A Control Packet Minimized Routing Protocol for Ad-hoc Wireless Networks

Youn-Sik Hong
Department of Computer Science and Engineering, University of Incheon,
12 Songdo-dong Yeonsu-gu 406-772 Incheon Korea
e-mail: yshong@incheon.ac.kr

Abstract

AODV routing protocol is intended for use by mobile nodes in ad-hoc wireless networks. As the degree of node mobility becomes high, however, the number of RREQ and RREP messages during the route discovery process increase so rapidly. The unexpected increases in the number of control packets cause the destination node to decrease the packet receiving rate. Besides, the overall energy consumption for the network can be increased. Thus, we propose a novel method of adaptively controlling the occurrences of the control packets based on AIAD (additive increase additive decrease) under a consideration of the current network status. We have tested our proposed method with both the conventional AODV and the method using timestamp based on the three performance metrics; i.e., node mobility, node velocity, and node density, to compare their performances.

Keywords: Ad-hoc network, AODV, expanding ring search, node mobility, node traversal time

1. Introduction

Ad-hoc wireless network architecture is a self-organizing and distributed controlled network formed by a set of stations (called *nodes*) that can freely and dynamically self-configure and organize themselves to set up a temporary wireless network [1]. In the ad-hoc network configuration, a node acts as a mobile terminal as well as a router to forward messages to neighboring nodes if possible. The network topology in an ad hoc network is highly dynamic due to the movement of nodes; hence an on-going session suffers frequent path breaks. A disruption may be occurred either due to the movement of the intermediate nodes in the path or due to the movement of end nodes. Routing protocols for ad-hoc networks must be able to perform efficient and effective mobility management.

Existing ad-hoc routing protocols may generally be categorized as *table-driven* and *source-initiated demand driven* [2]. In table-driven routing protocols, every node maintains the network topology information in the form of routing tables by periodically exchanging routing information. These protocols suffer from excessive control overhead that is proportional to the number of nodes in the network and therefore is not scalable in ad-hoc wireless networks, which have limited bandwidth and whose topologies are highly dynamic.

Unlike the table-driven routing protocols, on-demand routing protocols execute the *path-finding* process and exchange routing information only when a path is required by a source node to communicate with a destination. One of the on-demand routing protocols is AODV (*Ad-hoc On-demand Distance Vector*) [5]. In AODV, a source node floods the Route Request (RREQ) packet in the network when a route is not available for the desired destination. All intermediate nodes having valid routes to the destination, or the destination node itself, are allowed to send RREP packets to the source. AODV reduces the need for the system wide broadcasts by localizing the propagation of changes in the network. Even though it performs well in static and low-mobility environments, the performance degrades rapidly with increasing mobility.

As shown in Figure 1, data packet transmission is slightly affected by the node mobility. However, as the node mobility increases the control packets like RREQs and RREPs increase rapidly. Then it causes route discovery latency to increase. Besides, the amount of energy consumption increases corresponding to the increased number of control packets. To the end, the overall performance should be degraded with high node mobility. Our primary concern is to

enhance the performance of AODV by reducing the volume of these control packets effectively due to the node mobility.

The rest of this paper is organized as follows. In section 2, we describe the expanding ring search algorithm and related works. In section 3, we present a proposed approach based on AIAD method. In section 4, a series of experimental results will be given and finally we conclude our paper in section 5.
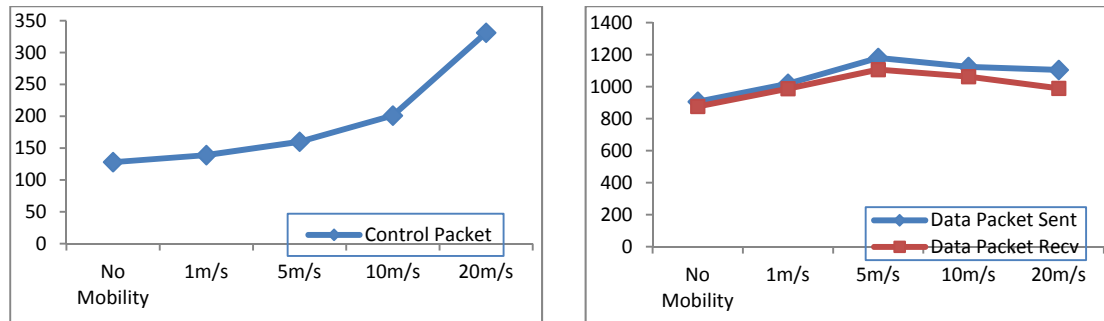

Figure 1. The Number of Control Packets (*left*) and the Number of Data Packets (*right*) with Node Mobility

2. Expanding Ring Search and Related Works
2.1. Expanding Ring Search
Consider the example depicted in Figure 2. In this figure, the source node 1 initiates a path-finding process by originating a RREQ to be flooded in the network for the destination node 8. When the intermediate nodes 2, 3, and 4 receive the RREQ packet, they check their routes to the destination. In case a route to the destination is not available, they further forward it to their neighbors. If the RREQ reaches the node 8 through path 1-4-6-8 or any other alternate route, it also sends a RREP to the source.

Figure 2. AODV Route Discovery; Propagation of the RREQ packets (left) and a Path of the RREP Packet to the Source (right)

To prevent unnecessary network-wide dissemination of RREQs, the source node should use an *expanding ring search* (ERS) technique [6]. Centered on the source node, ERS successively searches larger areas until a node with the information being searched is located. The search begins with a small *time to live* (TTL) value (usually takes as 1) for a query and a time-out period for receiving a reply (proportional to TTL). If there is no reply within the time-out period, the source node increments the TTL by a specified value and re-initiates the query. Thus, by setting the appropriate TTL value in the query, the source node can control the search radius. Table 1 gives default values for some important parameters associated with ERS techniques used in the AODV routing protocol.

In an expanding ring search, the source node initially uses a TTL = TTL_START in the RREQ packet IP header and sets the timeout for receiving a RREP to RING_TRAVERSAL_TIME milliseconds [5]. If the RREQ times out without a corresponding RREP, the source node broadcasts the RREQ again with the TTL incremented by TTL_INCREMENT. This continues until the TTL set in the RREQ reaches TTL_THRESHOLD, beyond which a TTL = NET_DIAMETER is used for each attempt. Beyond this, the RREQ is not

forwarded any further. Each time, the timeout for receiving a RREP is RING_TRAVERSAL_TIME.

Table 1. Default Values for AODV Parameters

Parameter name	value
NET_DIAMETER	35
NODE_TRAVERSAL_TIME	40 ms
NET_TRAVERSAL_TIME	2 × NODE_TRAVERSAL_TIME × NET_DIAMETER
RING_TRAVERSAL_TIME	2 × NODE_TRAVERSAL_TIME × TTL_VALUE
TTL_START	1
TTL_INCREMENT	2
TTL_THRESHOLD	7

NODE_TRAVERSAL_TIME (NTT) is a conservative estimate of the average one hop traversal time for packets and should include queuing delays, interrupt processing times and transfer times [5]. With a fixed value of 40ms for NTT, it does reflect a full dynamic topology with respect to node mobility in ad-hoc wireless networks. For example, one node moves to the right in some distance from its neighbors, while it still resides in a transmission range of them. It causes a longer response time for the query from its neighbors. If the node cannot reply within a specified time-out period, the source node re-initiates a RREQ packet. Thus, by setting the appropriate NTT value the unnecessary RREQ packets should be minimized. Changing NTT also alters the node's estimate of the both NET_TRAVERSAL_TIME and RING _TRAVERSAL_ TIME, and so can only be done with suitable knowledge about the behavior of other nodes in the ad hoc network [5]. Minimizing the cost of the initiating and the forwarding RREQ packets is crucial for resource-constraint multi-hop wireless network, which motivates this work.

2.2. Related Works

A number of heuristics for enhancements of the performance of AODV routing protocol has been proposed. *Hassan and Jha* [7] explored to find an optimum TTL threshold *L* that would minimize the expected bandwidth cost of ERS. They give the following experimental results; search threshold L of 3 is optimum for three categories of networks, that is, large networks with small radius, medium networks with large radius and medium networks with small radius. In the AODV, TTL_THRESHOLD is typically set to 7, which gives the search threshold of 3.

Some of the works tries to identify the inefficient elements of ERS. *Lee* et al [9] proposed a method of timestamp to compute NTT; every node adds the current time to the message (i.e., RREQ and RREP) before sending it out, and then the neighbor that receives it computes a measured traversal time (called *M*) using the timestamp. A smoothed NTT is updated by using the similar equation of the round-trip-time estimator:

$$NTT = \alpha \times NTT + (1 - \alpha) M \qquad (1)$$

Where, α is a smoothing factor with a range of 0 and 1. *Kim* et al [10] presented a technique of estimating NTT that divides a time difference between the source node and the destination node by the number of hops. In addition, *Park* et al [11] proposed a blocking ERS (B-ERS for short). One of the differences from the TTL-ERS is that the B-ERS does not resume its route search procedure from the source node each time a rebroadcast is required. *Tripathi* et al [8] proposed four heuristics; one for them is the utilization of route caches at intermediate nodes for route discovery.

All of the works mentioned above have an emphasis on the adaptive control of AODV routing protocol with respect to node mobility. Since the timestamp based approaches [9] changes NTT too rapidly, the stability of the network becomes relatively low. Besides, with low mobility their performance will be degraded due to increases in the number of RREQ packets initiated by the source node. Even if B-ERS reduces in route discovery latency, it may suffer from dynamic changes in the network topology with high mobility. With the proposed heuristics proposed by *Tripathi* et al, the complexity of maintenance of multiple paths demands a larger memory and greater processing power for every node.

3. A Control Packet Minimized AODV Protocol

TTL based ERS restraints its searching range by giving RREQs a pre-defined TTL number. The TTL number corresponds to the radius of a searching area. Once an intermediate node located at *H* hop distance receives an RREQ, the source node waits for the following waiting time. The *waiting time* WT is calculated as:

$$WT = 2 \times H \times NODE\_TRAVERAL\_TIME \qquad (2)$$

Clearly, *WT* is equal to or less than the RING_TRAVERSAL_TIME. In addition, *H* is proportional to the TTL number. With the fixed value for NTT, it can cause to increase the unnecessary RREQs initiated by a source node. For example, if a propagation delay of one link is longer than 40ms, the source node can be received a corresponding RREP after the RREQ times out. In that case, the source node already initiates a new RREQ with the incremented TTL number.

Our proposed method relies on packet based propagation of timestamp information to calculate more realistic NTT by considering the network status. While initiating a route reply, the destination node appends the timestamp information in the RREP packet. When the source node receives the RREP packet, it checks the timestamp *ST* available at the source node and the timestamp *DT* available at the destination node. Let *MT* denotes estimated time taken for a packet to be delivered from one node to one-hop neighboring node. Let the number of hops on the route path established between the source node and the destination node be denoted by H. Then the *MT* is calculated as:

$$MT = (ST - DT) / H \qquad (3)$$

A computed estimate *MT* of one-hop traversal time just gives a reference relative to a previous NTT. A new NTT should be adjusted depending on MT. For example, the path 8-6-4-1 in Figure 2 consists of three links. Assume that each of them has a propagation delay of 40ms, 30ms, and 20ms, respectively. In that case, MT is computed as 30ms. However, since the longest propagation delay should be considered, a NTT is not changed and still remains 40ms. If the propagation delays of the all links increase equally in 6ms due to node mobility, MT is 36ms and then the amount of the difference between NTT and MT becomes smaller. Thus, a new NTT should be increased by a certain amount ∆. On the contrary, if the delays of the all links decrease equally in 6ms, MT is 24ms and then the amount of difference between them becomes greater. So a new NTT should be decreased by ∆. The above explanations are summarized as shown in Table 2.

Table 2. The Recalculated NTT with Respect to the Propagation Delays of each Link (unit: ms)

Propagation delay			MT	previous NTT	NTT – MT	New NTT
Link1	Link2	Link3				
40	30	20	30	40	10	40
46	36	26	36	40	4	40+∆
34	24	14	24	40	16	40-∆

Then the following algorithm, called CP_AODV (*Control Packet minimized* AODV), is used to resolve the number of control packets including RREQ and RREP packets. To minimize the flooding of control packets, a NTT should be adjusted smoothly by AIAD (*additive increase additive decrease*) technique.

```
IF (min_threshold > NTT – MT)
{
      NTT = NTT + increment;
}
ELSE IF (max_threshold < NTT – MT)
{
      NTT = NTT – increment;
      IF (NTT < lower_bound)
      {
          NTT = lower_bound;
      }
}
```

The parameters *min_threshold, max_threshold,* and *increment* are adjustable with respect to the network status. Notice that *max_threshold > min_threshold*. When the amount of the difference between NTT and MT is less than *min_threshold*, the average propagation delay of the links seems to be increased. So a new NTT should be increased by a specific amount *increment*. On the contrary, when the amount of the difference between them is greater than *max_threshold*, the average delay seems to be shorter. Then a new NTT should be decreased by *increment*. The parameter *increment* is controlled adaptively depending on the movement of nodes. Our experimental results show that with less than 40ms of NTT the number of retransmission of the RREQ packets increases so rapidly and thus the overall performance is degraded. Thus, the lower bound on NTT is set to 40ms. While the amount of difference between them is in the range of *min_threshold* and *max_threshold,* the node mobility seems to be relatively low. Then the value of a current NTT remains as before.

4. Performance Evaluations

For the purpose of the simulation, AODV protocol implemented in QualNet 4.0 simulator [12] has been modified to incorporate the changes which have been proposed in this paper. Data payload is of 512 bytes. All the data packets are CBR (*continuous bit rate*) packets. Each node moves with randomly chosen maximum speed up to 10m/s. The simulation parameters are summarized as shown in Table 3. Notice that the control parameters used in the CP_AODV are set through our simulations as the following: *min_threshold* = 5, *max_threshold* = 10, *increment* = 5.

Table 3. Simulation Set-up

attributes	value
space	1,500m x 1,500m
bandwidth	2MB
Number of nodes	49
Placement strategy	GRID
Transmission range	250m
Node velocity	0 ~ 10m/s
run time	600 seconds

The performance evaluation is based on the comparison of the three different metrics; node mobility, node velocity and node density. In order to get a realistic insight into the effects of the proposed method (CP_AODV), these metrics also have been evaluated for both the legacy AODV [6] and the modified AODV with the timestamp method [9] (*T-AODV* for short).

4.1. The Experiments for the Node Mobility

After reaching the destination the node begins to move again after pause time. In our experiments we varied it between 0 to 600 seconds. The network between the pause times 0 to 150 have a high mobility, whereas the network beyond 450 have a low mobility. The network between the pause times 150 to 450 have a moderate mobility. The pause time implies the level of the mobility in the network.

We classify RREQ messages into two types; a RREQ initiated by the source node and a RREQ forwarded by an intermediate node. The source node re-initiates the RREQ if there is no route reply within a time-out period. Thus, a possible measurement to evaluate the cost of network-wide flooding is the number of the RREQ packets initiated by the source node. RREQ packets are flooded in the network by intermediate nodes forwarding the requests to their neighbors. So, another measurement is the number of RREQ packets forwarded by them.

For the number of RREQ packets initiated by the source node, CP_AODV is remarkably less than AODV as shown in Figure 3. In addition, CP_AODV is better than T-AODV except in low mobility. For the number of RREQ packets forwarded by intermediate nodes, CP_AODV has significantly lower average packets than the two protocols. Typically, with CP_AODV the number of RREQ packets forwarded is reduced by 28% and 21% compared to AODV and T-AODV, respectively.
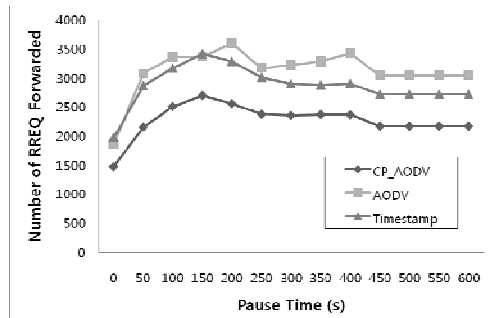
Figure 3. The Number of RREQ Forwarded by Intermediate Nodes

The decreased number of RREQ packets result in the decreased number of both RREP and RERR. For the number of RREP packets initiated by the destination node, CP_AODV has significantly low average packets in all levels of node mobility than both protocols as shown in Figure 5. In addition, with CP_AODV the number of RERR packets is reduced by more than 30% in average than the two protocols. Notice that both curves for RREP (Figure 5) and for RERR (Figure 6) are similar to it for RREQ in Figure 3.
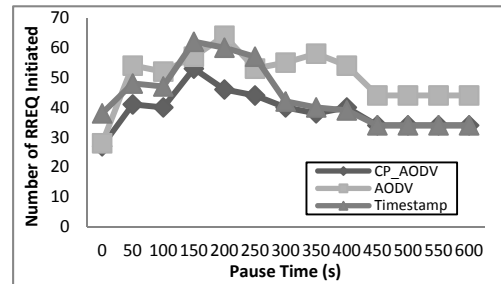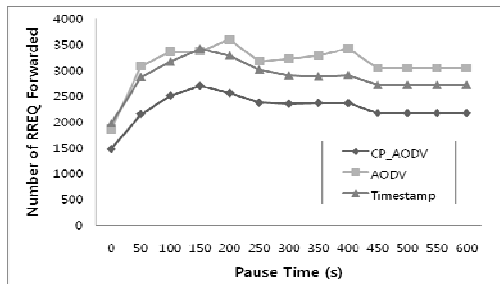
Figure 4. The Number of RREQ Forwarded by Intermediate Nodes
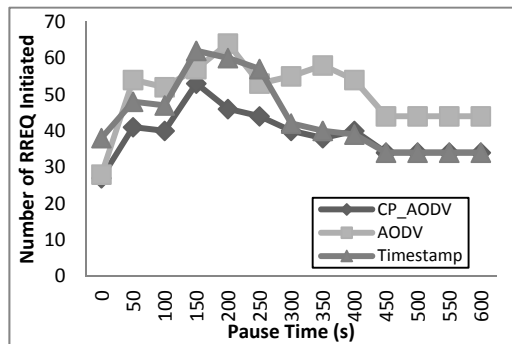
Figure 5. The Number of RREP Packets Initiated by a Destination Node
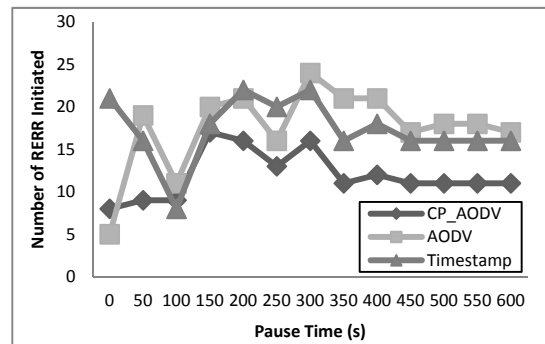
Figure 6. The Number of RERR Packets Initiated by Nodes in the Network

The proposed approach can save the amount of energy consumption by reducing the number of control packets including RREQs, RREPs and RERRs without increase in route discovery latency. The amount of energy consumption by CP_AODV, AODV and T-AODV are plotted against the pause time (i.e., *node mobility*) in Figure 7. Notice that the amount of energy consumption in Figure 7 includes both one for the route discovery process and one for data packet transmissions. Thus the shapes of the curves in Figure 7 are quite different from it in Figure 3. In all levels of node mobility, CP_AODV is remarkably less than AODV.
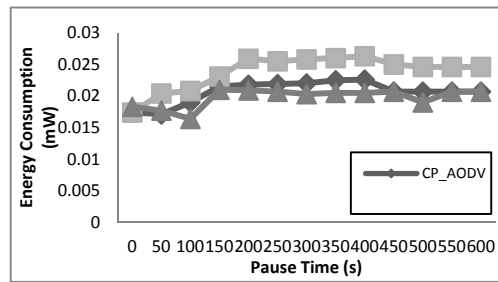
Figure 7. Energy Consumption versus Node Mobility

Notice that the amount of energy consumption is greater than T-AODV by 6% in average. The reason may be attributed to the fact that with T-AODV the number of data packets received by the destination node is relatively small. To be more precise, few observations can be made from our simulation results (Figure 8 and Figure 9). We see that there is no clear distinction for the throughput versus the pause time in Figure 8 for both CP_AODV and T-AODV. For the number of data packets received by the destination node, CP_AODV, however, is significantly higher than T-AODV by 10% as shown in Figure 9. We can say that T-AODV transmits less data packets compared to CP_AODV and thus its energy consumption becomes low.
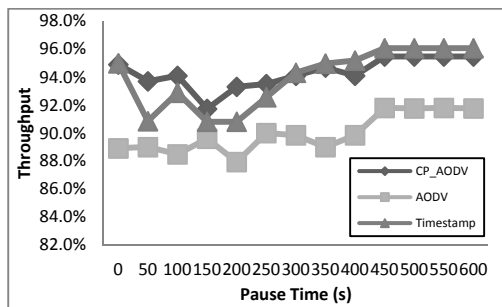

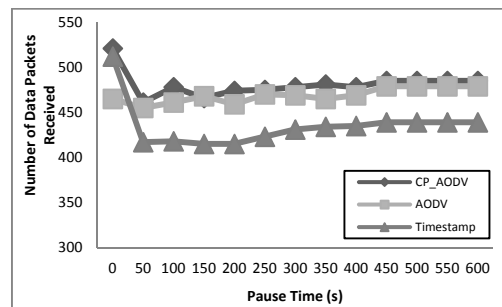Figure 8. The Throughput versus the Pause Time

Figure 9. The Number of Data Packets Received

CP_AODV achieves 5% and 3% better results for the throughput and the number of data packets received, respectively, compared to AODV irrespective of node mobility. From these observations, we can say that substantial reduction in the flooding of the control packets may result in a less route discovery latency. In turn, it causes low data packet traffic to increase the overall throughput.

4.2. The Experiments for Node Velocity

In this experiment each node moves with a nearly constant speed. When each node is set to the speed of 5m/s, it moves with the speed of 4~5m/s. At the experiments in the previous section, it moves with the random speed of 0~10m/s. Thus, this section focuses on the performance evaluation under the node speed ranges. Through this experiment, the pause time is set to 100 seconds and thus the network has a high mobility. The other parameters are used in Table 3. As you see in Figure 10, the average number of RREQ packets of CP_AODV turns out to be better. More specifically, it of CP_ADOV has been measured 58.7, whereas that of AODV and that of T-AODV is 67.9 and 66.1, respectively. Besides, the standard deviation for the number of RREQs of CP_ADOV is the lowest among the three protocols. Thus, we can say that our proposed approach is more stable under the node speed ranges.
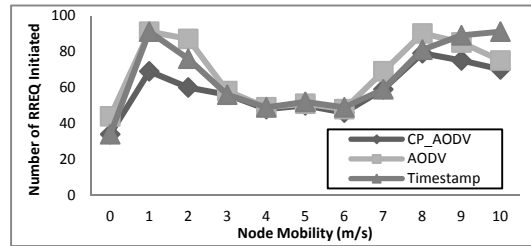
Figure 10. The Number of RREQ Initiated Versus Node Velocity

CP_AODV achieves substantial energy savings by 11~1.7% in average compared to both AODV and T-AODV (Figure 11). In addition, for the number of data packets delivered to the destination it is better than the two protocols by 14~2.4% in average (Figure 12). It cannot say that a network with slower movement of nodes achieves a better throughput than one with faster movement of nodes. In other words, to reduce the volume of control packets that needs to be exchanged between the source and the destination, all of the neighboring nodes keep the distance to exchange messages each other irrespective of node velocity.
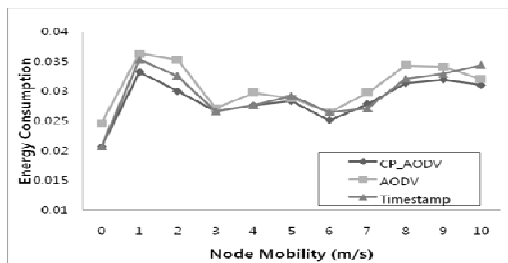

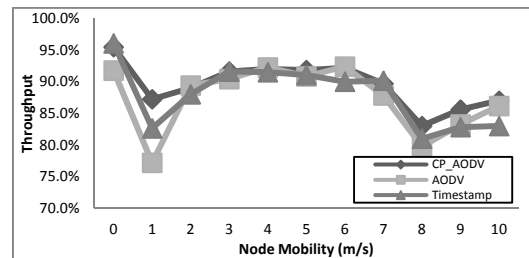Figure 11. The Energy Consumption Rate Versus Node Velocity

Figure 12. The Throughput Versus Node Velocity

4.3. The Experiments for Node Density

In this experiment we vary the number of nodes starting from 36 up to 121 to evaluate the number of RREQ initiated (Figure 13), the amount of energy consumption (Figure 14) and the number of data packets received by the destination (Figure 15). Since the rectangular field configuration is fixed as 1,500m x 1,500m, the distance between each pair of the neighboring nodes becomes shorter with a denser network. Depending on the number of nodes present in the ad-hoc network, we refer to the network as *sparse* which consists of less than 50 nodes. A *dense* network consists of more than 100 nodes. Through this experiment, the pause time is also set to 100 seconds and thus the network has a high mobility. In the following results (Figure 13-15), we omit the results of T-AODV because they are similar to them in the previous sections. Thus, through this subsection, CP_AODV is directly compared to the legacy AODV for simplicity and clarity.
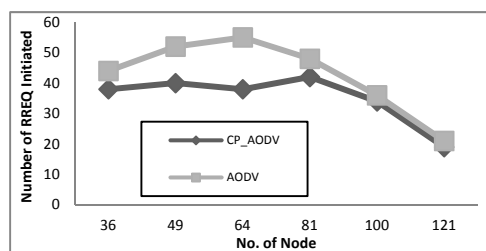

Figure 13. The Number of RREQ Initiated Versus the Number of Nodes

As in a dense network the number of pairs of the two neighboring nodes which locate within a possible transmission range of each other increase, the number of RREQ packets initiated by the source decreases significantly (Figure 13). That means in a dense network the possibility for finding a neighboring node to forward the request packets to the destination node is high. It causes the rate of RREQ packets initiated and forwarded to reduce remarkably. In a dense network, CP_AODV performs similar to AODV as shown in Figure 13. In addition, in a dense network no clear difference between them occurs for the amount of energy consumption (Figure 14).

For sparse and moderate networks, overall performance of CP_AODV is found to be better than AODV. For the number of RREQ packets initiated, CP_AODV is 35.2 in average, whereas AODV is 42.7 in average. For both the amount of energy consumption (Figure 14) and the number of data packets (Figure 15) CP_AODV is better than AODV by 13% and 2% in average, respectively. However, as the node density increases such the differences between them decreases. Thus our proposed method can achieve a better result with a sparse and moderate network than legacy AODV.
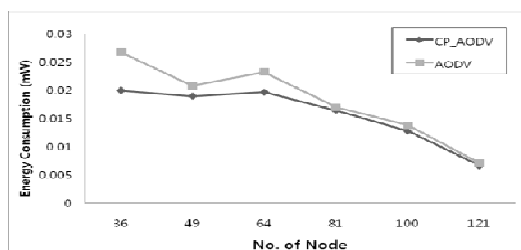

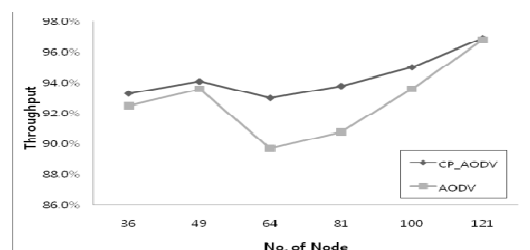Figure 14. The Energy Consumption Rate Versus the Number of Nodes

Figure 15. The Throughput Versus the Number of Nodes

5. Concluding Remarks

We proposed a control packet minimized AODV protocol for rapidly changing the network topology in ad-hoc wireless networks due to node mobility. During the route discovery process, the volume of the control packets increases with the changes in the network topology due to the movement of nodes. To reduce them significantly, more realistic estimation of NTT is needed. To minimize the flooding of control packets, a NTT should be adjusted smoothly based on AIAD technique. CP_AODV protocol provides good improvement in terms of the number of control packets, the amount of energy consumption, the throughput and the number of data packets received by the destination, compared to the legacy AODV and the modified AODV with timestamp technique. These results have been obtained under three distinct metrics; node mobility, node velocity and node density. However, in a dense network, CP_AODV performs similar to AODV, whereas it achieves a better result with a sparse and moderate network.

Acknowledgement

References
[1] S Corson, J Macker. Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501. 1999.
[2] EM Royer, CK Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications Magazine*. 1999; 46-55.
[3] CK Toh. Ad-hoc Mobile Wireless Networks: Protocols and Systems, Prentice Hall PTR. *ACM SIGCOMM*. 2002.
[4] C Perkins. *Performance Comparison of Two On-Demand Routing Protocols for Ad-hoc Networks*. Proc. of the IEEE Infocom. 2000; 3-12.
[5] http://www.faqs.org/rfcs/rfc3561.html, Ad hoc On-Demand Distance Vector (AODV) Routing (RFC 3561)

[6] K Carlberg, J Crowcroft. Building Shared Trees Using a One-to-Many Joining Mechanism. *ACM Computer Communication Review*. 1997; 27(1): 5-11.
[7] Hassan J, Jha S. Optimizing expanding ring search for multi-hop wireless networks. *GLOBECOM*. 2004; 1061-1065.
[8] SM Tripathi, RK Ghosh. *An Implementational Study of Certain Heuristics for the Performance Enhancements of AODV.* Proc. of Euro-Micro Workshop on Parallel, Distributed and Network-based Processing. 2002; 395-402.
[9] J Lee, H Park, K Lee, K Suk. Performance Improvement of AODV Routing Algorithm Using Timestamp. *Journal of Korea Information and Communication Society (domestic)*. 2006; 31(8): 830~836.
[10] Hyun-chang Kim, Suk-moon Chunng. A study of Optimization using state space survey in ad hoc network. *Journal of Korea Institute of Military Science and Technology (domestic)*. 2005; 8(4): 68-76.
[11] Park, Incheon, Pu, Ida. Energy Efficient Expanding Ring Search, 1st ASIA Int. Conf. on Modeling & Simulation (AMS) 2007; 198-199.
[12] http://www.scalable-networks.com