AODV compatibility, we introduced the pseudo-control-packet idea. AODV-SB-RREQ is an aggressive strategy that performs impressively in existing networks with a high percentage of AODV nodes by producing a lower packet latency. Whereas AODV-SB-RREP is a conservative strategy that performs better in existing networks with a low percentage of AODV nodes by giving a lower routing overhead. Simulation results indicate that AODV-SB outperforms the on-demand AODV protocol in packet latency while maintaining a moderate increase of control packet.

As part of our future work, we are considering other methods to reduce the routing overhead furtherly while maintaining low packet latency.

Acknowledgement

This research was supported in part by Grant-in-Aid for Encouragement of Young Scientists numbered 13780330 from Japan Society for the Promotion of Science.

References

[1] Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y-C., and Jetcheva, J.: "A performance comparison of multi-hop wireless ad hoc network routing protocols," *Proc. of ACM Mobicom'98*, pp. 85–97 (Oct. 1998).

[2] Das, S.R., Perkins, C.E., and Royer, E.M.: "Performance comparison of two on-demand routing protocols for ad hoc networks," *Proc. of IEEE Infocom 2000*, pp. 3–12 (Mar. 2000).

[3] Johnson, D.B., and Maltz, D.A.: "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, ed. Imielinski, T., and Korth, H., Chap. 5, pp. 153–181 (1996).

[4] Jubin, J., and Tornow, J.D.: "The DARPA packet radio network protocols," *Proc. of IEEE* , Vol. 75, No. 1, pp. 21–32 (Jan. 1987).

[5] Ko, Y.-B., and Yaidya, N.H.: "Location-aided routing (LAR) in mobile ad hoc networks," *Wireless Networks*, Vol. 6, No. 4, pp. 307–321 (2000).

[6] Nishizawa, M., Hagino, H., Hara, T., Tsukamoto, M., and Nishio, S.: "A routing method using unidirectional link in ad hoc networks," *Proc. of International Conference on Advanced Computing and Communications (ADCOM'99)*, pp. 78–82 (Jan. 1999).

[7] Perkins, C.E., and Bhagwat, P.: "Highly dynamic destination-sequenced distance Vector for mobile computer," *Proc. of ACM SIGCOMM'94*, pp. 234–244 (Aug. 1994).

[8] Perkins, C.E., and Royer, E.M.: "Ad-hoc on-demand distance vector routing," *Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100 (1999).

[9] Royer, E., and Toh, C.K.: "A review of current routing protocols for ad hoc mobile wireless networks," *IEEE Personal Communications Magazine*, Vol. 6, No. 5, pp. 46–55 (Apr. 1999).