國立交通大學

資訊工程系

碩士論文

設計一個適合車間通訊網路的繞徑協定

Designing a Routing Protocol Suitable for Inter-vehicle Communication Networks

研究生：陳璽允

指導教授：王協源　教授

中華民國九十三年六月

Abstract

In the mobile ad-hoc network, Ad hoc On-demand Distance Vector (AODV) is a famous routing protocol for its low overhead. According to the feature of on-demand operation, it experiences high delay and may obtain lower throughput. Here we propose an enhancement of AODV, GPS-support AODV, for the inter-vehicle communication. There are three parts: (i) To verify the deficiencies of AODV and provide solutions for them. (ii) To introduce a new metric for the route construction and further reduce the overhead of AODV. (iii) To avoid the throughput hole and delay from the link break till the route repair, we offer a mechanism called backup route construction to achieve smooth route switch. In contrast with previous works, our simulation cases are generated by vehicles flow simulator rather than random way point; the performance evaluation shows that our proposed GPS-support AODV works better than the original AODV and is more suitable for the inter-vehicle communication.

Acknowledgement

首先感謝各位口試委員撥冗來到交通大學進行口試。

來到交大的這段時間，對於我的人生是個重大的里程碑，我要感謝我的指導教授　王協源副教授，在這兩年內對我在技術和專業上的訓練，讓我在工作能力和態度上有極大的提升，在往後的日子裡我會繼續努力改進。

我要感謝實驗室同學，對於我進行實驗和模擬時時所提供的技術支援。

然後，我要感謝我的家人和朋友，在這段時間內給予我精神和物質上無限制的支持。

Table of contents

List of Figures

1. Introduction

In recent years, the desire for internetworking has transited from the wired to wireless environment. Due to the difference of the essential features, many protocols are proposed for the wireless transmission. There are two types of the wireless architectures. One is the infrastructure network, and the other is the ad hoc network.

The key roles of the infrastructure network are these routers that act like bridges between the wired and wireless networks. These routers are equipped with more than one network interfaces. One of them connected to the internet while some other connected to the access point (AP). The access point is similar to the base station of the mobile phone. It communicates with the roaming mobile node through wireless link. Since these routers are usually fixed and the transmission range of the access point is limited, those mobile nodes which desire for the networking service must remain within the range. Therefore, the coverage of the deployed routes and the APs affects the network availability in the infrastructure architecture. Thanks to the advances in the wireless communication technologies, the wireless network interface card becomes more accessible than before. And the routes with AP are often deployed in the campus, bookstores and coffee shops. Despite the infrastructure device becomes popular, the network service is still not always available for the nodes with high mobility. Hence the other architecture (ad hoc network) appears.

The ad hoc network is substantially different from the infrastructure network. Without the term of the access point, each node of the ad hoc network can act as a router for packet forwarding. According to the hop-to-hop transmission, the protocols are subtler than the infrastructure network. Moreover, these protocols for ad hoc network will suffer from the change of network topology due to the mobility. Therefore two kinds of protocols are proposed: proactive (table-driven) and reactive (on-demand). The proactive protocols make sure that routes to any destination exist if any. For this purpose, these mobile nodes exchange large amount of control packets to maintain the routes to all destinations even when not required; the reactive protocols work in other way. While some traffic needs routes for transmission, these protocols start up and initiate the route construction. Before accomplishing the route construction, those packets desiring for the route remain in the buffer. In contrast with the proactive protocols, the reactive ones suffer from higher end-to-end packet delay. Example of the proactive protocol is Destination Sequenced Distance Vector (DSDV) while examples of reactive protocols are Dynamic Source Routing (DSR) and Ah-Hoc On Demand Distance Vector (AODV). After the Chapter 3, we will focus on the AODV, and figure out the deficiencies of this protocol then improve it.

1.1 Motivation

As mentioned before, AODV is a reactive protocol for the wireless network. Due to the route construction is associated with the packet demand, it will suffer from higher end-to-end delay. On the other hand, it generates the control packets if necessary. So the lower protocol overhead is its main advantage. Furthermore, this advantage over proactive protocols should be appreciated. In the wireless network due to the essential media feature, the high overhead even comparable to the normal traffic will degrade the performance significantly. For this reason, we choose AODV from the reactive protocols for this advantage and perform experiment on it for further improvement. In the numerous experiments, we observe some defects of AODV and AODV may suffer in some situations. In essence, the end-to-end packet delay during the first route construction is inevitable. But the subsequent route reconstructions work badly in these experiments. Originally the route reconstruction is triggered to handle disconnected routes. Without any additional assistance, AODV takes long time to detect the disconnected routes and repair them, so that those traffics involving with that disconnected route experience very high delay and gain very low throughput. After thinking over it, some ideas and solutions come to me. We will describe those solutions in Chapter 3 and evaluate the performance in the Chapter 4.

2. Background

There are two types of ad hoc routing protocols so far: (1) proactive (table-driven) (2) reactive (On-demand). These two kinds of routing protocols work in the distinct manner. This thesis will focus on the Ad-Hoc On Demand Distance Vector (AODV) protocol which is a famous reactive protocol for ad hoc network. In the following subsection, we will introduce some existing protocols for example.

2.1 Proactive (Table-driven) Protocols

The proactive protocols produce route control packets periodically between nodes. All nodes need to maintain a route table for all other nodes in the network. Each time the periodical route control packets are received by some node, this node must compute the route derived from the control packet and update the route table. Even in the low mobility environment, the route never changes was computed over and over. Furthermore, the node may maintain a lot of routes never used. In such case, a lot of control packets are regarded as pure overhead.

2.1.1 Destination Sequenced Distance Vector (DSDV)

In the traditional distance vector protocol, the route loop is the main problem. Destination-Sequenced Distance Vector (DSDV) introduces a new mechanism for that and guarantees loop-freedom. This mechanism works by attaching a sequence number to the route table entry to identify the validity. Each node maintains a routing table with all available destinations along with information like next hop, the number of hops to reach to the destination, sequence number of the destination originated by the destination node, etc. DSDV uses both periodic and triggered routing updates to maintain table consistency.

Triggered routing updates are used when network topology changes are detected, so that routing information is transmitted as quickly as possible. Routing table updates can be of two types - "full dump" and “incremental”. “Full dump” packets carry all available routing information; “incremental” packets carry only information changed since the last full dump in order to decrease the amount of traffic generated.

Mobile nodes cause broken links when they move from place to place. When a link to the next hop is broken, any route through that next hop is immediately assigned an infinite metric and an updated sequence number. This is the only situation when any mobile node other than the destination node assigns the sequence number. Sequence numbers assigned by the origination nodes are even numbers, and sequence numbers assigned to indicate infinity metrics are odd numbers. When a node receives an infinite metric, and it has an equal or later sequence number with a finite metric, it triggers a route update broadcast, and the route with infinity metric will be quickly replaced by the new route. When a mobile node receives a new route update packet, it compares it to the information already available in the table and the table is updated based on the following criteria:

- If the received sequence number is greater, then the information in the table is replaced with the information in the update packet
- Otherwise, the table is updated if the sequence numbers are the same and the

metric in the update packet is better.

The DSDV protocol guarantees loop-free paths to each destination and detects routes very close to optimal. It requires nodes to periodically transmit routing update packets. These update packets are broadcast throughout the network. When the number of nodes in the network grows, the size of the routing tables and the bandwidth required to update them also grows, which could cause excessive communication overhead. This overhead is nearly constant with respect to mobility rate.

2.2 Reactive (On-demand) Protocols

The reactive protocols trigger the route construction if necessary. When a node wants to transmit a packet, it consults its route table to find a valid route to the destination of the packet. If one valid route is found, the route information is applied and sent out the packet. If not, this node initiates a route request process to the destination. The destination or intermediate nodes may response for this route request process. The response which reaches the source node will generate corresponding valid route entry. The validity of the route is determined by the lifetime. If the route is not used for some period, the route is considered to be no longer needed and is removed from the routing table. Before the route expired if the route is accessed, the lifetime of it is extended.

2.2.1 Ad-Hoc On Demand Distance Vector (AODV)

Ad-Hoc On Demand Distance Vector (AODV) is a representative of the reactive protocols in the wireless network. As its name, the protocol operation is performed based on the packet demand. Since the protocol is called “on demand”, there will be no active route maintained if no traffic desires to be transmitted. So the route discovery process is initiated only when the source node tries to send a packet and there is no active route for it.

At first we introduce two main control packets: route request (RREQ) and route reply (RREP). The source node initiates the route discovery process by broadcasting the route request for the destination through wireless link. Each node (excluding the destination) received the route request will forward the route request by broadcasting as well. The dissemination of route request works in the flooding manner until the destination is reached. The destination received the route request will send back a uni-casting route reply to the route request originator through the reverse path. The intermediate nodes on the reverse path will forward the route reply to the source hop by hop. There is an alternative way that the intermediate nodes received the route request can send back the reply if it has an “fresh enough” route for the destination. Whether the route is fresh enough or not is determined by the sequence number. At first sight of this protocol, we may find several problems.

(i) Since the route request is transmitted by broadcasting, the identical route request may be received and forwarded over one time. It wastes the bandwidth of the network.

(ii) At the first time the destination node received the route request, there is no active route available toward the route request originator. So the destination node may encounter some difficulties in sending back route reply.

(iii) According to the dissemination of RREQ is performed by broadcasting, the destination node may receive many RREQs from different route paths. A metric is necessary for the route choosing.

(iv) In the mobile environment, the route may disconnect and trigger the reconstruction. There should be a mechanism to avoid the route loop.

For (i), each node in the ad hoc network maintains a broadcast id. Each time a RREQ attempt is sent. The broadcast id is incremented by 1 and attached into the route request. Combining with the source IP address, it can uniquely identify the RREQ. Besides, each node maintains a history queue for received RREQs. While a RREQ is received, it will compare with the entries in the history queue. Once it matches some in the history queue, it will be taken as redundant route request and not be forwarded any more.

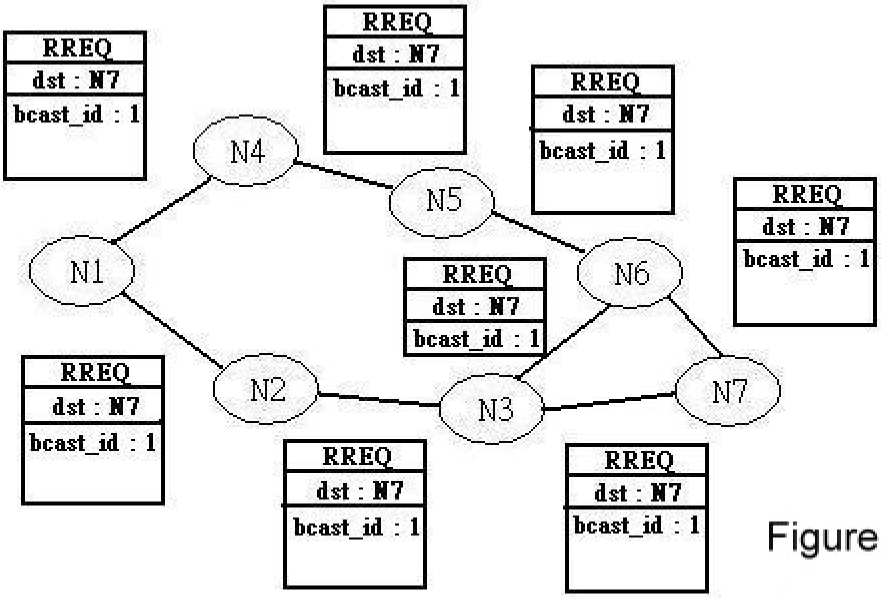


Figure 2-1

For (ii), during the dissemination of the route request, each nodes received the RREQ will generate the active route for the route request originator. The next hop field of this route records the node from which it received RREQ. As the RREQ is forwarded to the destination node, all nodes (including the destination node) involving within the route path have an active route to the route request originator. It is so called “reverse route path”. When the destination node tries to send back the route reply, the reply will transmit through the reverse route path to the source node.

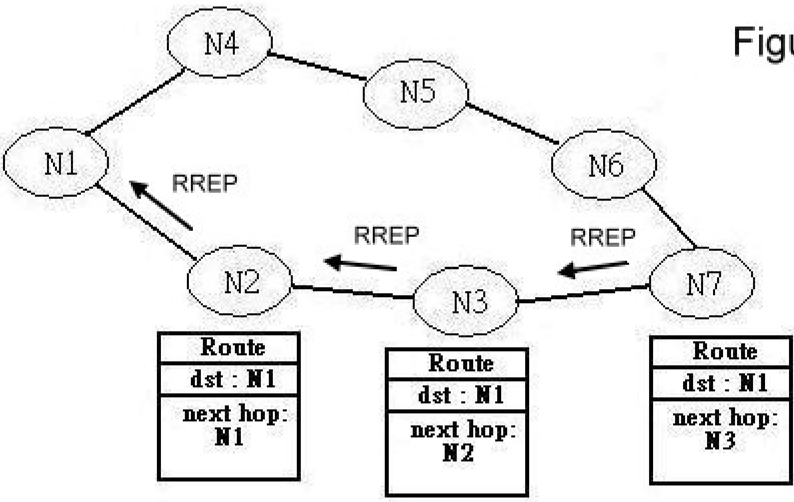


Figure 2-2

For(iii), in the original AODV mechanism, the only metric “hop-count” is derived during the route discovery process. Each time a node received a RREQ, it will increment the hop-count field by one in the RREQ and forward the request again. The destination will discard those RREQs with higher hop-count than that received before. These discarded RREQs won’t trigger the destination node to send back the route reply.

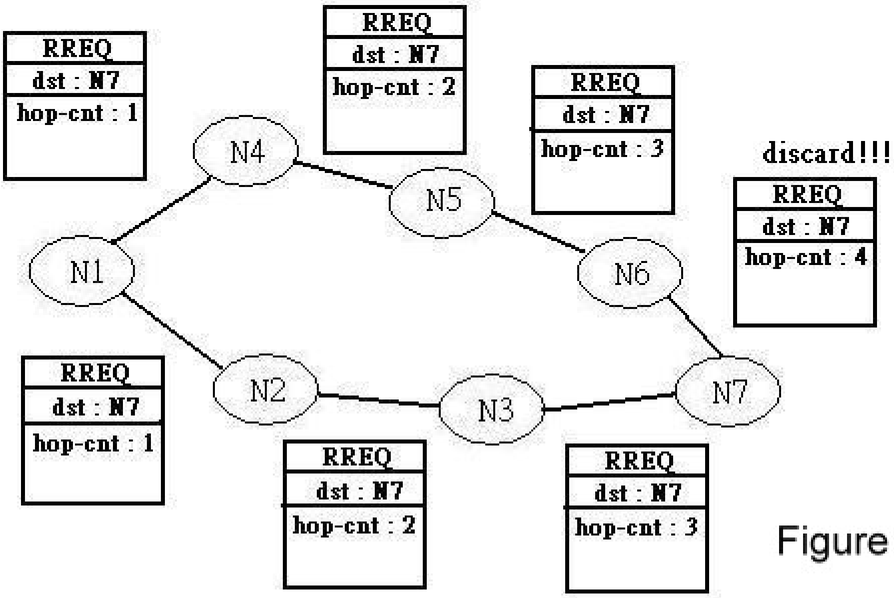
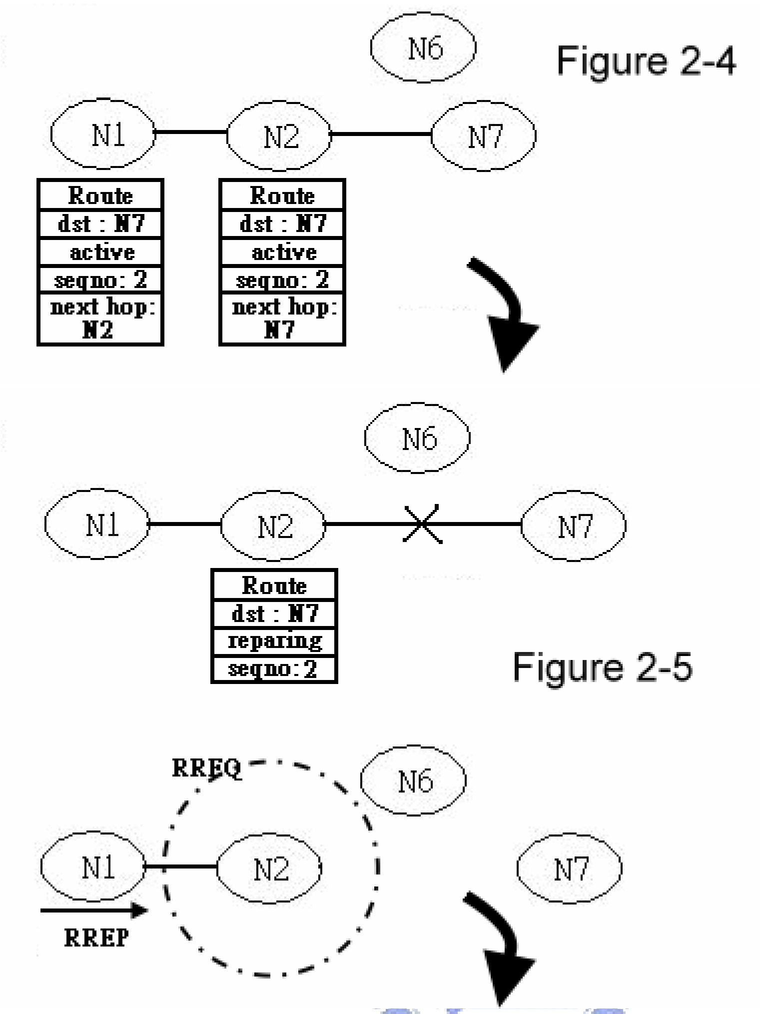


Figure 2-3

For (iv), to avoid the route loop in the network, each node maintains its own sequence number. In the overall network there won’t be any “active” route with higher sequence number than that the destination of the route currently maintains. All route requests over the network can carry the sequence number for the destination one more than the destination currently maintains. It happened when one link of the route broke.



For example, in the Figure 2-5, the link break is detected by N2. Without incrementing the sequence number of the route, N2 broadcasts a RREQ(seqno:2) for N7. It works incorrectly as following description. Since the sequence number of the route for N7 on N1 matches the sequence number of received RREQ, N1 will take this route as “fresh enough” for the route request and send an intermediate reply to N2. Originally the route toward N7 on N1 takes N2 as next hop, after N2 received the intermediate reply from N1. N2 will update the route for N7 and take N1 as next hop toward N7. The route loop between N1 and N2 takes place. It’s a terrible situation for any routing protocol. Any packets consult these routes will be transmitted in the loop until the TTL limit is reached.

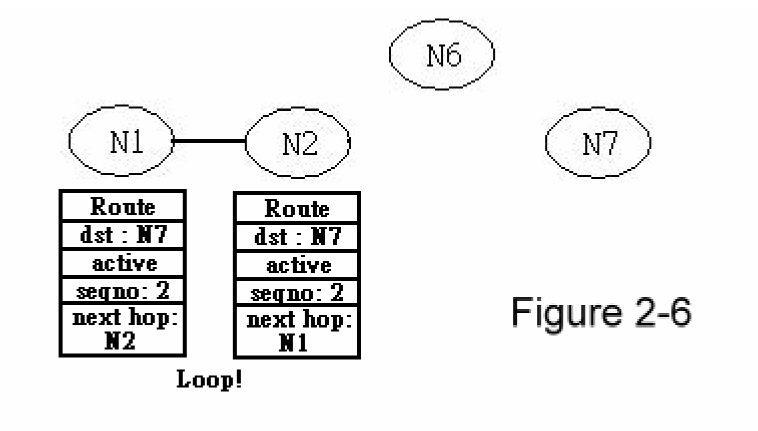


Figure 2-6

In addition, there are two more types of control packets: route error (RERR) and route reply acknowledge (RREP ACK). The route error (RERR) is sent whenever a link break causes one or more destination to become unreachable from some of the node's neighbors. The RREP ACK is an extra retransmission mechanism. Even the RREP is transmitted in uni-casting, and the wireless MAC layer already provides the retransmission mechanism. The RREP ACK is still an important mechanism for route construction. Once the RREP encounters retransmission failure in the reverse route path, it means the overall route discovery fails. It makes the preceding part of the route discovery process as pure overhead. In the Chapter 3, we will describe the deficiency of this mechanism and provide a solution for it.

3. GPS-support AODV

After introduction of the current work, we can find three ways to improve AODV. First, there are some defects and deficiencies existing in the original AODV specification. Second, the control packet overhead of AODV should be reduced further. Third, hop-count is not the best metric of route. There should be some mechanism to introduce new criterion for the route construction.

Here we propose a new AODV protocol with GPS support. In the scheme of GPS-support AODV, we assume each vehicle will be equipped with the GPS-device. With the help of GPS-device, each vehicle can acquire its own global location. Furthermore, approximate speed and moving direction can be calculated through collecting a series of locations. Once the global position, moving speed and direction is available, it helps us to predict the hop-to-hop contact time and detect the link break in advance. In the section 2.3, 2.4, we will describe the "Moving pattern comparison" and "Backup route construction" in detail. These are two main new features of the GPS-support AODV.

3.1 Protocol Description

3.1.1 AODV Revision

First of all, there are several deficiencies in the AODV specification. We will introduce these defects and provide the solutions first then describe the additional improvements in the following sections.

First, the RREP ACK is not well-defined in the spec. In the specification of AODV, there is no information included at all. Even within the transmission range, it doesn't guarantee that the route reply will be transmitted correctly. For example in the following Figure 3-1, the RREP ACK is ambiguous for these two distinct RREPs. For the vehicle 4, it sent the two distinct RREPs at almost the same time, and expected for the RREP ACK from the vehicle 3. If RREP (V2, V6) was dropped in the MAC layer or suffered from the retransmission failure by accident; RREP(V1, V5) is received by the vehicle 3 and corresponding RREP ACK is sent back to the vehicle 4. In such a situation, vehicle 4 received a RREP ACK with blank field such that it can't tell which RREP is ACKed. Under the common implementation, both RREPs will be taken as ACKed. Eventually the lost RREP (V2, V6) won't be retransmitted by the AODV. And the route construction (V2, V6) will remain incomplete until the next RREQ attempt takes place. In our revision, we add the corresponding fields of RREP into RREP ACK. It helps to identify the relation between the RREP and RREP ACK and ensures robust route reply transmission.

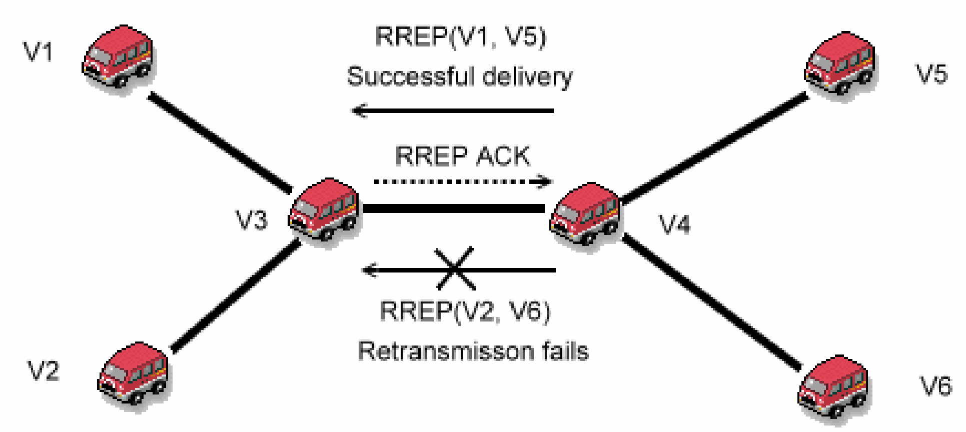


Figure 3-1

Second, the AODV control packets will suffer from the traffic congestion. In the preliminary experiments, we observe the experiments in the traffic congestion often produce the degradation of the AODV performance. After further investigation, it's reasonable to behave like that. As the AODV is an on-demand protocol, the control packets of AODV won't be transmitted as frequently as the data packets in the traffic congestion. These AODV control packets will be dropped in the MAC layer in a high probability. Each time one control packet of AODV is dropped, it causes a significant impact to the performance of AODV. In detail, once RREQ of RREP is dropped due to the traffic congestion, the route construction will be postponed. The successive RREQs or RREPs may suffer as well. During the period, there will be no throughput at all. Hence we will introduce a mechanism to solve this problem. This mechanism will be implemented in the MAC layer to give the AODV control packets higher priority than normal data packets for queuing. One extra queue space is allocated in advance for AODV control packets. In case the queue of MAC layer is full, the consequent out-going AODV control packets will be enqueued into the extra space rather than being dropped. And this control packet saved here will be dequeued after all packets in the original queue are done. It will achieve the goal of higher priority for AODV control packets without disturbing the packet order. In the following performance evaluation, we can observe the mechanism avoid the problem indeed.

3.1.2 Moving Pattern Comparison

In original AODV criteria (i.e. hop count), there is no concern about the link quality or lifetime. During the preliminary experiments, we observe some side effect as a result of the hop count metric. The most obvious phenomenon is that the chosen route with minimal hop count usually suffers from link break in a short time afterward. It degrades the performance of AODV severely. Once the link break takes place, the throughput of traffic will decrease to zero. And the packets will be buffered till route reconstruction is complete. Since the influence of link break on performance of AODV is significant. We introduce a solution as following to relieve it.

With the assistance of GPS, the moving pattern is defined as {speed, vector, location} for each vehicle. In the proposed protocol, the main difference from original AODV is that information derived from the GPS-device will be transmitted during the route construction. In other words, the moving pattern of current vehicle is carried within the RREQ. In each forwarding, the vehicles which received RREQ will compare moving pattern within RREQ with its own. After the moving pattern comparison, the relative moving pattern is determined. As the transmission range is provided, the link lifetime (hop to hop) can be derived. So after the RREQ dissemination, all hop-to-hop link (one of candidate route paths) lifetimes are available.

Hence we introduce a new metric, "minimal link lifetime", for choosing from the candidate routes. For each candidate route, the minimal hop-to-hop link lifetime is taken as "lifetime of the whole route". With this information, the derived route lifetime will be more precise than fixed lifetime of original AODV. As the larger lifetime route is chosen, the less link breaks take place. In the route construction, we can choose one from candidate routes with maximal lifetime against the link break.

The second benefit as a result of the derived route lifetime is to further control the dissemination of RREQ. Once the route request transmits through some link with very small lifetime, those candidate routes containing this link are least probably to be chosen as an active route. Even some route containing this link is, the route path may break in a very short time afterward or worse before the route reply traversed back to the source. Instead of contribution of data packets transmission, these active routes with tiny lifetime cause link break and serious delay. Therefore we declare a minimal threshold of link lifetime called ROUTE_LIFETIME_THRESHOLD. In case a vehicle received a route request, the link lifetime derived from moving pattern comparison is less than ROUTE_LIFETIME_THRESHOLD, we restrain the consequent RREQ dissemination by discarding this RREQ. So all the RREQs reaching the destination are equipped with more than ROUTE_LIFETIME_THRESHOLD lifetime. Consequently the final active route must possess at least ROUTE_LIFETIME_THRESHOLD lifetime.

3.1.3 Backup Route Construction

In the original AODV mechanism, the link break is detected by two ways. One is successive lost Hello message. The other is MAC-layer support for notification of retransmission failure. The former one is applied more generally than latter one. Even though the Mac-layer support is more sensitive to link break than lost Hello message is. The local route repair triggered by lost Hello message is postponed for ALLOWED_HELLO_LOST * HELLO_INTERVAL from link break actually took place. Furthermore the REPAIRING route will become active until success of route reconstruction. Between the link break and success of route reconstruction, there is zero throughput and delay appended to buffered packet. It is an important factor of lower throughput and higher delay in original scheme.

In the previous section, the derived route lifetime is used to relieve the link break problem by reducing the occurrence of link break during the route construction. Despite this improvement is deployed, the link breaks are still inevitable. For the influence of link break on Ad-hoc routing is significant, we propose a mechanism to eliminate the throughput hole and delay due to link break.

The main goal of mechanism introduced is to detect the link break in advance and construct backup routes for the breaking link. Once the link break actually occurs, these routes involving with the broken link will switch to the backup ones without any delay. This mechanism will effectively get rid the throughput hole and delay.

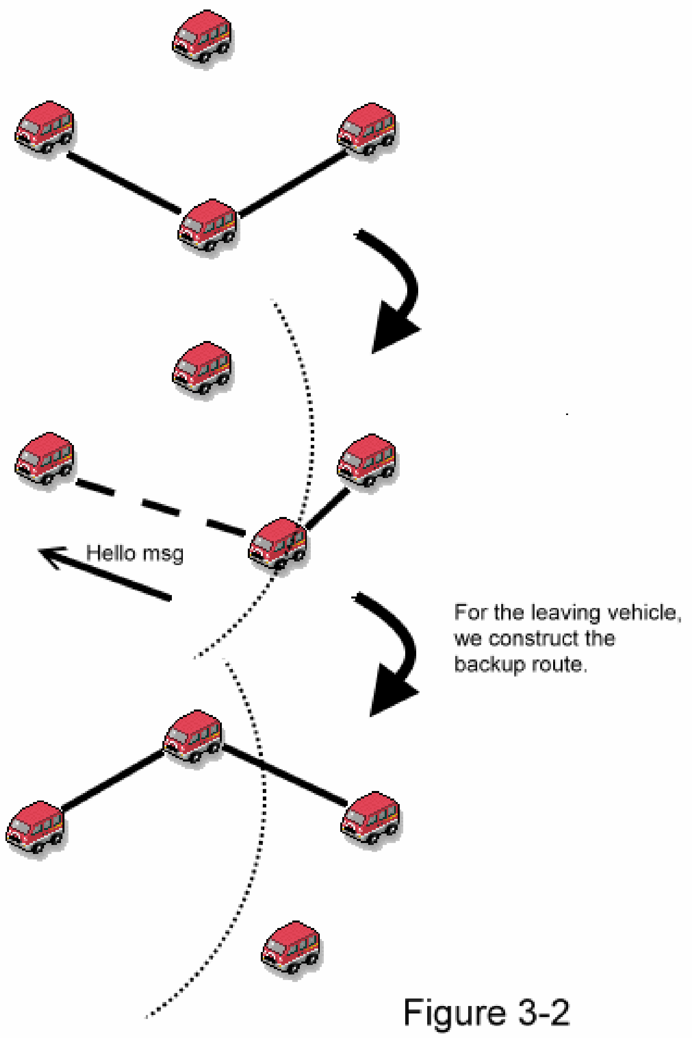


Figure 3-2

In detail, first we must have a capability to detect the link break in advance. With the GPS-support, we append the moving pattern defined as in section 2.1.2 to the periodic Hello message. When one vehicle belonging to the active route received Hello message from others, the hop-to-hop link lifetime can be determined from the moving pattern comparison. Hence we declare a minimal threshold for active link lifetime called NEIGHBOR_LINK_LIFETIME_THRESHOLD. Once the link lifetime derived from the moving pattern comparison due to the received Hello message is lower than the threshold, we can predict the link is going to break. Therefore backup route construction is performed prior to actual link break. In the dissemination of backup route request, as mentioned in section 2.1.2, ROUTE_LIFETIME_THRESHOLD is regarded as a criteria likewise. Here we must raise the ROUTE_LIFETIME_THRESHOLD to be no less than the derived link lifetime. It makes sure that the backup route will still work when the route is switched to it. Finally at the mean time of detection of link break, we replace the broken route with the backup one instead of triggering the local repair. Since there is one more route for the broken one such that the throughput hole and delay can be eliminated effectively.

3.2 Protocol Implementation

In the following experiments and simulation will be held on NCTUns. The NCTUns is a high-fidelity and extensible network simulator capable of simulating various protocols used in both wired and wireless IP networks. Since the NCTUns provides the flexible network module for extensible simulation, we will add the following mechanisms into the corresponding modules in the NCTUns and run simulation for further examination.

3.2.1 AODV revision

The first part is to add necessary field into RREP ACK against ambiguity. There are three fields (Destination IP address, Destination Sequence Number, Originator IP address) necessary for identifying distinct RREPs.

The other part is to give the AODV control packets higher priority than normal packets for queuing. As the queue in the MAC layer is manipulated in the FIFO module of NCTUns, we modify the FIFO module for our purpose.

```
/* ifq full */
char *aodv_lable=((Packet *)pkt->DataInfo_)->pkt_getinfo("isAODV");
if(aodv_lable && (!strncmp(aodv_lable, "yes", 3))) {
        IF_ENQUEUE(&if_snd, pkt);
} else {
        IF_DROP(&if_snd);
        freePacket(pkt);
        return(1);
}
```

3.2.2 Moving pattern comparison

Here we declare 2 class types: Mstate and Mvector. The remaining contact time (RCT) between two vehicles is calculated as following: Given two Mstate objects, we can derive the distance (D) between two vehicles. Then we fixed one vehicle (V1) and calculated the net vector (net-vec) of the other vehicle (V2).

$$\text{net-vec} = \text{vec\_V2} - \text{vec\_V1}$$

Θ is the angle as depicted in the Figure while the transmission range (T-range) is a fixed value. There are four cases to calculate the remaining contact time as following: $0^o \le \theta < 90^o; 90^o \le \theta < 180^o; 180^o \le \theta < 270^o; 270^o \le \theta < 360^o$

Case 1: $0^o \le \theta < 90^o$

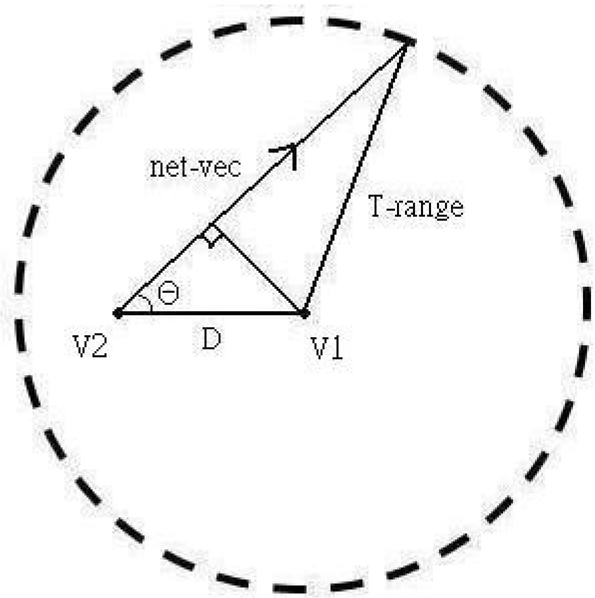


Figure 3-3

Remaining Contact Time (RCT) = $(\sqrt{Trange^2 - (D*\sin\theta)^2} + D*\cos\theta)/speed$

Case2: $90^0 \le \theta < 180^o$

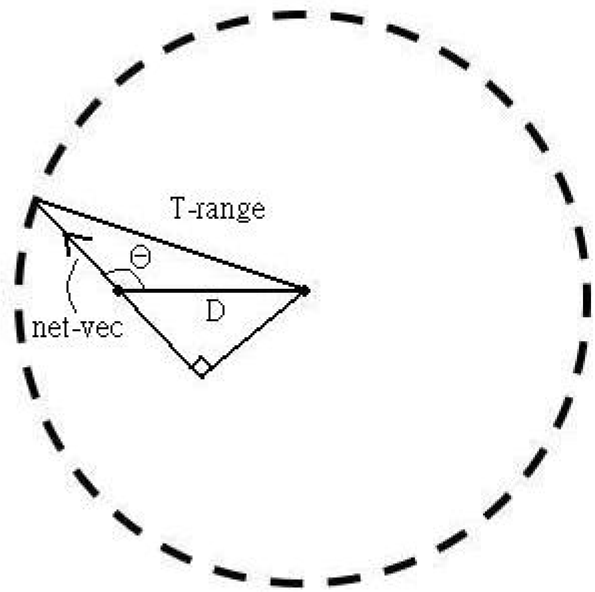


Figure 3-4

Remaining Contact Time (RCT) =

$(\sqrt{Trange^2 - (D*\sin(180^o - \theta))^2} - D*\cos(180^o - \theta))/speed$

Case 3: $180^o \le \theta < 270^o$

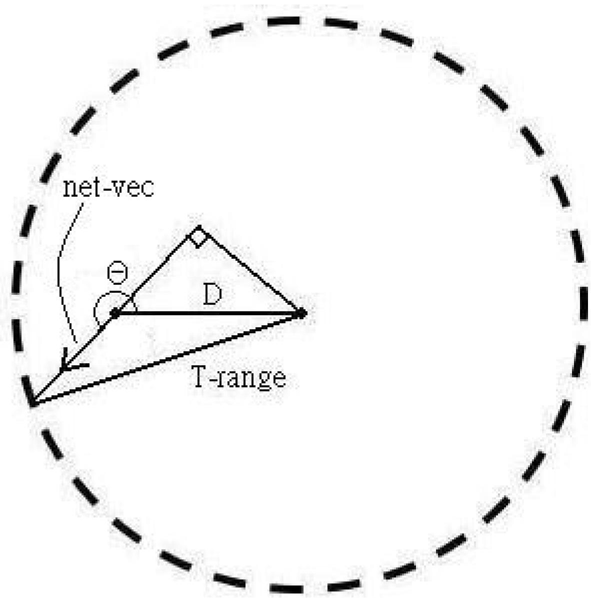


Figure 3-5

Remaining Contact Time

RCT = $(\sqrt{Trange^2 - (D * \sin(\theta - 180^o))^2} - D * \cos(\theta - 180^o)) / speed$

Case 4: $270^o \leq \theta < 360^o$

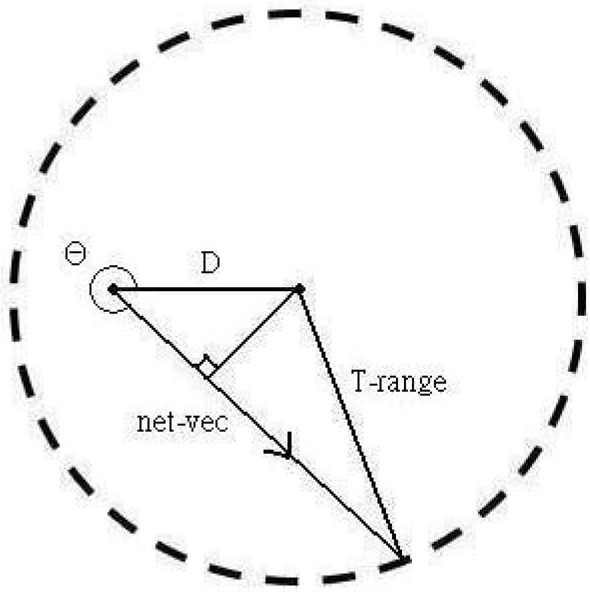


Figure 3-6

Remaining Contact Time (RCT) =

$$(\sqrt{Trange^2 - (D * \sin(360^o - \theta))^2} + D * \cos(360^o - \theta)) / speed$$

The main part of implementation is listed as following:

```
#ifdef GPS_ASSIST_ENABLE
        // calculate the link_lifetime with moving pattern of two nodes
```

```
        Mstate *ms_prehop = new Mstate(my_rreq->loc_x, my_rreq->loc_y,
                                                       my_rreq->speed,
                                          my_rreq->vec_x, my_rreq->vec_y);

        updateMstate();

        double link2prehop_lifetime = mstate.connectLifetime(ms_prehop, 250.0);
        delete ms_prehop;

        if(link2prehop_lifetime < my_rreq->min_link_lifetime)
            my_rreq->min_link_lifetime = link2prehop_lifetime;

        // Once the min_link_lifetime is less than the threshold,
        // the RREQ won't be processed or forwarded.
        if(my_rreq->min_link_lifetime < ROUTE_LIFETIME_THRESHOLD) {
            freePacket(pkt);
            return (1);
        }

    #endif // GPS_ASSIST_ENABLE
```

3.2.3 Backup route construction

As described in the section 2.1.3, to detect the link break in advance, we need to add GPS information into the periodic Hello message.

```
    struct RREP_msg {
                char      type;
                u_short   R:1;
                u_short   A:1;
    #ifdef AODV_BACKUP_ROUTE_ENABLE
                u_short   B:1;      // for backup route
                u_short   Reserved:8;
    #else
                u_short   Reserved:9;
    #endif // AODV_BACKUP_ROUTE_ENABLE
                u_short   prefix_size:5;
```

```
        u_int8_t            rrep_hopcount;
        u_long              rrep_dst_addr;
        u_int32_t           rrep_dst_seqno;
        u_long              rrep_ori_addr;
        u_int64_t           rrep_lifetime;

#ifdef AODV_BACKUP_ROUTE_ENABLE
        // used only in Hello pkt
        double              loc_x;
        double              loc_y;
        double              speed;
        double              vec_x;
        double              vec_y;
#endif // AODV_BACKUP_ROUTE_ENABLE
};
```

The main part of implementation is listed as following:

```
#ifdef AODV_BACKUP_ROUTE_ENABLE
    Mstate *ms_nei = new Mstate(my_rrep->loc_x, my_rrep->loc_y,
                                               my_rrep->speed,
                                my_rrep->vec_x, my_rrep->vec_y);

    updateMstate();

    double link2nei_lifetime = mstate.connectLifetime(ms_nei, 250.0);
    delete ms_nei;

    if(link2nei_lifetime < NEIGHBOR_LINK_LIFETIME_THRESHOLD)
    {
        Rt_entry *p_rt = rtable.rt_getHead();

        // For each route using the leaving neighbor as
        // next hop, construct backup route
        while(p_rt) {
            if( (p_rt->rt_flags == RTF_VALID) && (p_rt->rt_nexthop ==
my_rrep->rrep_dst_addr) && (p_rt->rt_hopcount >= 2) ) {
                if (!backup_route_table.ifExist(p_rt->rt_dst)) {
                    backup_route_table.insert(p_rt->rt_dst,
```

```
GetCurrentTime()+net_traversal_time_);
                                        // seqno++ to avoid loop
                                        p_rt->rt_seqno++;
                                        sendRREQ(p_rt->rt_dst, NET_DIAMETER,
RREQ_BACKUP);
                                    }
                             }
                              p_rt = p_rt->next;
                      } // end of while
              } // end of if
     #endif // AODV_BACKUP_ROUTE_ENABLE
```

Each time the node received the Hello message from its neighbors, it will perform the moving pattern comparison to obtain the remaining contact time. The RCT represents the link lifetime as well. As the link lifetime is less than the predefined NEIGHBOR_LINK_LIFETIME_THRESHOLD, the node will perform the backup route construction for all those routes use the leaving neighbor as next hop.

4. Performance Evaluations

4.1 Simulation Criteria

The performance evaluation is based on the comparison of following metrics.

- *Throughput* : We will show the throughput of the receiver side.

- *Average Delay* : It is an average of the end-to-end delay of all successfully delivered packets. The end-to-end delay is defined as the time elapsed from the moment a packet is sent out from the AODV module, to the time the packet is received at the corresponding module at the receiving node.

- *Control Packet Overhead* : It is the total amount of AODV control packets.

- *Broken Route Count* : It is the total count for all route switches taking place in the simulation environment.

- *Average Route Lifetime* : It is the average of the time that routes remain connected.

4.2 Simulation Environment

As mentioned in the section 2.2, all the following simulation will be performed over the NCTUns platform. All those cases for the simulation are generated by VSSIM, which is a professional simulator for the vehicle traffic. At first, we specify the road topology and the number of vehicles moving in it. After the process of VSSIM, it will generate the vehicle trace file in its own format. Then we use a tiny tool of NCTUns to transform the vehicle trace file into mobile node trace file format of NCTUns. Therefore we can import the mobile node trace file to generate the topology. The final step is to assign the network traffic. The detail will be described as following.

Here we proposed two simulation cases for the vehicle communication.

Case 1:

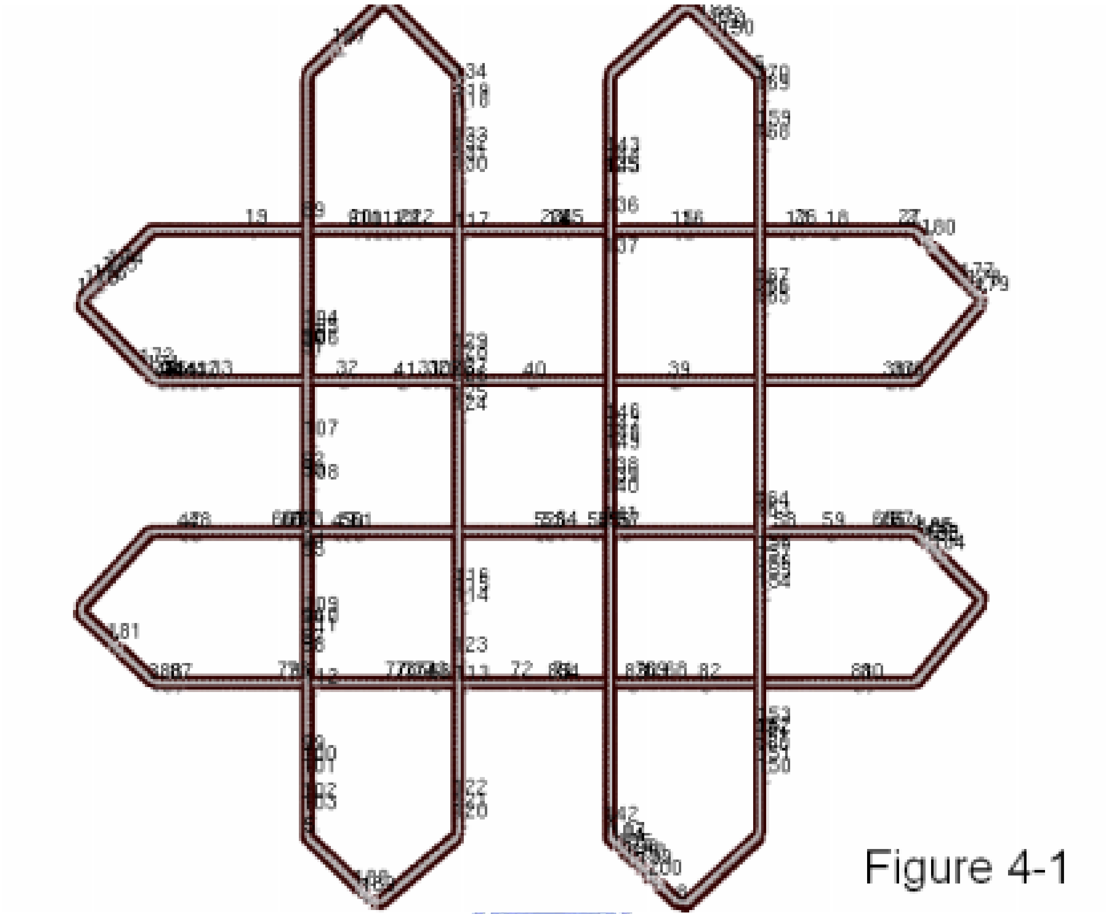


Figure 4-1

This case contains 200 vehicles. And there are ten random connections in it. The topology is designed to simulate the crossing streets or roads in the real world. In this case, the vehicles in the same direction are preferred to be chosen as next hop under our scheme. Since the vehicles on the crossing road often have less remaining contact lifetime. In addition, the moving trace is generated by the VSSIM, it guarantees that any two vehicles won't be the same location at the mean time. In other words, there is no collision and overlap at all. The average velocity is about 50 km/hr.

Case 2:

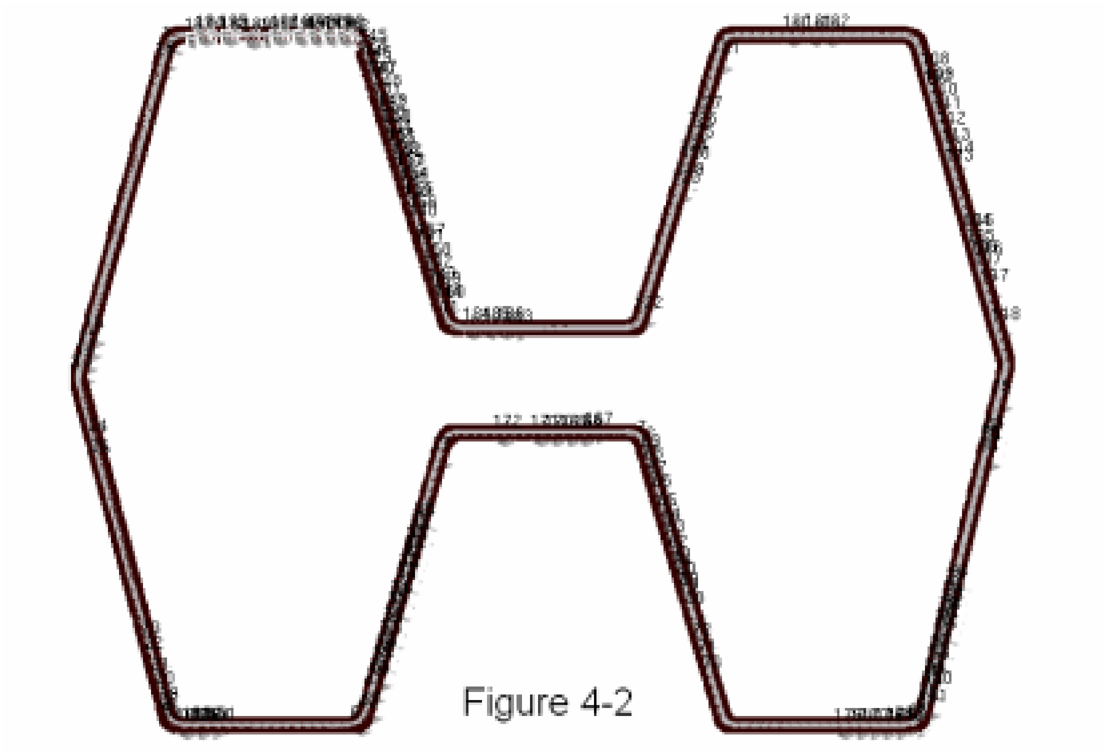


Figure 4-2

This case contains 200 vehicles. And there are ten random connections in it. The topology is designed to simulate the vehicles moving on the highway. In this case, vehicles in a limited area often have about the same direction. As mentioned before, there is no collision and overlap at all. The average velocity is about 50 km/hr.

The following experiments are performed under three kinds of traffic pattern. The traffic patterns are based on the constant bit rate: (1)10 packets per second, (2)100 packets per second and (3)1000 packets per second. Each packet is 1400 bytes in size.

We will compare the simulation results among the ideal routing, original AODV and GPS-support AODV. The ideal routing is the best routing algorithm, even it doesn’t exist in the world. The result of ideal routing is generate by performing the GOD routing of the NCTUns, with which all routes in the simulation environment are computed in advance. Under this scheme, there is no control packet overhead at all. And the packet transmission won’t suffer any delay as a result of the routing process.

4.3 Simulation Results

4.3.1 Throughput

In the following Figures 4-3 to 4-5, we can observe the throughputs of Case 1 in three traffic patterns. In the Figure 4-3 (Case 1, 10 pkts/per sec) and Figure 4-5 (Case 1, 1000 pkts/per sec), the AODV and the GPS-support AODV obtain almost the same throughput. However, in the Figure 4-4 (Case 1, 100 pkts/per sec) the GPS-support AODV works better than the original AODV does and gets 21% promotion.

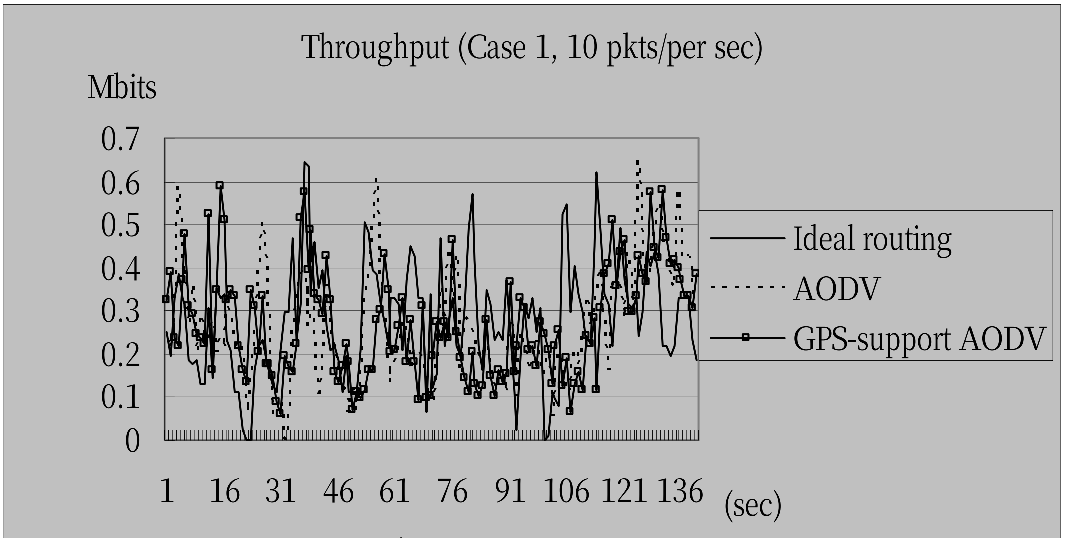


Figure 4-3

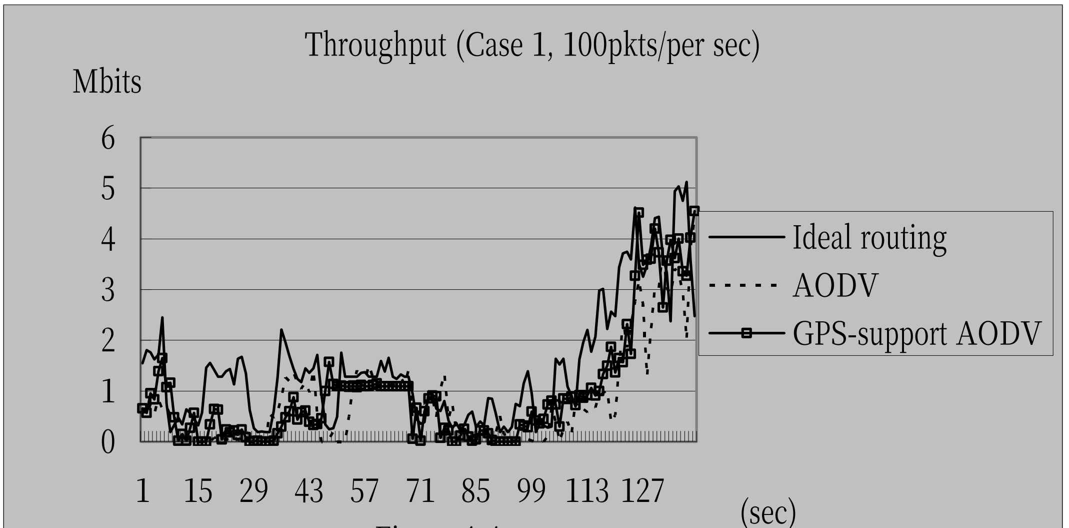


Figure 4-4

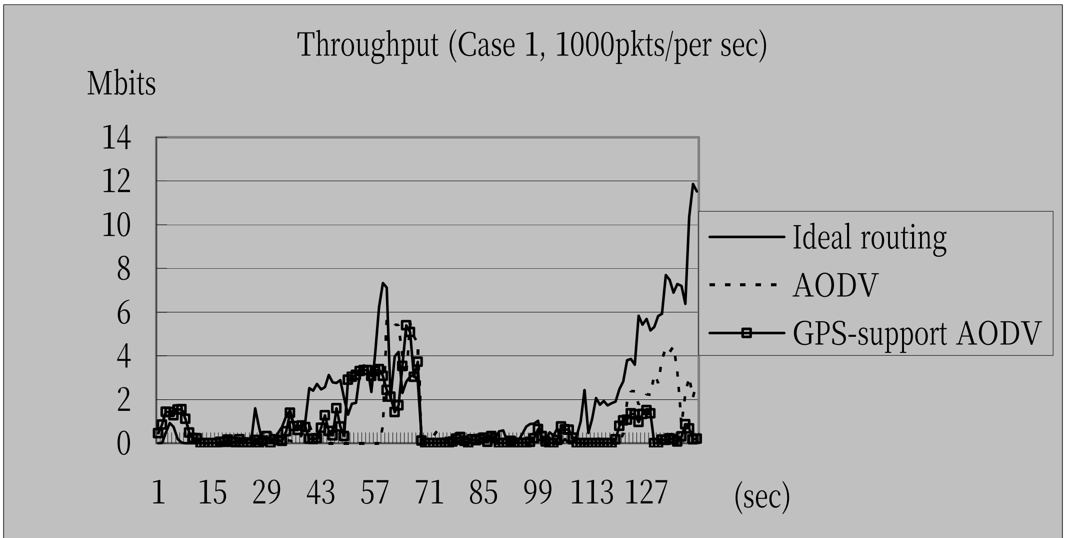


Figure 4-5

In the following Figures 4-6 to 4-8, we can observe the throughputs of Case 2 in three traffic patterns. In the Figure 4-6 (Case 2, 10 pkts/per sec), the GPS-support AODV behaves a little better than the original AODV. Nevertheless, the Figure 4-7 and 4-8 show the benefit of GPS-support AODV over the original AODV. In the Figure 4-7, the GPS-support acquires more throughput than the original AODV at most time and gains 11% promotion. In the Figure 4-8, the throughput of GPS-support is much more over it of the original AODV all the time. The promotion is 76% in this case.

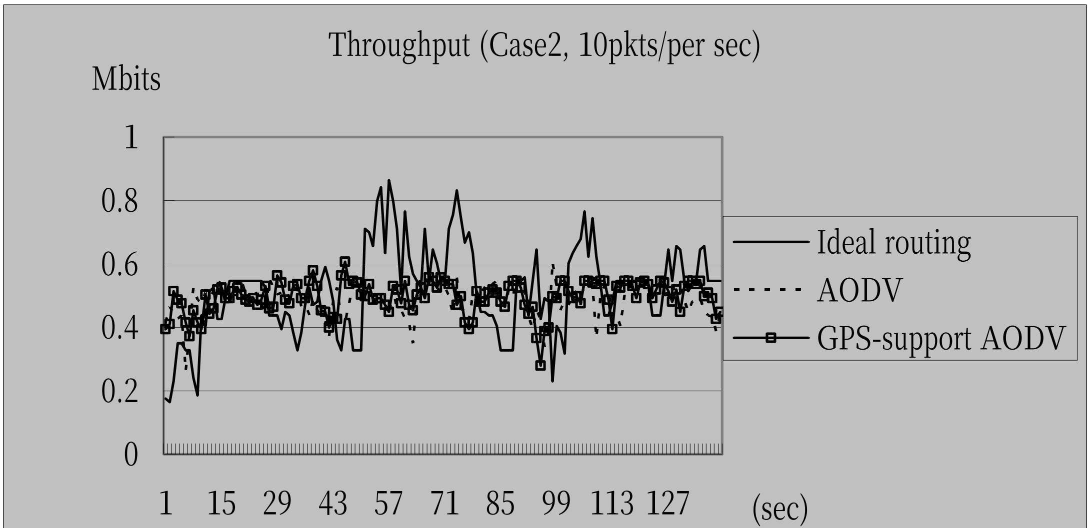
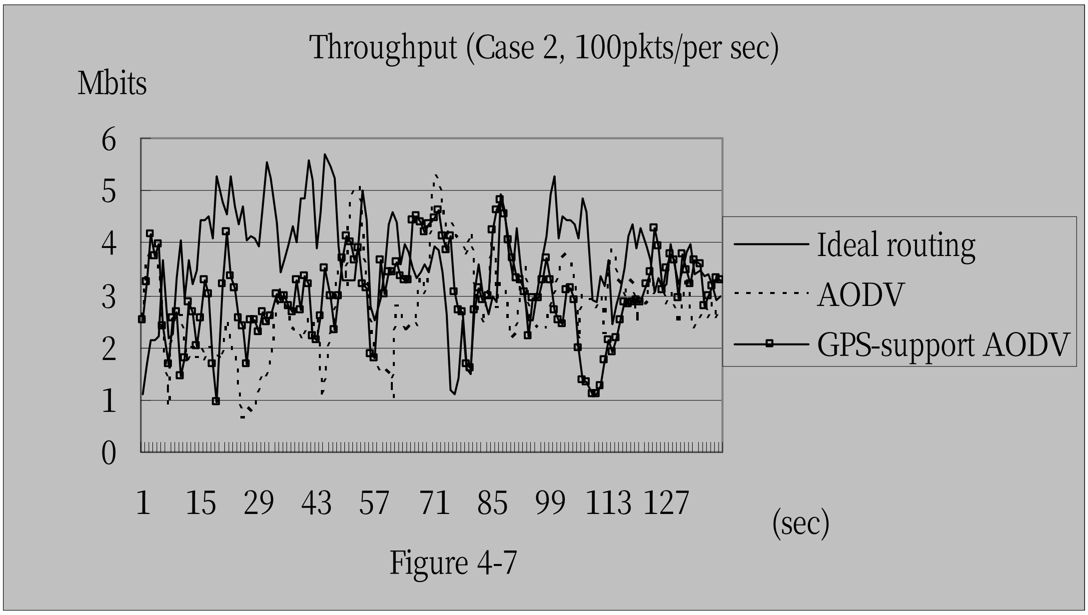
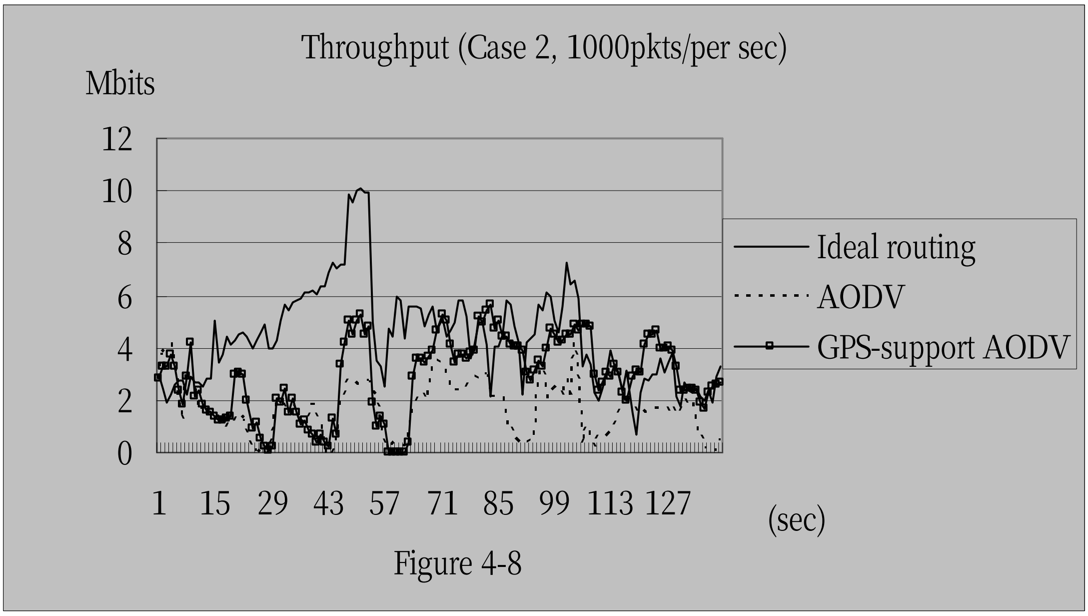


Figure 4-6





The overall throughputs obtained during the entire simulations are represented in the Figure 4-9 and Figure 4-10. Obviously, the GPS-support AODV has the superiority on the throughput over the original AODV.

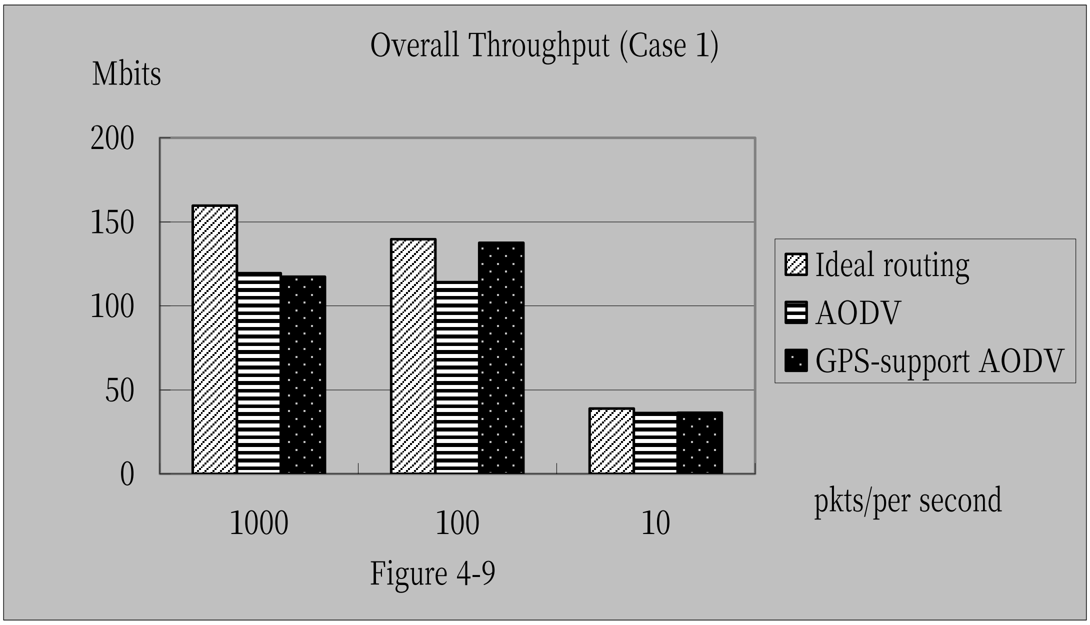


Figure 4-9

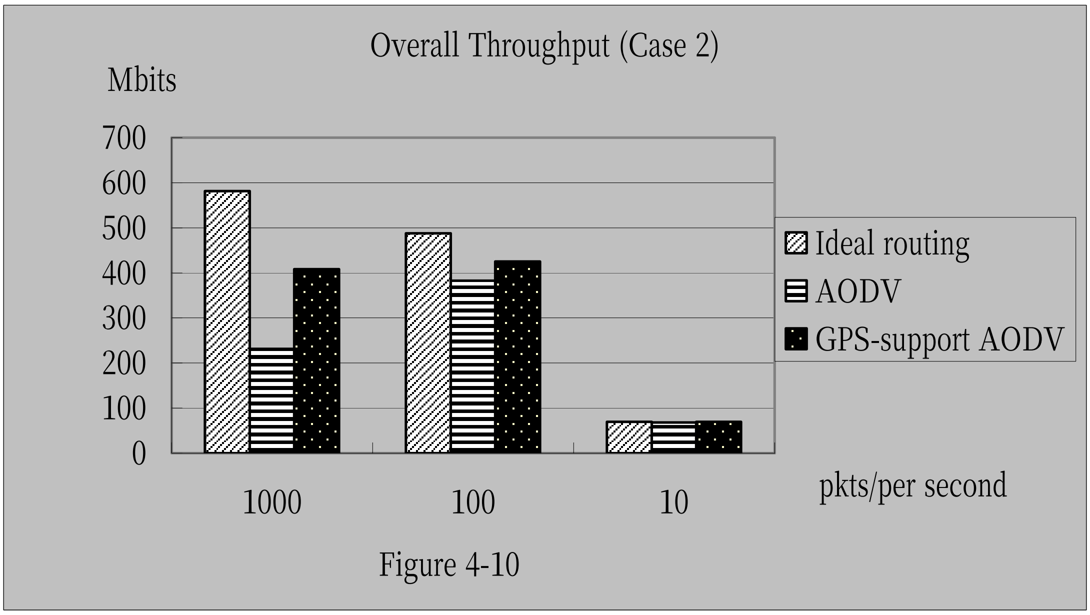


Figure 4-10

4.3.2 Control Packet Overhead

The control packets generated by the route construction process are logged during the simulation. In our prediction, the GPS-support AODV should reduce the control packet overhead significantly. However, the simulation results are not so good as expected. In most cases, the GPS-support AODV reduces the minor portion of the control packets. To be worse, in the Figure 4-11 we can observe that the overhead is more than that of the original AODV.

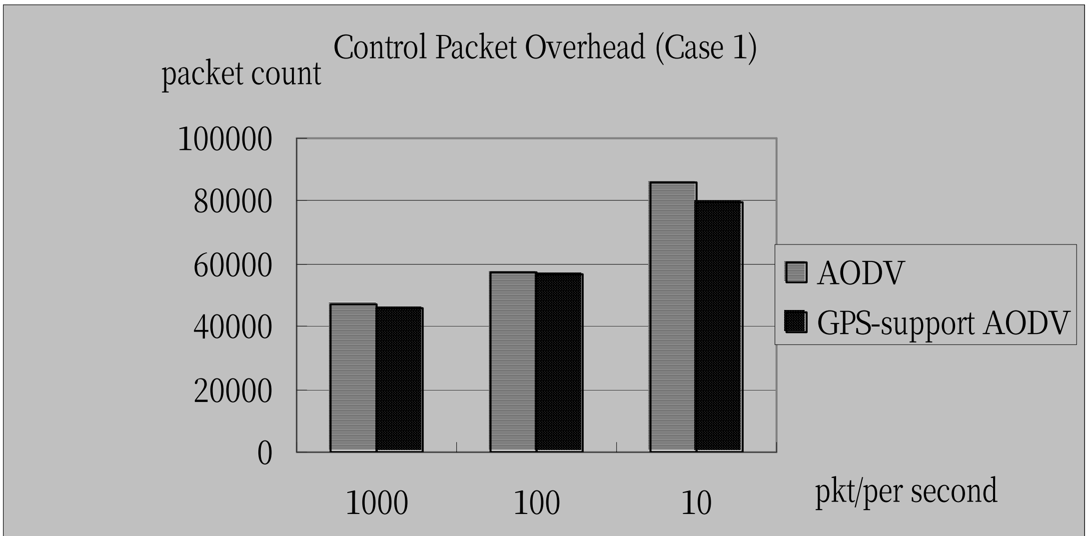


Figure 4-11

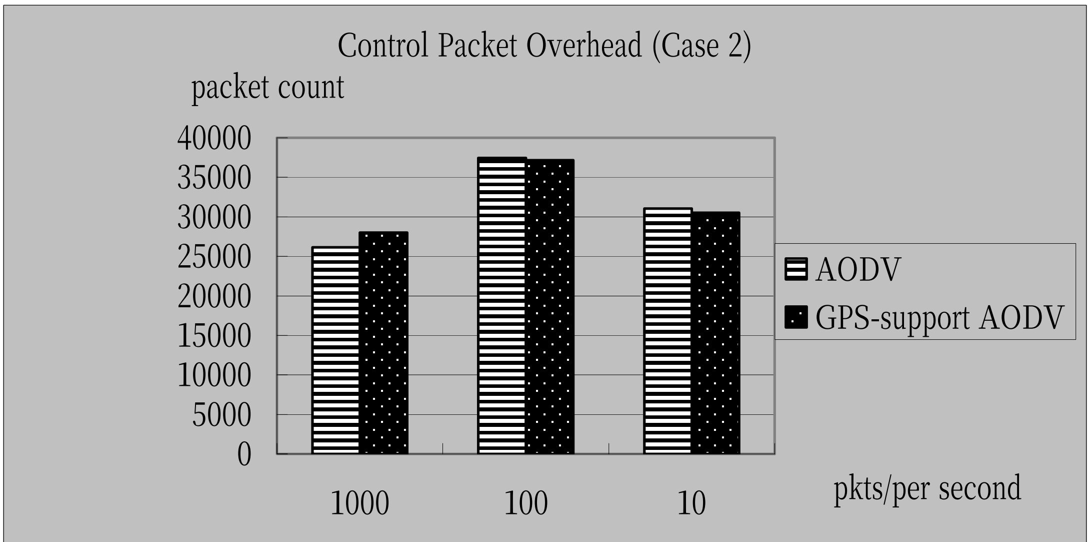


Figure 4-12

4.3.3 Latency

In the Figure 4-15 and 4-16, they represent the end-to-end delay experienced by the packets. Here the ideal routing possesses the extremely lower delay than others do. Obviously, the GPS-support AODV reduces the latency successfully.

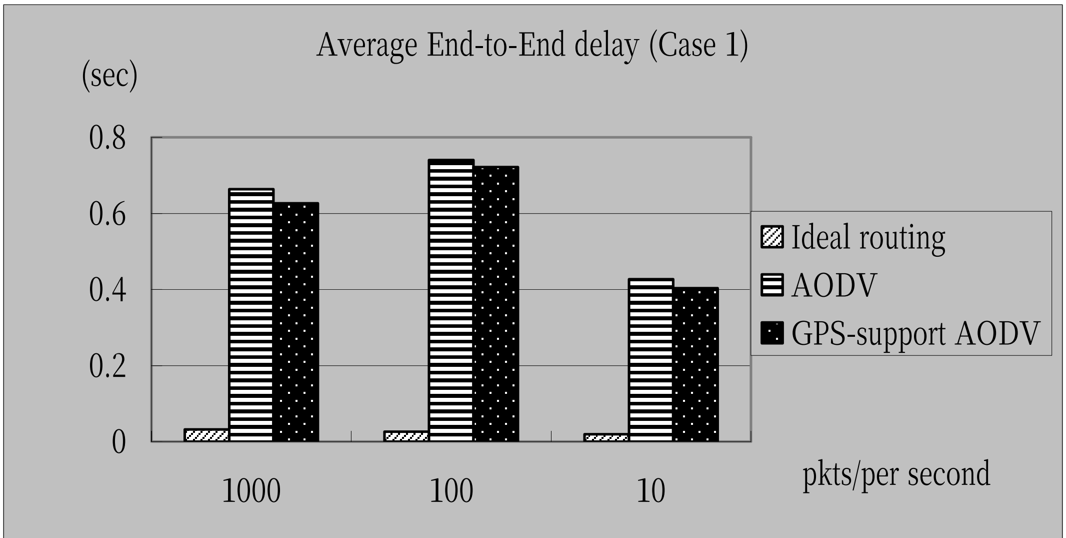


Figure 4-15

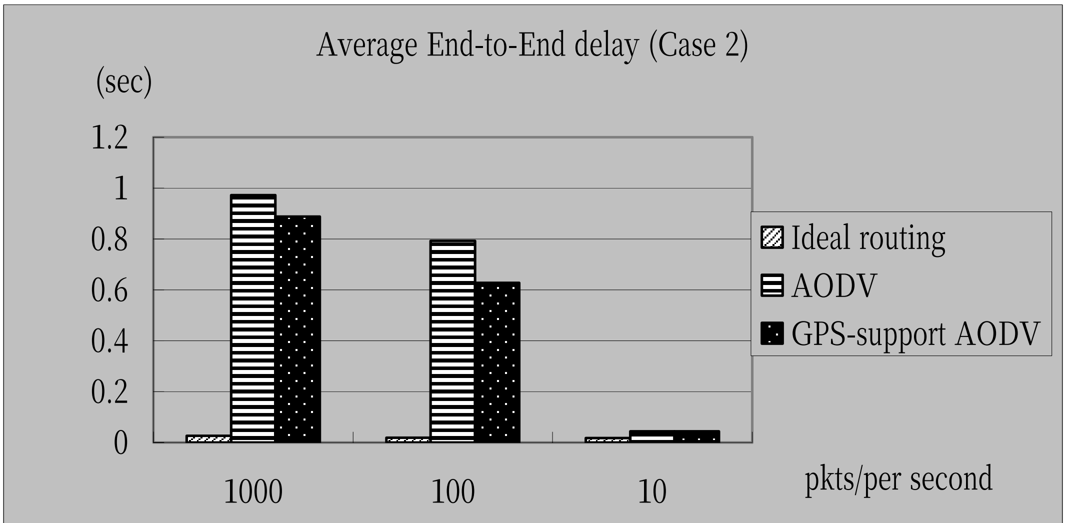


Figure 4-16

4.3.4 Broken Route Count

The count of broken route which took place during the simulation reflects the quality of the chosen routes. As the real route lifetime is short, the route may disconnect at a high frequency. This metric is useful to measure the performance of the backup route construction. To be emphasized, the broken route has a significant impact on the performance. One more broken route may cause a throughput hole for some seconds. Hence the lower broken route count of GPS-support AODV actually causes great promotion.

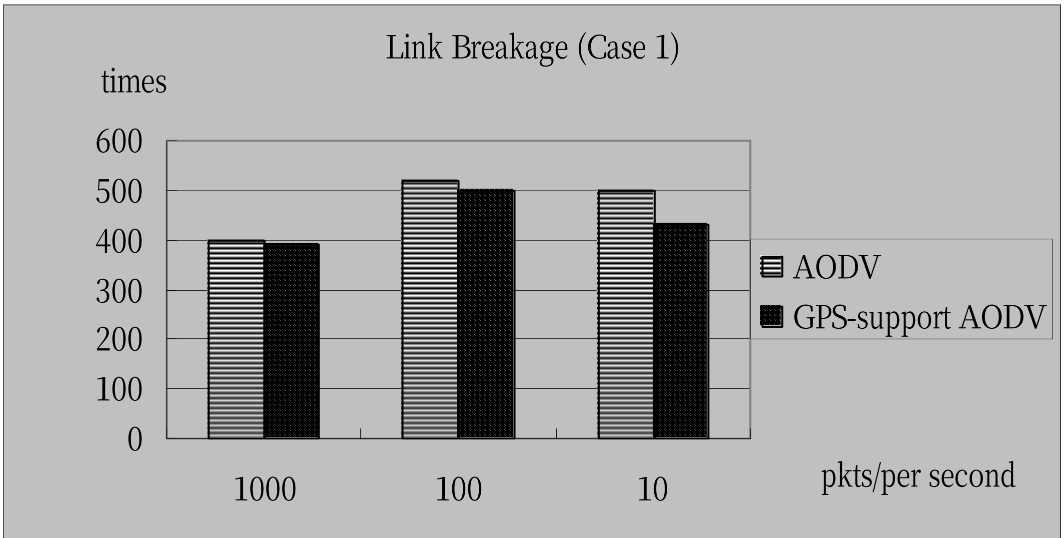


Figure 4-13

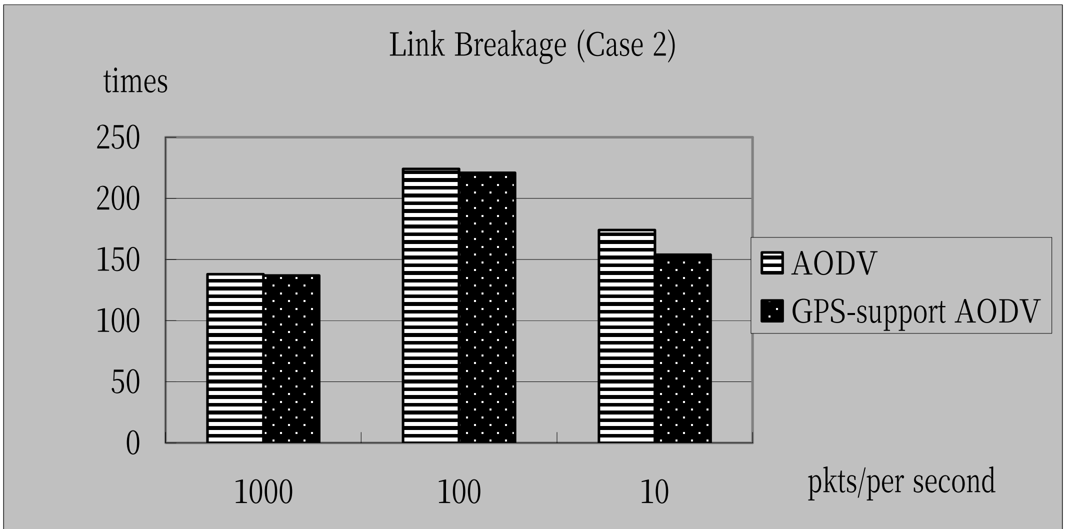


Figure 4-14

4.3.5 Average Route Lifetime

The main goal of our proposed protocol is to relieve the impact of link break by increasing the route lifetime. As in the following figures, we can observe that the average route lifetimes in both cases are longer than those in the original AODV. Due to these more long-lived route, we can improve the throughput and end-to-end delay.

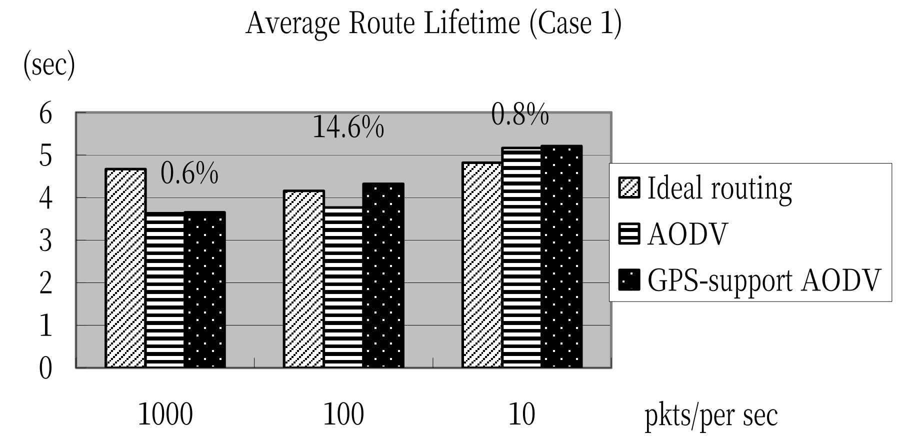


Figure 4-17

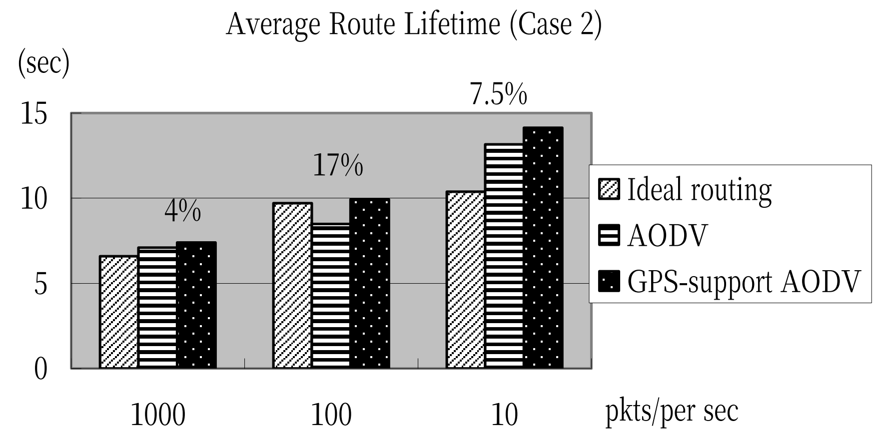


Figure 4-18

4.4 Result Discussions

Before comparing between the original AODV and our GPS-support AODV, we need to examine the performance of these three traffic patterns(10 pkt/sec, 100 pkts/sec, 1000 pkts/sec) in the ideal routing. Even the amount of packets injected into the network in the 100 pkts/sec is ten times those in the 10 pkts/sec, the throughput just achieves about 3.6 times in Case 1, while 7 times in Case 2. Furthermore, from 100 pkts/sec to 1000 pkts/sec, the throughput can merely achieve 1.14 times in Case 1 and 1.19 times in Case 2. After further investigation, it is caused by the wireless link congestion. Though the congestion may trigger the retransmission of the MAC layer, many packets still suffer from the retransmission failure. We will find some performance degradation due to this behavior.

First, the throughputs of AODV and the GPS-support AODV under 10 pkts/sec traffic scheme in both the Case 1 and Case 2 catch up with those of the ideal routing. It is reasonable that the control packets of original AODV get rid of dropping in MAC layer and make the mechanism work well without wireless link congestion. In the 100 pkts/sec scheme, the GPS-support AODV works well and gains 21% and 11% promotion in Case 1 and Case 2. In the 1000 pkts/sec scheme, we observe that the link breakages occur at a higher frequency in Case 1 than in Case 2. With the GPS-support AODV, we will construct the backup route before the breakage takes place. But the backup route construction suffered from the congestion of the wireless link, so many backup route constructions failed. As in the Figure 4-9, GPS-support AODV has no advantages in this scheme. By contrast, GPS-support AODV obtains 76% promotion in Case 2.

Second, under our moving pattern comparison, the control packets through the link with very short lifetime will be discarded and not be forwarded. So we expect considerable overhead will be reduced. But the backup route construction of the GPS-support AODV may introduce more overhead into the network. As in the Figure 4-11 and Figure 4-12, the GPS-support AODV generates less overhead than original AODV in 10 pkts/sec, 100 pkts/sec and 100 pkts/sec schemes of Case 1 and in 10 pkts/sec, 100 pkts/sec schemes of Case 2. There is only one counterexample which is in 1000 pkts/sec schemes of Case 2.

Third, the end-to-end delay in our proposed GPS-support AODV works better than the original AODV in all cases. It benefits from the AODV revision and the backup route construction. Under our AODV revision, the route reply transmission is more robust than before. And the control packets have higher priority for queuing in the MAC layer. So the route construction will succeed after less route request attempts. Together with the backup route construction, the transmitted packets after the link breakage won't experience the delay for route reconstruction.

Fourth, the broken routes take place in GPS-support AODV is less than the original AODV in all cases. Since the amount of the broken routes has significant impact on the performance, even the reduction looks tiny, it improves the performance fatally. The moving pattern comparison contributes to the reduction of the broken routes by introducing the new metric for the route choice. In our improvement, the route quality is based on the route lifetime as defined in subsection 3.1.2. With this metric, the routes chosen by GPS-support AODV will encounter less link breakages.

5. Related Works

In our proposed GPS-supported AODV, we focus on the defects of the AODV and provide a solution for them to suit the vehicle moving. As mentioned above, the link break is the main cause of the degradation performance. Hence we introduce a new metric for the route choosing, it will reduce the link break. Furthermore, we deploy a prediction mechanism for link break and backup route construction to avoid the link break. About the simulation environment, we generate the vehicle moving topology by the professional vehicle simulator VSSIM. The moving patterns of vehicles in the simulation cases are much similar to those in the real world. By contrast with the traditional moving model, random way point, our simulation cases eliminate the sudden direction change and inherit the feature of the real world vehicles. Therefore our performance evaluation appears to be more suitable for the inter-vehicle communication.

Here are some related works:

Lifetime Prediction Routing in Mobile Ad Hoc Networks:

It is based on the battery lifetime prediction, and each node estimates its battery lifetime based on its past activities. The route lifetime is defined as the minimal battery lifetime of those nodes on the route. During the route construction, the route path with maximum lifetime is chosen. This routing protocol works well on static networks. However it can’t handle for the high-mobility networks. Since the high mobility doesn’t reflect high power consumption. The lifetime defined in this protocol only corresponds to the availability of the node, not exactly the link lifetime. In the high-mobility networks, one link of the route path will break before the lifetime defined in this protocol expires. In conclusion, it is not so suitable for mobile vehicle communication.

Maximum Lifetime Routing in Wireless Ad-hoc Networks:

As in the previous protocol, this protocol is based on the battery lifetime prediction as well. It introduces the linear programming formulation and multi-commodity flow to estimate the availabilities of nodes. However the node availability estimated by the power consumption can not directly reflect the link break, so the mobility is not concerned here. It shows that the algorithm works for the static networks. Nevertheless, this protocol suffers from the high mobility environment.

GPS-based Route Discovery Algorithms for On-demand Routing Protocols in MANETs:

This protocol looks similar to our proposed one, but they are different in essence. The main goal of this protocol is to control the dissemination of the route request. In its Position-based Selective Flooding, it depresses the unnecessary request broadcasting. Consequently, this protocol obtains an improvement on the overhead. But in the theory the Position-based Selective Flooding doesn't directly benefit the throughput and end-to-end delay. In fact the Position-based Selective Flooding implicitly eliminates the routes with tiny lifetime. The slight improvement over the throughput and end-to-end delay should be caused by the implicit elimination.

By contrast, our proposed protocol introduces the new lifetime metric for the route choosing. In our scheme, the lifetime metric will relieve the impact of the link break. With the backup route construction, we provide an extra solution to deal with the inevitable link breaks. With these mechanisms, we can directly increase the throughput and reduce the end-to-end delay. Our proposed protocol will work better than this related work.

6. Concluding Remarks

After the performance evaluation of Chapter 4, the throughput of our proposed GPS-support AODV is higher than the original AODV at most 76%. Even it is about the same as original AODV in one case. As to the control packet overhead of our proposed GPS-support AODV, it's less than that of the original AODV in most cases. About the latency and the link breakage, our proposed protocol works better than the original AODV. Further, the moving trace of these nodes in the simulation is based on the vehicle simulator. The simulation results also show that our proposed GPS-support AODV makes improvements over AODV and more suitable for the inter-vehicle communication in the real world.

References

[1] Jatinder Pal Singh, Ashish Ahuja and Sulabh Agarwal, **"**LINK CONNECTIVITY ASSESSMENT BASED APPLICATIONS FOR MOBILE AD-HOC NETWORKS"

[2] D. Jhonson and D.Maltz, "Dynamic Source Routing in Adhoc Wireless Networks", *Internet* Draft, Dec. 1998.

[3] E.M. Royer and C-K Toh, "A Review of Current routing Protocols for Ad-hoc mobile wireless networks", *IEEE Personal Communications*, April 1999.

[4] Shailendra M. Tripathi, R. K. Ghosh, "An Implementational Study of Certain Heuristics for the Performance Enhancements of AODV"

[5] PRADEEPKUMAR MANI, "Development and Performance Characterization of Enhanced AODV Routing for CBR and TCP Traffic"