1-12-2006

Node Caching Enhancement of Reactive Ad Hoc Routing Protocol

Sunsook Jung

NODE CACHING ENHANCEMENT OF REATIVE AD HOC ROUTING PROTOCOLS

by

SUNSOOK JUNG

Under the Direction of Alexander Zelikovsky

ABSTRACT

Enhancing route request broadcasting protocols constitutes a substantial part of research in mobile ad hoc network routing. In the thesis, enhancements of ad hoc routing protocols, energy efficiency metrics and clustered topology generators are discussed. The contributions include the followings.

First, a node caching enhancement of Ad-hoc On-demand Distance Vector (AODV) routing protocol is introduced. Extensive simulation studies of the enhanced AODV in NS2 shows up to 9-fold reduction in the routing overhead, up to 20% improvement in the packet delivery ratio and up to 60% reduction in the end-to-end delay. The largest improvement happens to highly stressed situations.

Secondly, new metrics for evaluating energy efficiency of routing protocols are suggested. New node cached AODV protocols employing non-adaptive and adaptive load balancing techniques were proposed for extending network lifetime and increasing network throughput.

Finally, the impact of node clustered topology on ad hoc network is explored. A novel method for generating clustered layout in NS2 is introduced and experiments indicate performance degradation of AODV protocols for the case of two clusters.

INDEX WORDS: AODV, Ad hoc routing protocols, Energy efficiency, Clustered layout

NODE CACHING ENHANCEMENT OF REATIVE AD HOC ROUTING PROTOCOLS

by

SUNSOOK JUNG

Presented in Partial Fulfillment of Requirements for the

Degree of Master of Science

Georgia State University

2005

NODE CACHING ENHANCEMENT OF REATIVE AD HOC ROUTING PROTOCOLS

by

SUNSOOK JUNG

Chair:	Alexander Zelkovsky
Committee:	Sushil Prasad (member)
	Anu Bourgeois (member)

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
August 2005

ACKNOWLEDGEMENTS

I earnestly thank my advisor Dr. Alex Zelikovsky for his excellent guidance and support which made my graduate studies a rewarding experience of my life. His continuous interest of the research was a great source of motivation on me. I thank committee members, Dr. Sushil Prasad and Dr. Anu Bourgeois for their comments and taking precious time to review this thesis. Also, I am grateful to Nisar Hundewale for his help during my graduate studies.

Many thanks to my parents and parents-in-law who gave me a great chance to study in America with deep concern and love. Finally, special thanks to my husband, Hojoon for his thoughtful support.

Table of Contents

List of Figures

1. Introduction

Mobile ad hoc network (MANET) is a special type of wireless network in which a collection of mobile network interfaces may form a temporary network without the aid of any established infrastructure or centralized administration. Recently, enhancing route request broadcasting protocols constitutes a substantial part of research in mobile ad hoc network routing. In this thesis, we suggest a novel approach to constrain route request broadcasting and evaluate performance with various aspects.

1.1. Ad hoc Wireless Network

Ad hoc wireless network has applications in emergency search-and-rescue operations, decision making in the battlefield, data acquisition operations in hostile terrain, etc. It is featured by dynamic topology (infrastructureless), multi-hop communication, limited resources (bandwidth, CPU, battery, etc.) and limited security. These characteristics put special challenges in routing protocol [1].

Several routing protocols have been suggested since late 90's. DSR, AODV, DSDV, and TORA are representative (see [2] for comprehensive review of these protocols). These protocols do not reply on node locations or relative locations of other nodes. It implies that mobile nodes only depend on the information collected by routing protocols. Due to the node mobility, frequent exchanges of control packets for location updates result in significant communication overhead.

We focus on enhancing route request broadcasting protocols constituting a substantial part of the MANET routing. A simple flooding broadcast for route requests generates a considerable redundant packet overhead which is a major cause of inefficiency of MANET routing protocols. Several broadcasting techniques are compared in [7] and [8] concluding that neighbor-knowledge based broadcasting is better probabilistic and area based methods in reducing packet redundancy. Another interesting approach constrains the number of detours or deviations from the known routes resulting in 50% overhead and delay reduction but insignificant decrease in delivery ratio for DSR [9]. The AODV protocol has been enhanced in [10] by pruning dominant nodes, i.e. effectively constraining route requests to a certain connected dominated set.

1.1.1. Node caching in ad hoc networks

We suggest a novel approach to constrain route request broadcast which is based on node caching. By dropping route request packets based on the cached information about recent data packet forwarding, this protocol reduces the routing overhead significantly. As well as previous approaches, node caching also employs the fact that the broadcast for route request is not really a broadcast – it does not need to reach all nodes but only a single required destination. Therefore, we drop route requests forwarding from the nodes which are not cached at the expense of possible destination missing.

Our node caching techniques can be also viewed as a dynamic implementation of a connected dominating set (CDS) based routing. Indeed, the cached nodes are supposed to cover the recent sources and destinations and are mostly connected by recent intersected paths. The known drawback of CDS is overuse of dominating nodes. We suggest ensuring the protocol fairness using parameters of forwarding load distribution among MANET nodes. We confirm that node caching may cause unfair forwarding load distribution and propose load-balancing schemes for fixing this drawback. An evaluation of routing protocol fairness measured as distribution of forwarding load among nodes

1.1.2. Energy Efficiency in MANET routing protocols

The primary objectives of MANET routing protocols are to maximize network throughput, to minimize energy consumption, and to minimize delay. The network throughput is usually measured by packet delivery ratio while the most significant contribution to energy consumption is measured by routing overhead which number or size of routing control packets.

However, due to the limited energy supply in the mobile node in MANET, maximizing network lifetime should be also considered. Network lifetime is defined as the time from beginning of simulation until first node in MANET runs out of energy. Network lifetime is an important factor choosing a routing protocol because of characteristics of MANET. Therefore, energy efficiency with limited energy amount should be considered as well as routing efficiency in performance evaluation.

Also, maximizing network lifetime implies that nodes in MANET are utilized fairly to deliver packets. To maintain impartiality, load balancing schemes would be effective.

In the thesis, we present the new performance metrics in regard to energy efficiency as well as several load balancing algorithms which relieve overused nodes.

1.1.3. Layout and Movement of Nodes in MANET

In ad hoc network, the main factors which affect the performance of routing protocol are the speed of node movement and the traffic intensity of communication. In addition to these factors, we consider the topology of nodes. In many experiments of MANET routing protocols, “random waypoint layout” is commonly used as node movement model. However, in real world, randomly and uniformly distributed layout rarely happens. Nodes tend to be placed to the specific areas that need to be monitored. As a result, it forms clusters. This clustered layout is suggested by V. Kawadia et al in [15]. They named the randomly distributed layout in space as homogeneous layout while clustered layout as non-homogeneous layout. In the paper, they suggested several protocols which are related to MAC layout control; by controlling the transmission power to overcome this special problem. In our thesis, we suggest the clustered layout movement as a new factor of simulation and observe the performance impact of node clustering in MANET routing.

1.2. Contributions

This thesis is devoted to enhancement of AODV by reducing the number of control packets. Also, we discuss the limitation of our new protocol and introduce load balancing schemes to solve the problem. New approaches to evaluate performance of routing protocols are introduced with respect to performance metrics and new node movement scenario.

An extensive simulation study of AODV-NC in Network Simulator (NS-2) shows that in case of highly stressed MANET the routing overhead is reduced by average 85%, the delivery ratio is increased by average 20%, and the end-to-end delay is decreased by average 63%.

An implementation and simulation study in NS-2 of forwarding load balanced AODV-NC sustains considerable improvement in overhead, delivery ratio and delay over the standard AODV.

In addition to performance evaluation such as routing overhead, delivery ratio and end-to-end delay, we measure energy consumption, network throughput and network lifetime with enhanced AODV. AODV-NC with load balancing shows the best network throughput and network lifetime with limited power.

The last contribution is the performance evaluation of AODV and AODV-NC with node clustered layout model applied to node movement. We present a novel cluster layout generator that generates clusters based on the input parameters provided by the user and show the performance impact of node movement models in routing protocols.

1.3. Roadmap

The rest of the thesis is organized as follows: in the next section, we describe the details of route discovery in AODV protocol and compare performance between AODV and DSR protocols. In section 3, we describe the node caching enhancement with AODV (AODV-NC). Also, we compare performance AODV-NC and DSR-NC. In section 4, we discuss fairness of routing protocol and suggest load balancing schemes. Section 5 explains simulation details as well as performance metrics. In section 6, we present the results of our simulations in NS2. In Section 7, we describe the modification of source code in NS2. It includes AODV-NC, AODV-NC with load balancing and cluster topology generator. Finally, section 8 concludes the thesis.

2. Ad-hoc On demand Distance Vector (AODV) Routing Protocol

AODV [2] is a reactive and hop-by-hop routing protocol which means that every node finds routes by demand and every intermediate node decides where the routed packet should be forwarded next. Therefore, each of nodes in AODV has a route table to maintain fresh route information which has three essential fields: a next hope node, a sequence number, and a hop count. All packets destined to the destination are sent to the next hop node. The sequence number acts as a form of time-stamping, and is a measure of the freshness of a route. The sequence number is increased in its originating node. The hop count represents the current distance to the destination node.

AODV protocol consists of three types of messages: Route Requests (RREQs), Route Replies (RREPs) and Route Errors (RERRs). RREQ message is broadcasted throughout the ad hoc network when a route to a destination is needed and RREP message is sent back to the originator from a destination or intermediate nodes which have fresh route to the destination. And RERR message is used to notify other nodes the broken links.

2.1 Routing Discovery

When a node needs to send or to forward a data packet to a destination, it checks the route table to determine whether the route between the endpoints of communication connection is valid or not. If the route is valid, a node sends a data packet to the next hop node. Otherwise, it begins route discovery with a RREQ packet by broadcasting. A RREQ packet includes the source node’s IP address, broadcast ID, current sequence number of itself and the sequence number of the destination. Nodes receiving the RREQ update their information for the source node and set up backwards pointers to the source node in the route table. The RREP is made when the RREQ reaches either the destination itself, or an intermediate node with a valid route to the destination. A valid route is the route entry whose sequence number is greater than that contained in the RREQ. And the real route is made by unicasting a RREP to the origination of the RREQ. Since each node receiving the RREQ caches a route back to the origination for the RREQ , the intermediate nodes send back RREP to the its previous node and become a part of route to the destination. If they receive the same RREQ packets later, they discard and do not forward it. In case of a source node, it updates the entry in the route table and uses this route the next time.

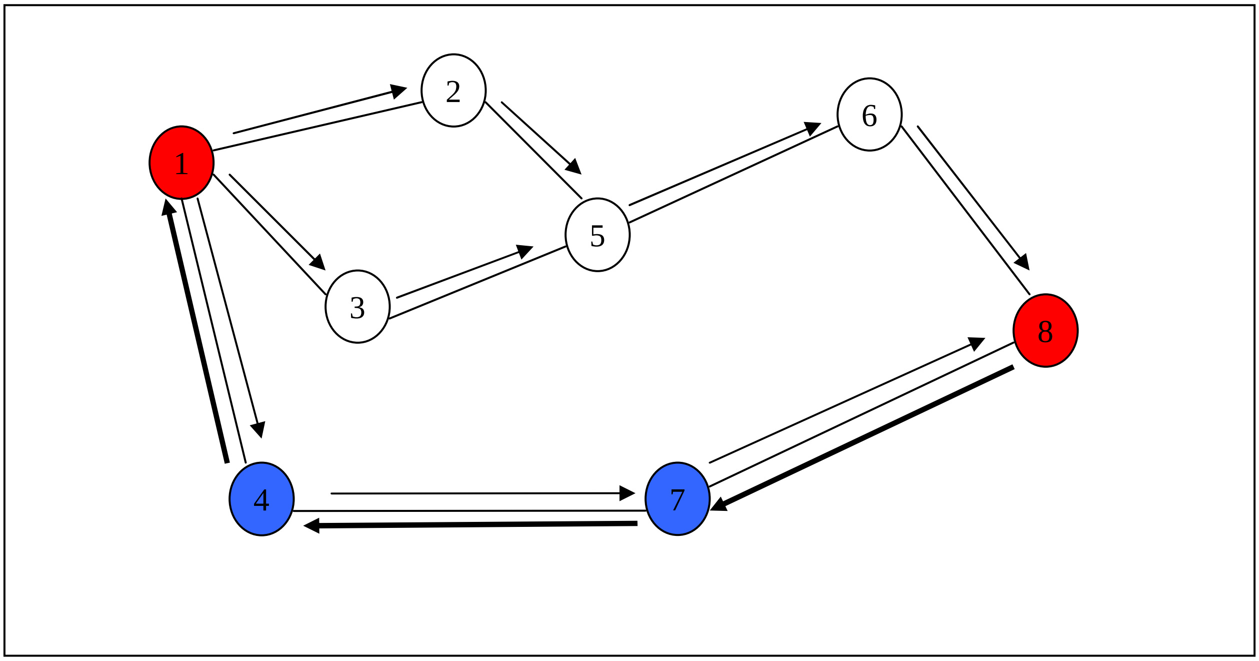


Figure 2.1.1. AODV route discovery

2.2 AODV vs. DSR

Dominating Set Routing (DSR) [13] is commonly compared with AODV. Even though DSR is a multi-hop protocol and reactive protocol, route discovery mechanism is different. The most prominent difference is that DSR uses the source routing in which each packet contains the route to the destination in its own header. Therefore, intermediate nodes do not need to maintain up-to-date routing information in order to forward data packets. Another unique feature of DSR is packet salvaging. When an intermediate node detects the broken link to the next hop, the node begins to find an alternative route instead of discarding the data packet. In our experiments in NS2, we found that the packet salvaging causes the extension of end-to-end delay.

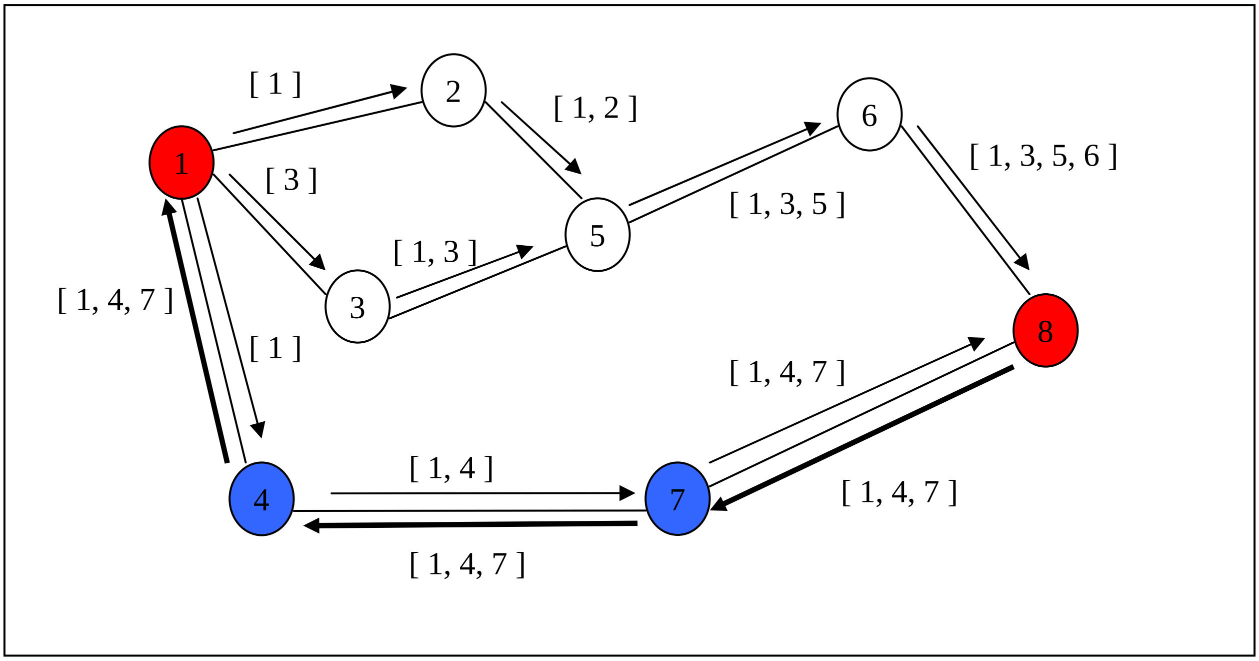


Figure 2.2.1. AODV route discovery

In case of less stressed situation (i.e. smaller number of nodes and lower load and/or mobility), DSR outperforms AODV in delay and throughput but when mobility and traffic increase, AODV outperforms DSR [5]. However, DSR consistently experiences less routing overhead than AODV. A hybrid protocol enhancing AODV with the advantageous route caching feature of DSR is proposed in [6].

3. Node Caching Enhancement of AODV (AODV-NC)

In this section, we discuss the main drawback of reactive routing protocols and describe node caching enhancement of route request broadcasting in AODV and give implementation details of node caching AODV

3.1 Motivations of Enhancement

The main drawback in reactive routing protocols is that a route request packet is simply broadcasted across the network even though some nodes will be not involved the route to the destination after route discovery. It generates a considerable redundant packet overhead which is a major cause of inefficiency of MANET routing protocols.

Our intuition of node caching is that the nodes involved in recent data packet forwarding have more reliable information about its neighbors and have better locations (e.g., on the intersection of several data routs) than other MANET nodes. We cache nodes which are recently involved in data packet forwarding, and use only them to forward route requests. As a result, node caching employs the fact that the broadcast for route request is not really a broadcast - it does not need to reach all nodes but only a single required destination. Therefore, we drop route requests forwarding from the nodes which are not cached at the expense of possible destination missing.

As mentioned in the previous section, we want to cache connected and dominating set of nodes that have updated information about their neighbors while wasting no resources for finding and maintaining the cache. All these requirements are very well satisfied by the nodes which have recently forwarded data packets. Indeed, a union of source-destination paths with multiple intersections is well connected and dominates almost all nodes since such nodes are mostly in the center of the network. Of course, such set does not require any maintenance.

3.2 Modifications to the AODV protocol

To implement node caching, we use a fixed time threshold parameter *H*. The first route request is sent with the small threshold *H*. When a node *N* receives the route request, it compares the current time *T* with the time *T(N)* when the last data packet through *N* has been forwarded. If *T-H > T(N)*, then *N* does not belong to the current node cache and, therefore, *N* will not propagate the route request. Otherwise, if *T-H > T(N)*, then *N* is in the node cache and the route request is propagated as usual. Of course, the node cache cannot guarantee existence of paths between all source-destination pairs. Therefore, if the route request with the small threshold *H* fails to find a route to destination, then a standard route request (which is not constrained by cache) is generated at the source.

In the default settings of AODV, if the route to the destination is broken, obsolete or not established, the route request originated from the source is propagated through the entire MANET. If the route reply is not received by source in a certain period of time, then the route request is periodically repeated several times. If all these Route Requests happened to be unsuccessful, several more requests with increasing time gaps are sent.

In NS2, we tried to avoid drastic changes to the very well established AODV protocol. We restrict modifications to the initiation of Route Request packet and the forwarding of Route Request packet to neighbors.

Initiation of the Route Request

With time threshold *H*, a route request is initiated as followed.

(1) If a requested route is not available, then send an *H*-restricted route request with the threshold *H*.

(2) Repeat *H*-restricted route request 2 times if route reply is not received during time 0.3 sec after sending a route request.

(3) If no route reply received, then send an unconstrained (standard AODV) route request with the standard repetition pattern.

Forwarding the Route Request packets

When receives the route request packet, each route request recipient *N* does followings.

(1) If *N* has fresh route information to the destination, it begins to send a route reply packet.

(2) Otherwise, N does

a. if no more than *H* seconds are gone from the last time a data packet has been forwarded by *N*, then *N* rebroadcasts the route request to all its neighbors.

b. otherwise, *N* drops the route request packet.

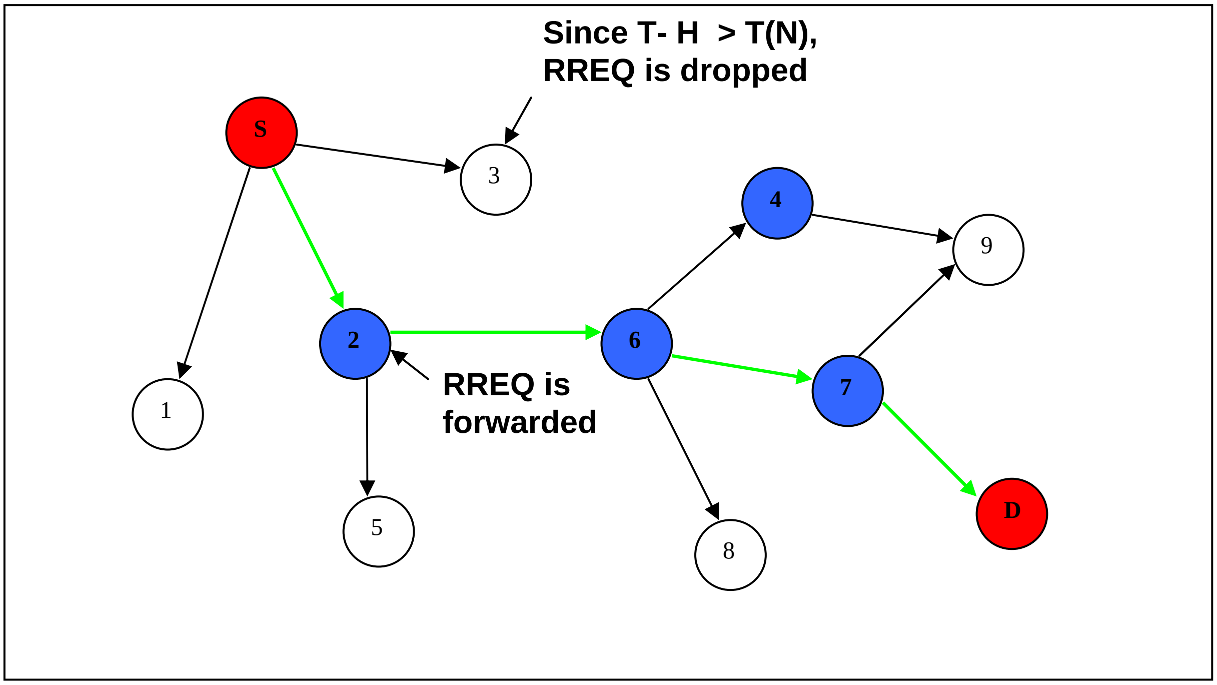


Figure 3.1.1. Route Request packet forwarding with AODV-NC

We did not attempt to find the best initial threshold *H* theoretically. Our simulations show that on average the best choices of *H* are between 0.1 sec and 1 sec. If we would know in advance MANET parameters, then we can tune threshold more carefully - higher traffic intensity and mobility level correspond to the smaller threshold.

The value of *H* directly affects the hit ratio of the node cache, i.e., the fraction of cache-constrained route request attempts succeeded to find the destination over all cache-constrained requests. Figure 3.2.1.illustrates our simulations with different values of *H* - larger *H* corresponds to larger hit ratio. The value of *H* is also inversely proportional to cache size, average number of nodes forwarding a route request (see Fig.3.2.2.). Note that the standard route request will be forwarded by all nodes except source and destination.

3.3. AODV-NC vs. DSR-NC

C. Sha et al [21] present node caching enhancement to DSR. The authors approach this idea with Connected Dominating Set (CDS). Since both DSR and AODV are reactive protocols, redundant control packets produced by broadcasting is a common drawback. Our node caching was well combined with DSR and showed significant performance improvement.

Extensive experimental study in NS2 shows that DSR-NC decreases the routing overhead by average 24 % and increases the packet delivery ratio by average 7 % with high node mobility. Also, the average number of hops is decreased by average 1.5 %. On the contrary, AODV-NC shows the average decrease by 85 % in the routing overhead, average increase by 20 % in the delivery ratio and average decrease by 63 % in the end-to-end delay. From the comparison between DSR-NC and AODV-NC, we can see that node caching is more effectively integrated with AODV than DSR and conclude that AODV-NC has better performance than DSR-NC because AODV and DSR have similar performance results.

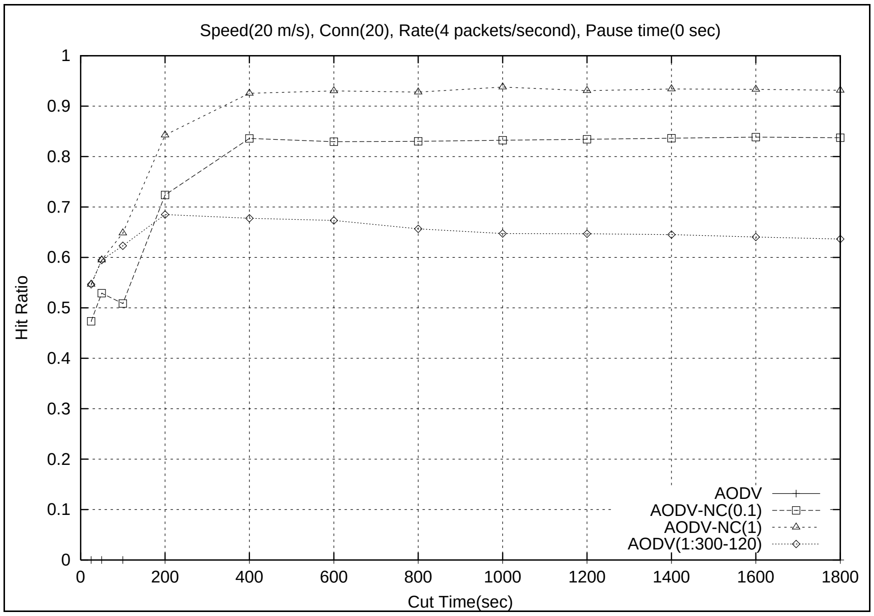


Figure 3.2.1. Average success rate of cache-constrained route requests

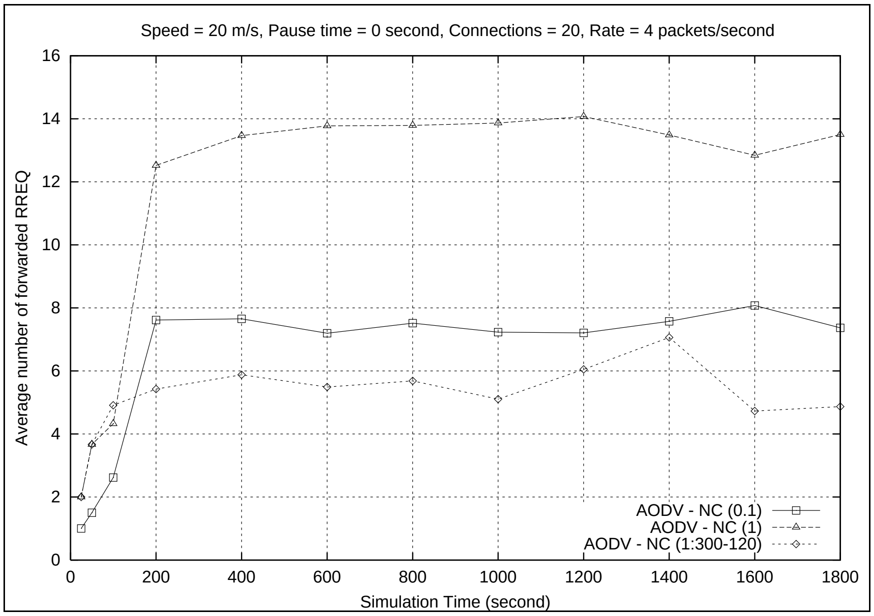


Figure 3.2.2. Average number of forwarding nodes per cache-constrained route request

4. Fairness of Routing Protocol and forwarding load balancing

In this section, we discuss fairness of routing protocols and show how fairness can be measured. Also, we compare AODV and AODV-NC protocols with respect to fairness and suggest load-balancing schemes to improve fairness of AODV-NC as well as network lifetime.

4.1 Drawback of AODV-NC

As we have mentioned in Section 1, the node cache can be viewed as CDS. While being very efficient, CDS-based protocols can overexploit the nodes which belong to CDS. This results in reducing *network lifetime*, the time between beginning of operation and the first node exhausts its batteries (assuming, e.g., equal battery supply for all nodes). We can also look at this phenomenon from *fairness* prospective as follows[1]. Each node by joining an ad hoc network is required to support multi-hop communication, i.e., forward data and control packets upon request. If certain nodes are unlucky enough to forward too many of such packets, then they can claim unfairness of the network protocol and drop membership. Note that it does not matter how many packets are by a node - if a node sends too many packets, then it is fair that such node pays the corresponding energy amount. Only forwarding load is fair to account for.

For measuring unfairness, we use the ratio of the maximum forwarding load among individual nodes over the average forwarding load. We exclude the number of sent packets since it does not depend on the routing protocol rather it depends on the scenario. Note that the absolute value of the maximum forwarding load is also important - if the network lifetime is considerably larger for one protocol than for another, then fairness ratio loses its relevance.

[1] The first attempt to measure fairness of CDS-based routing uses a different approach [10].

Figure 4.1.1. illustrates distribution of forwarding loads among 50 nodes for several protocols - the range between 0 and 4000 Kbytes is partitioned into 250 Kbytes subintervals and the number of nodes forwarding the load from each subinterval is reported. We can see that AODV is fair - its fairness ratio is below 1.7 – while AODV-NC(1) and AODV-NC(0.1) are unfair because of a bump at 4000. For the low mobility situation in Figure 4.1.2., unfairness is occurred for every protocols but the amount of traffic is not as large as high mobility situation.

4.2 Simple load balancing scheme

In order to prevent unfairness of node caching, we should relieve nodes which stay in cache for too long time. Several geometric models have been proposed [11], [12] to impose load balancing. Our simple scheme balances the control and data packet forwarding load without using geometric knowledge of the network. We suggest a load-balancing scheme AODV-NC (*H*: *n* - *t*) with the following two additional parameters - the threshold number of packets *n* forwarded during time *t*. If number of data packets forwarded by a node *N* during time period *t* is greater than *n*, then we relieve the node *N* from forwarding cache-constrained route requests for the same time period *t*. During the break *t*, the node *N* still forwards data packets as well as standard unconstrained route requests. But the forwarding load for *N* is decreased since new routes with high probability will avoid *N*. In Figure 4.1.1., one can see that AODV-NC (1:300-120) is almost as fair as AODV.

4.3 Workload based load balancing scheme

However, the forwarding-load balancing algorithm is not self-adaptive because the proper parameter values are found through several experiments. It means that the value of

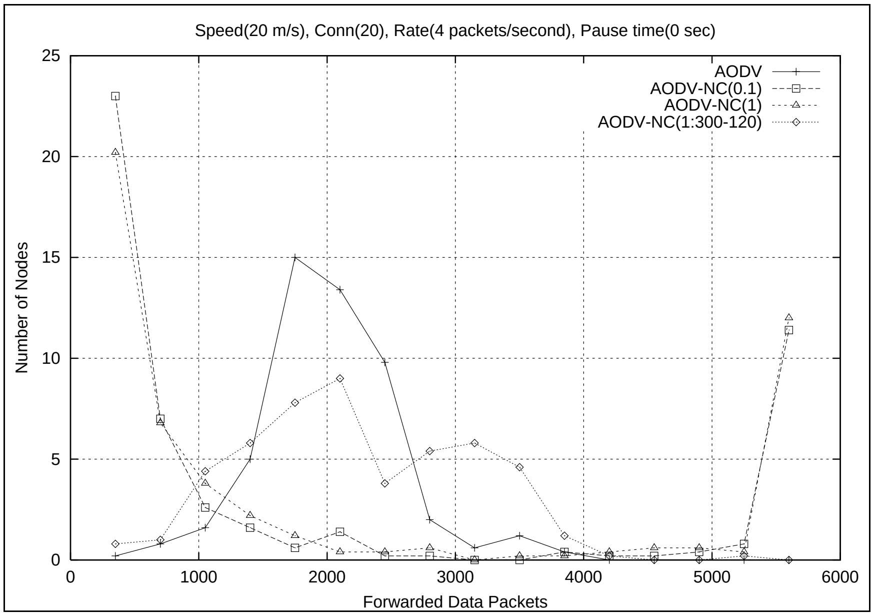


Figure 4.1.1. Distribution of the forwarding load among 50 nodes for high traffic and mobility.

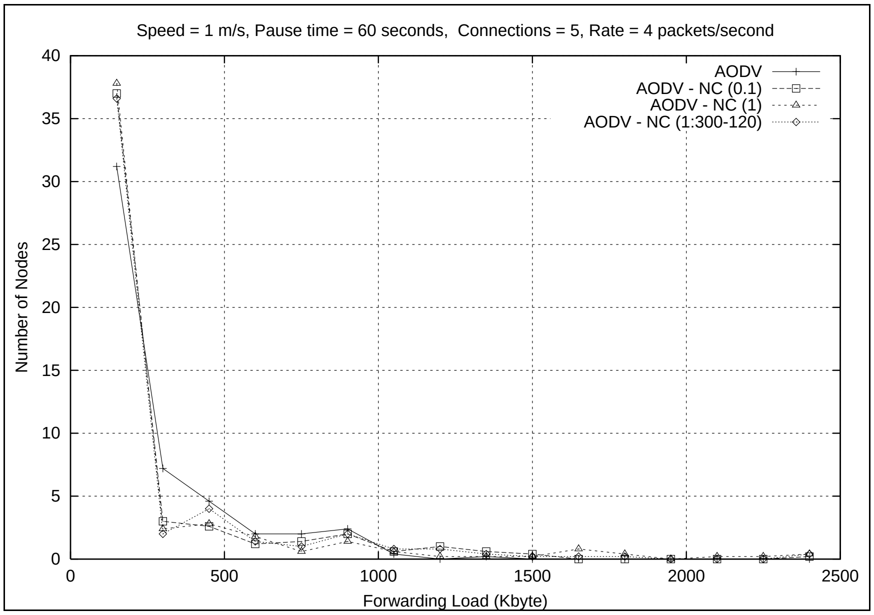


Figure 4.1.2. Distribution for the forwarding load among 50 nodes for low traffic and mobility

parameters should be changed for every different situation.

Lee et. al. presented in [14] workload-based adaptive load balancing technique that is based on the idea that by dropping rout request packets according to the load status of each nodes. nodes can be excluded from route paths. This algorithm uses the length of the message queue in nodes and the outstanding workload which is defined as the combination of the queue length and residence time of packets in the queue. At the beginning of simulation, the minimum and maximum lengths of message queue and workload threshold are initialized. When a node receives RREQ packets, it checks the length of queue and calculates the average of two thresholds values. And then, a node calculates outstanding workload. If queue length is greater than the average threshold value and outstanding workload is greater than workload threshold, it drops RREQ packets. Otherwise, the node forwards the RREQ packet to neighbors. In the meanwhile, a node calculates new threshold value of message queue length if outstanding workload is greater than workload threshold. According to the workload, threshold values are changed automatically.

We present AODV-NC-WLB that is a combination of the adaptive workload balancing technique with the node cached routing protocol. The forwarding-load balancing technique from AODV-NC (H: $n - t$) can not be combined with the adaptive workload balancing technique. The node caching technique is orthogonal to the adaptive workload balancing technique that allows combining them without making major changes.

We applied workload-based adaptive load balancing (WLB) with AODV and AODV-NC and obtained the following workload distributions. In simulations, we used the different numbers of connections to see how WLB works in various workload situations. With AODV, WLB relieves heavy workloads of approximately 18 nodes by dispersing forwarding packets to other nodes in most cases. In Figure 4.3.1., 4.3.2. and 4.3.3., the long bell shape graph of the AODV are changed to wild bell shape graph with WLB. However, it does not solve the original unfair load balancing limitation of AODV-NC. Instead, it relieves nodes in high workload. In Figure 4.3.1., we still see the bumps with AODV-NCs around 4000 Kbytes in x axis. But, the bumps get lowered in Figure 4.3.2. and 4.3.3.. By dropping packets in message queues of busy nodes, WLB avoids the chance that node might join the forwarding route.

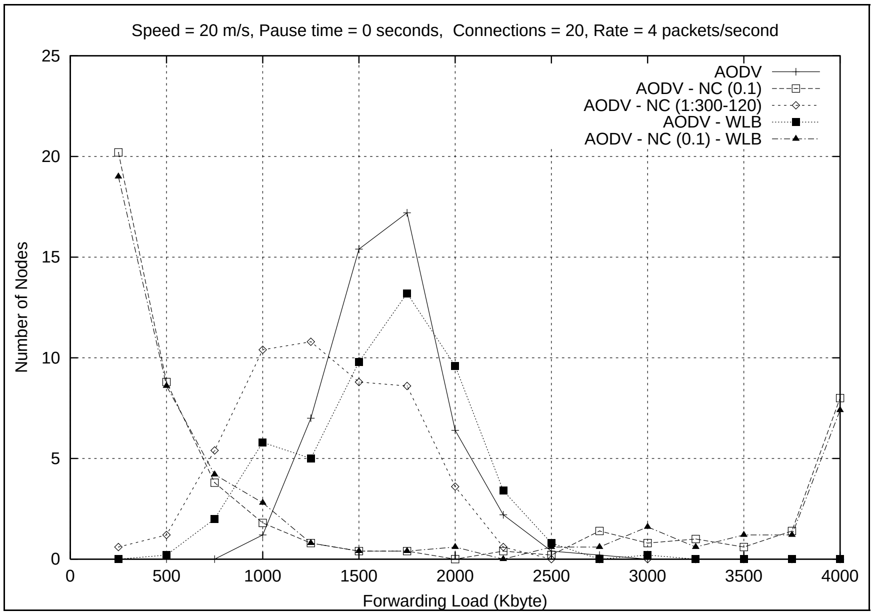


Figure 4.3.1. Distribution of the forwarding load among 50 nodes for 20 connections

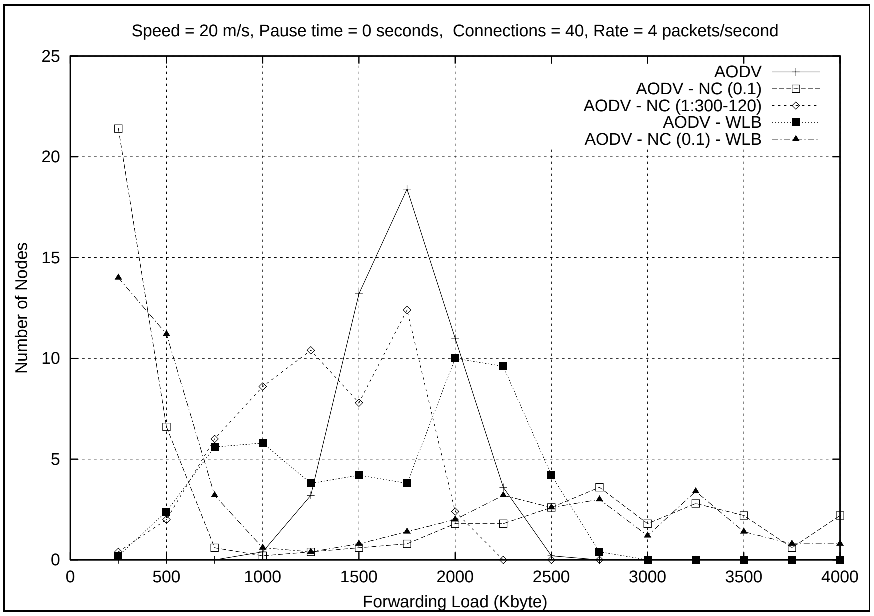


Figure 4.3.2. Distribution of the forwarding load among 50 nodes for 40 connections

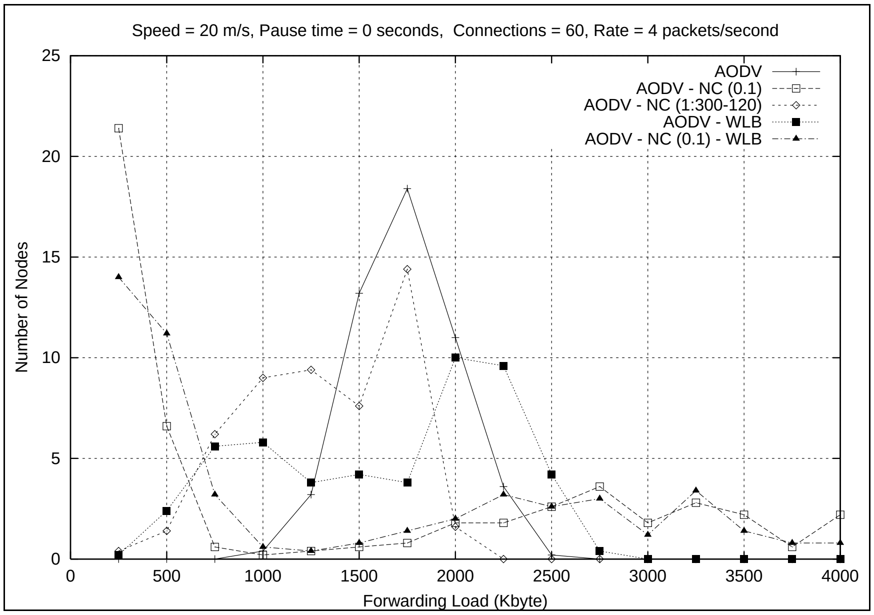


Figure 4.3.3. Distribution of the forwarding load among 50 nodes for 60 connections

5. Scenarios

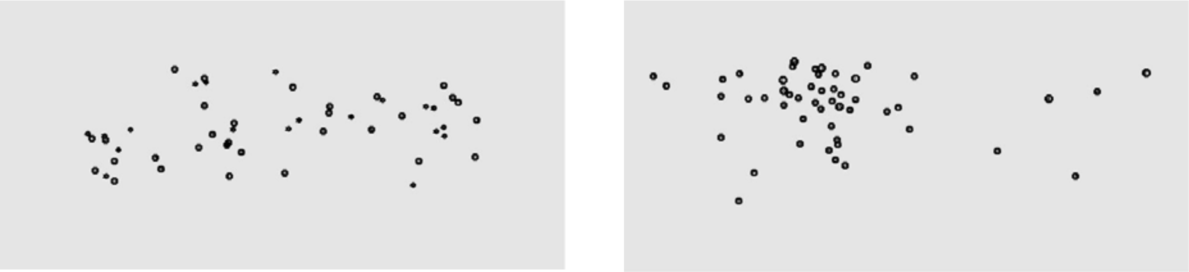
In this section, we describe the details of scenarios used for all simulations. First, we describe basic values of parameters for overall simulations in Network Simulator (NS2 version 2.26). And then we explain node movement model, communication model and new performance metrics for measuring energy efficiency of routing protocols in MANET.

Most simulations were performed with 50 wireless mobile nodes forming an ad hoc network and moving over a rectangle area of 1000m x 300m. But, we also repeated simulations for 1500m x 300m and 1000m x 1000m rectangles. The simulation time was 900 sec or 1800 sec and we have recalculated the basic parameters each 100 sec. We fixed the radio transmission range as 250m and used 802.11 MAC protocol for physical layer.

We experimented with different threshold values of H = 0.01, 0.05, 0.1, 1, 5 and 10 for AODV-NC (H) and found that H = 0.1 and 1 demonstrate the best performance. So, for AODV-NC protocol, we set time threshold values with 0.1 and 1. For AODV-NC (H: n – t), we set threshold value of H = 1 and the number of forwarded packet threshold n = 300 during time t= 120 sec. Also, we initialized five parameters for AODV-WLB with the exactly same values as in [14].

5.1. Node Movement Model

We experiment two different movement models: “random waypoint” model and new “clustered layout” model. In random waypoint model, nodes are uniformly distributed over the rectangular area. On the contrary, in clustered layout model, mobile nodes establish clusters around certain points. Figure 5.1.1. shows the snapshots of two different node movement models.



(a) random waypoint model

(b) clustered layout model

Figure 5.1.1. Snapshots of random and clustered model

The idea of applying the clustered layout model to MANET routing protocol is novel. Clustered topology is well-known phenomenon in Internet Network area, called power laws distribution or heavy tail distribution, while it is uncommon in ad hoc network. In the following subsection, we discuss random waypoint model and then present the cluster layout model closely.

5.1.1. Random waypoint movement

Random waypoint model generated by *setdest* program in NS2 is commonly used for ad hoc network protocol evaluation. This model was first introduced in [13] for the evaluation of DSR and has become the standard node movement model in the area of ad hoc network. For our simulation, we generate various movement scenarios with maximum node speed from 1 m/s to 50 m/s.

However, J. Yoon et al [20] has pointed out the problem of random waypoint model because it failed to maintain the average speed of nodes over time and resulted in incorrect simulation results. In addition, this model does not reflect the real world situation. For example, in disaster area, most of mobile nodes tend to be placed around a certain places where several damages occurred instead of being placed uniformly. As a result, nodes establish clusters corresponding to power-law distribution while random waypoint layout is established corresponding to binominal distribution.

5.1.2. Clustered layout Movement Model

As mentioned above, Internet topology described by power law distribution has been considered as one of main properties of Internet topologies [18]. Several topology generators such as GT-ITM [22], Inet [23] and BRITE [19] and others have been studied to generate topology of Internet nodes. Among them, BRITE generates clustered topology using bounded Pareto distribution. The idea in BRITE is that the plane is divided into several high-level squares and each one of these high-level square is subdivided into smaller low-level squares. And then the generator picks the number of nodes n to be assigned to that high-level square according to a bounded Pareto distribution. Finally, a node is placed randomly in one of the low-level squares. The snapshot of random node placement and heavy-tailed node placement are shown in [19]. In their results, we found that clusters were forming small sub-squares rather then natural shapes. Instead of unnatural quadrilateral clusters, we tried to generate natural-shaped clusters. Of course, our clustered layout model was inspired by the power law distribution which forms $y = x^{\alpha}$. We redefined the power law as $y = \frac{1}{x^{\alpha}}$, where $0 < \alpha$. The algorithm generating clusters is described below. Another advantage of our generator is not only generating clustered layout but also generating the clustered movement scenario suitable for NS2. Instead of staying in one place during the whole simulation, nodes move to other places and form clusters in the destination.

Clustered Layout Generate Algorithm

The main parameters of the generator are the number of nodes *N*, the number of cluster C, and the value of α of power-law. These values are assigned by users. Our algorithm consists of largely two parts: (1) find positions of nodes N and (2) match N with destinations.

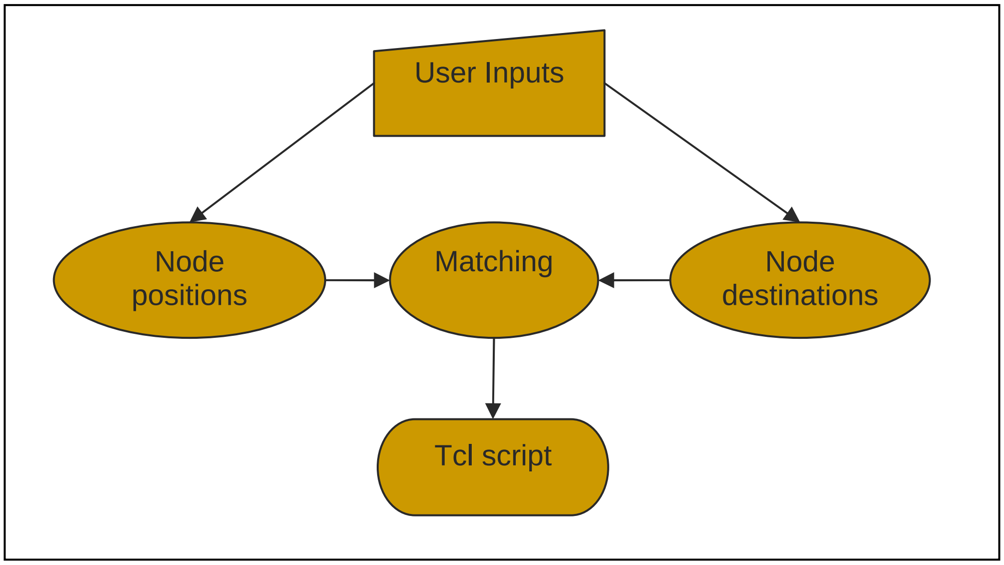


Figure 5.1.2.1. Clustered layout generator diagram

Finding positions of N nodes

(1) Randomly distribute a quarter of N on the network area.

(2) Calculate the number of neighbors of each node and sort them in descending order.

- we restrict neighbors as nodes within transmit power range 100m to avoid all of nodes having the same number of neighbor.

(3) Choose C nodes from sorted nodes as the center of clusters.

- To avoid cluster centers too close each other, we do not choose C nodes sequentially from the sorted nodes.

(4) Create dummy nodes of N * 2 and assign positions randomly.

(5) For N*2 nodes, repeat the following module.

a. Calculate the Euclidean Distance D_i between cluster centers and each dummy node *i*.

- If C is larger than 1, choose the shortest distance among them.

b. save the $\frac{1}{D_i^{\alpha}}$ to a variable *S*. Finally, S will form the formula, $S = \sum_{i=1}^{N*2} \frac{1}{D_i^{\alpha}}$.

- S can be considered as the sum of probabilities. It will be used to choose nodes from dummy nodes. Therefore, as α becomes larger, the probability becomes smaller. As a result, the clusters become denser.

(6) Until find three quarters of N, repeat the following module.

a. pick a random number between $0 < r < S$.

b. find a node which satisfies the formula, $D_i = D_i - r$ and $D_i \leq 0$, from dummy nodes.

Matching nodes with destinations

(1) Execute the finding node position module for N * 2 dummy nodes.

(2) For N nodes, repeat calculating the Euclidean Distance D_i between position of node *i* and dummy nodes.

a. check whether D_i is a reachable distance with the maximum speed of a node.

b. if it is reachable, register the node as a candidate for the possible destination. Otherwise, go to the next dummy node.

(3) Calculate the number of candidates and sort in ascending order.

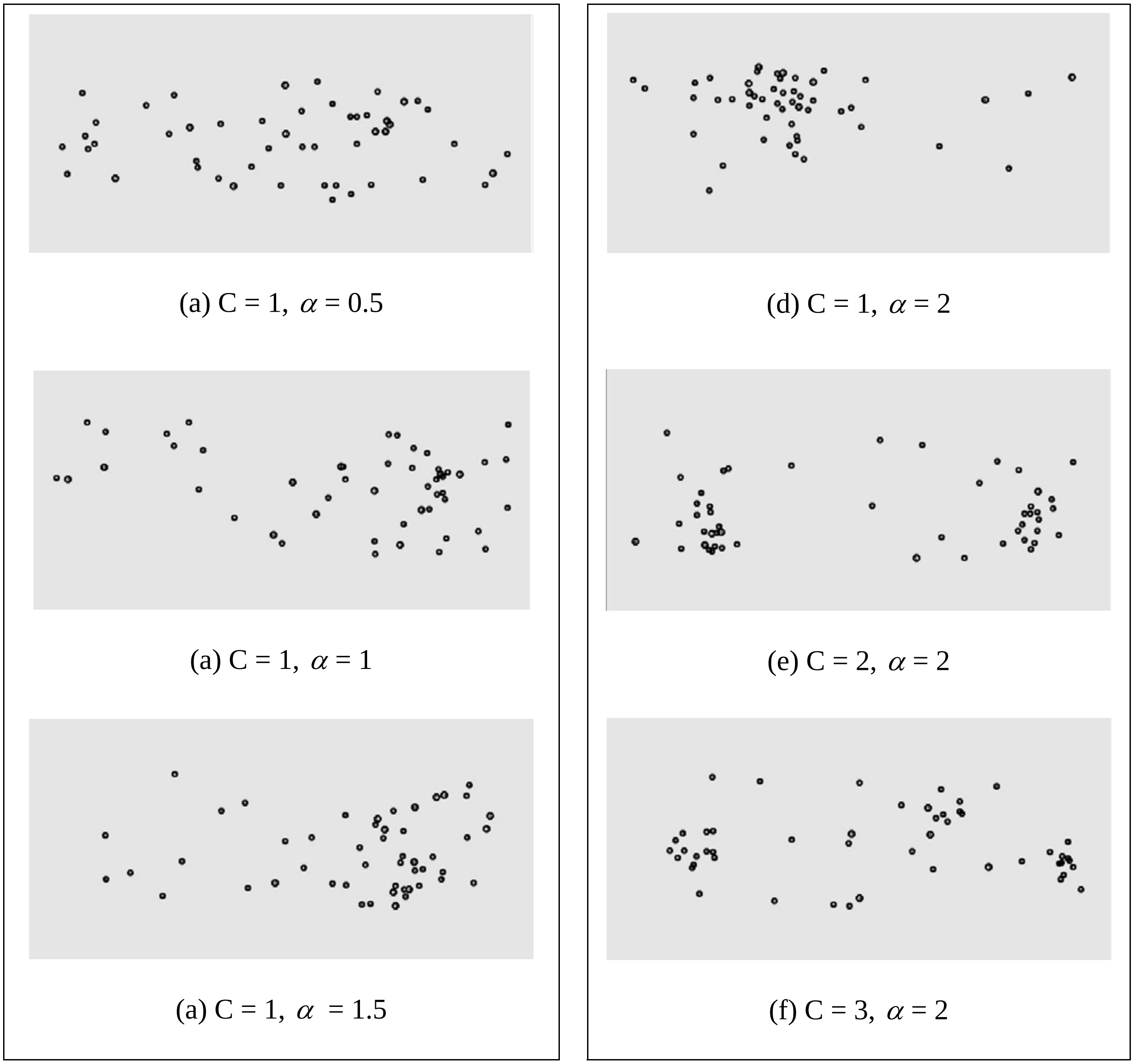
(4) Start matching with the node which the smallest number of candidates.

- to avoid the worst case such as nodes which have a few candidates fail to find the pair, we start matching with these nodes first.
- if nodes fail to, we assign the randomly generated position.

In order to make additional data for NS2, e.g. GOD, we reused the functions in *setdest* program. The usage of our generator is described in section 7.

The snapshots of clustered layout are shown in Figure 5.1.2.1. As wee can see, for different values of α , from 0.5 to 2, the density also vary in a cluster. In Figure 5.1.2.1, (a), (b), (c) and (d) have one cluster but the number of nodes in the cluster is totally different. As α increased, the density goes high. (d), (e) and (f) show various numbers of clusters of each rectangular area. The difference is really obvious.

After several experiments, we fix the value of α with 2 for performance evaluation because it showed the best clustered layout.



(a) C = 1, α = 0.5

(a) C = 1, α = 1

(a) C = 1, α = 1.5

(d) C = 1, α = 2

(e) C = 2, α = 2

(f) C = 3, α = 2

Figure 5.1.2.2. snapshots of clustered layout model. Network area = 1000m x 300m.
The number of node = 50, max. speed = 20m/s, pause time = 0 second.

5.2. Communication Model

We have chosen constant bit rate (CBR) as traffic sources and all traffic scenarios are generated by *cbrgen.tcl* program in NS2.We set the packet sending rate as 4 packets per second and data packet size is 512 bytes and control packet size is 48 bytes. We varied the maximum number of connections from 10 to 60. For each connection, we generated 5 different traffic scenarios with seed numbers – 1500, 2000, 2500, 3000 and 3500 – and calculated the average values of all simulation results.

5.3. Performance Metrics

5.3.1. Performance Metrics for Routing Efficiency

Usually, the performance of routing protocols is reported with the following metrics:

- the relative routing overhead, which is the ratio of the number of control packets over the number of delivered data packets,
- the delivery ratio, which is the number of packets delivered over the total number of packets sent,
- the end-to-end delay, which is average of delays between each pair of a data communication session, and
- the average hop distance, which the average number of hops taken by a delivered packet on its way to the destination.

We suggest three new metrics to measure routing efficiency based on hop counts.

- the average number of hops and optimal hops, which is calculated by NS2,
- the normalized hops, which is the ratio of the average hops over the optimal hops and
- the number of delivered packets versus average number of hops.

5.3.2. Performance Metrics for Energy Efficiency

In this section, first we discuss the energy efficiency metrics introduced in [16]. Then we present the details of applying energy efficiency metrics to evaluate energy efficiency in MANET routing protocols. We also discuss how to measure the energy consumption in NS2.

Michail et. al. in [16] presented the performance metric for energy efficiency that measures average number of accepted calls per simulation time. The energy efficiency is measured by maximizing the time until the first node turns OFF (looses its power). That is when rest of the nodes start turning OFF.

We evaluate energy efficiency using following performance metrics:

- the total energy consumption in three different workloads.
- throughput as well as network lifetime with limited energy amount in nodes.
- the energy usage per packet, which is the ratio of the total energy consumption over the number of delivered data packets, and
- the energy usage per hop, which is the ratio of the total energy consumed over the number of hops,

Network lifetime is defined as the time from beginning of simulation until first node in MANET runs out of energy.

We used energy model in NS2 to measure energy consumption of AODV, AODV-NC and AODV-NC with WLB. Even though the accuracy of energy model in NS2 has been pointed out in [8], we used this tool because we just focus on comparing efficiency of routing protocols in the same condition.

Energy model in NS2 has three states where energy is consumed: transmitting, receiving and idle state. Every node in NS2 starts with initial value which is the level of energy defined by user at the beginning of the simulation. It also has transmitting power (TXpower), receiving power (RXpower) and idle power parameters required by the node's physical layer. These values also can be defined by user. Initial energy level is decremented for transmission and reception of packets by TXpower and RXpower. When energy level in a node becomes zero, the node does not accept or send any packets.

We implemented the experiments under three different connections 20, 40 and 60. Speed 20 m/s was applied to all test cases and packet sending rate was 4 packets/sec. In an experiment that assumes unlimited amount of energy, each of the nodes start with energy 1000J which is enough to maintain whole 1800 sec simulation. We set the TX power to 0.6W, RX power to 0.3W. At this time, we did not consider idle state because we were just interested in energy consuming with transmitting and receiving packets.

We also tested the situation of the energy level in nodes that is not enough to remain alive until the end of the simulation. In this case, we set initial energy to 300J and set the power of idle state to 0.1W.

6. Experiment Study

Simulations have been run on Linux. All simulation results are averaged over five instances and provided with plots. We divide the experiments into the following sections.

- Routing efficiency measurement with AODV, AODV-NC (*H*), AODV-NC (*H*: *n* –*t*), AODV-WLB and AODV-NC (*H*) - WLB
    - cut time dependencies of AODV and AODV-NC (*H*),
    - cut time dependencies of AODV-NC (*H*), AODV-NC (*H*: *n* –*t*) and WLBs
    - various connections from 10 to 60
    - various speeds from 1m/s to 20 m/s with pause time 0 second.
    - new routing efficiency metrics
- Energy efficiency measurement with AODV, AODV- NC (*H*), AODV-NC (*H*: *n* –*t*), AODV-WLB and AODV-NC (*H*) – WLB
    - energy consumption
    - network throughput and network lifetime
    - energy consumption per a packet and hop
- Clustering impact measurement with AODV, AODV- NC (*H*), AODV-NC (*H*: *n* –*t*)
    - cut time dependencies of protocols with 1, 2 and 3 clusters
    - various connections from 10 to 50 with
    - Various speeds from 1m/s to 20 m/s with pause time 0 second.

6.1. Routing Efficiency

6.1.1. Cut time dependencies of AODV and AODV-NC (*H*)

Figure 6.1.1.1. shows that the cut time dependencies over different cut time of simulation. In this experiment, we have measured the relative overhead, delivery ratio, end-to-end delay and average hop numbers of AODV-NC with the different time threshold *H* at the different cut time of 1800 second simulation. We varied *H* values with 0.01, 0.1, 1 and 5 and fixed the maximum speed of node to 20 m/s, pause time to 0 second and the maximum numbers of connections to 20.

In Figure 6.1.1.1. the threshold 0.01 shows the lowest routing overhead. It means that the size of cache with 0.01 is highly constrained. However, threshold 0.01 does not show the best performance in the delivery ratio and end-to-end delay in Figure 6.1.1.2. and 6.1.1.3.. Threshold 0.1 and 1 show better delivery ratio and delay than 0.01 and 5 even though they have relatively big number of hops shown in Figure 6.1.1.4.. As a result, highly constrained node caching fails to find the shortest route and do not guarantee relatively safe routes. Therefore, we chose the threshold value H to 0.1 and 1 which show the best performance in delivery ratio and delay for the future experiments.

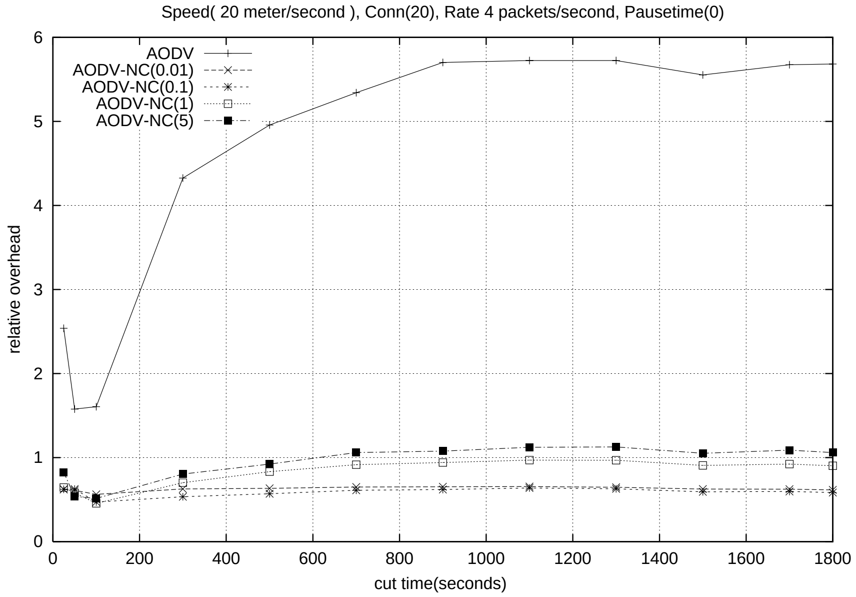


Figure 6.1.1.1. The relative routing overhead at the different cut times of simulation for AODV and AODV-NC with threshold 0.01, 0.1, 1, 5

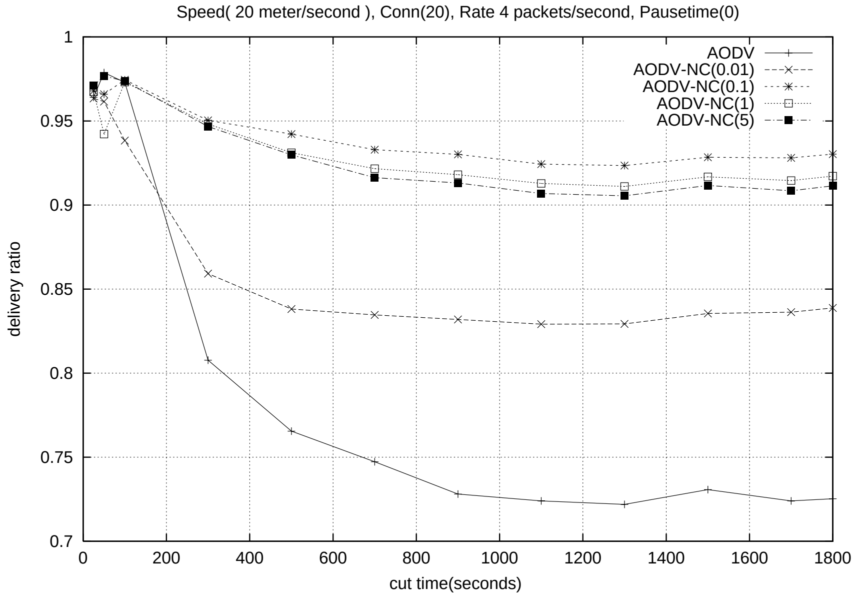


Figure 6.1.1.2. The packet delivery ratio at the different cut times of simulation for AODV and AODV-NC with threshold 0.01, 0.1, 1, 5

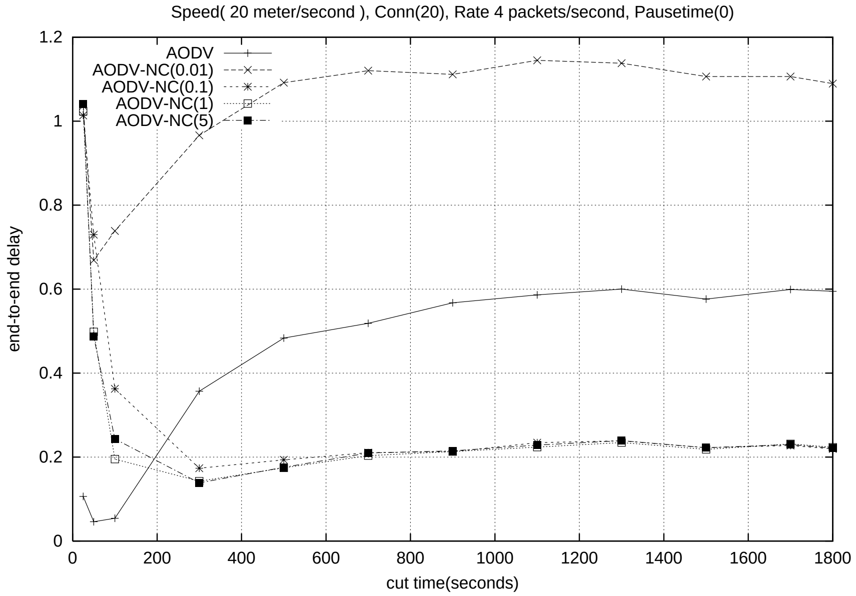


Figure 6.1.1.3. The end-to-end delay at the different cut times of simulation for AODV and AODV-NC with threshold 0.01, 0.1, 1, 5.

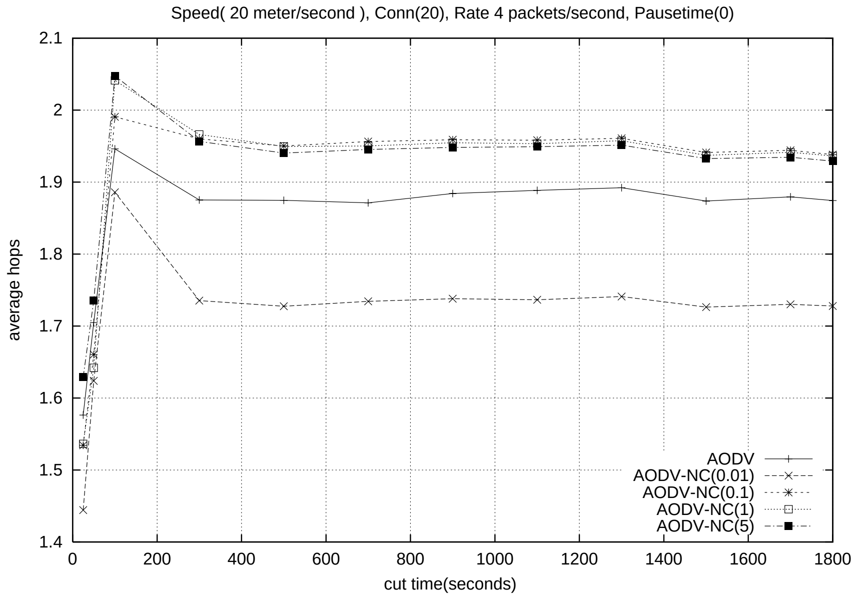


Figure 6.1.1.4. The average number of hop at the different cut times of simulation for AODV and AODV-NC with threshold 0.01, 0.1, 1, 5.

6.1.2. Cut time dependencies of AODV-NC and AODV-NC (*H*) with Load balancing

Figure 6.1.2.1 ,6.1.2.2., 6.1.2.3., and 6.1.2.4. show the performance results of AODV and AODV-NC applied load balancing scheme over different cut times . We applied the same parameters for node movement and communication scenario used in section 6.1.1.. In Figure 6.1.2.1., adaptive WLB reduced the routing overhead of AODV approximately 17% after 800 seconds. However, WLB with AODV-NC protocol show similar results AODV-NC itself. For the delivery ratio in Figure 6.1.2.2., WLB improves AODV slightly and AODV-NC(0.1) and AODV-NC(0.1)-WLB show the similar results too. Since AODV-NC(0.1) and AODV-NC (0.1)-WLB shows the similar performance overall simulation, we can guess WLB fails to improve routing efficiency when combine with AODV-NC. However, in other situation such as heavy traffic scenario, it improves the energy efficiency. It will be discussed later.

Our simple load balancing scheme reduces the performance of AODV-NC in the delivery ratio and delay. However, as seen in section 4.1., it solves the unfairness of AODV-NC and over performs AODV in almost metrics.

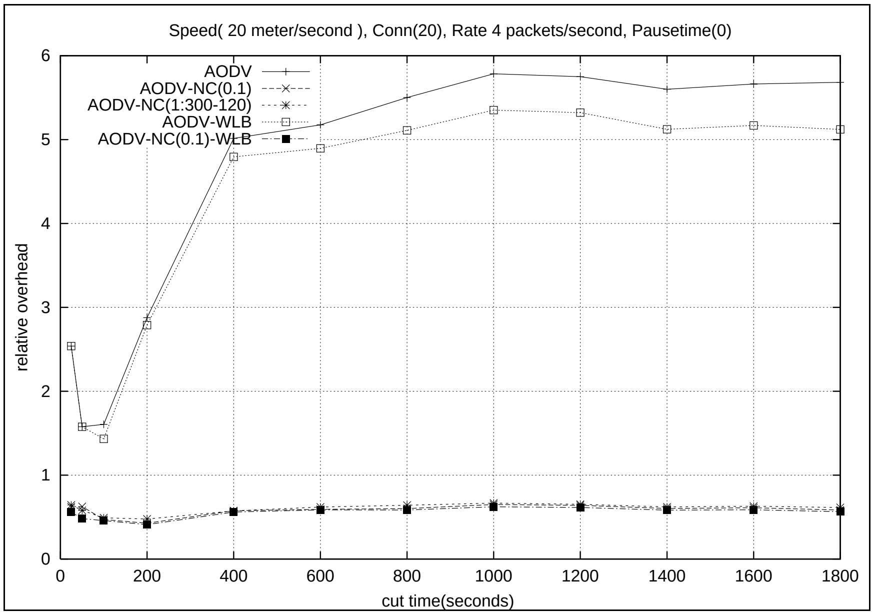


Figure 6.1.2.1. The relative routing overhead at the different cut times of simulation for AODV and AODV-NC with load balancing

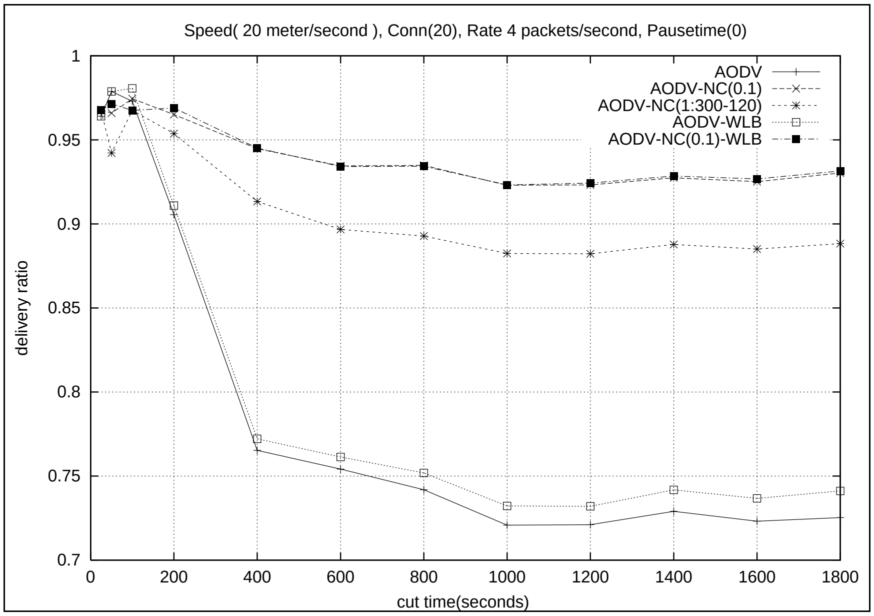


Figure 6.1.2.2. The Delivery ratio at the different cut times of simulation for AODV and AODV-NC with load balancing

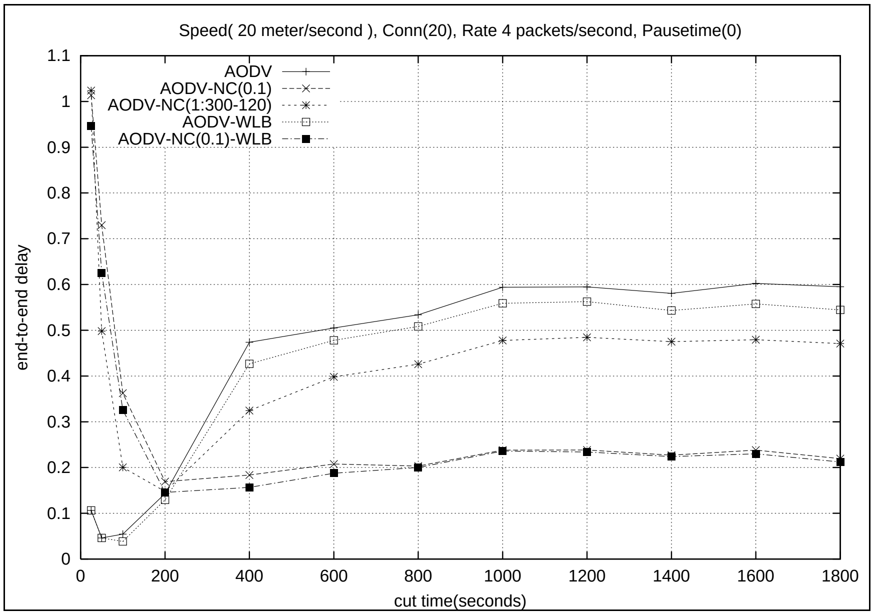


Figure 6.1.2.3. The end-to-end delay at the different cut times of simulation for AODV and AODV-NC with load balancing

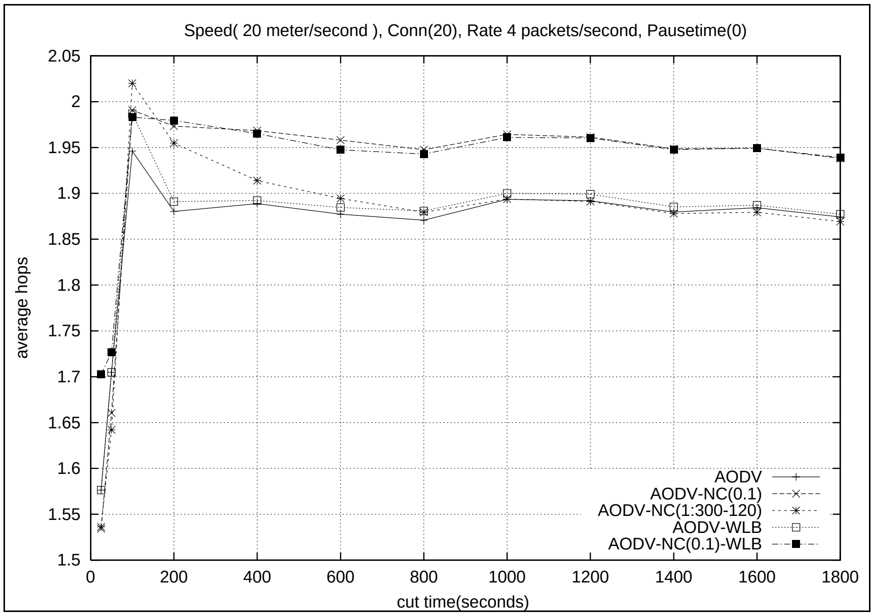


Figure 6.1.2.4. The average number of hops at the different cut times of simulation for AODV and AODV-NC with load balancing

6.1.3. Various connections from 10 to 60

Figure 6.1.3.1. explores behavior of the protocols when the number of connections is growing from 10 to 60. We fix the maximum speed to 20 m/s and pause time to 0 second. Unfair AODV-NC protocols (respectively, the fair AODV-NC(1:300-120) protocol) increase delivery ratio by 20% (respectively, 15%) and reduce overhead by factor 7 (respectively, 3). The delay reduction for the unfair protocols AODV-NC(0.1) and AODV-NC(1) is 40% while the fair protocol AODV-NC(1:300-120) has mere 25% delay reduction. In case of 30 connections, AODV-NC(0.1)-WLB improves delivery ratio and deceases relative overhead. Also, AODV-WLB shows better performance than AODV itself at high workload. As shown in [], WLB is efficient at the high workload condition. In case of 40 and 50 connections, AODV-WLB improves delivery ratio, relative overhead and end-to-end delay up to 6%, 23% and 7.5% respectively. With AODV-NC, WLB improves delivery ratio, relative overhead and end-to-end delay up to 32%, 85% and 41% respectively. These results are better than AODV-NC itself. At implies that workload-based load balancing technique shows better performance when working with AODV-NC instead of working with AODV alone especially at the high workload environment.

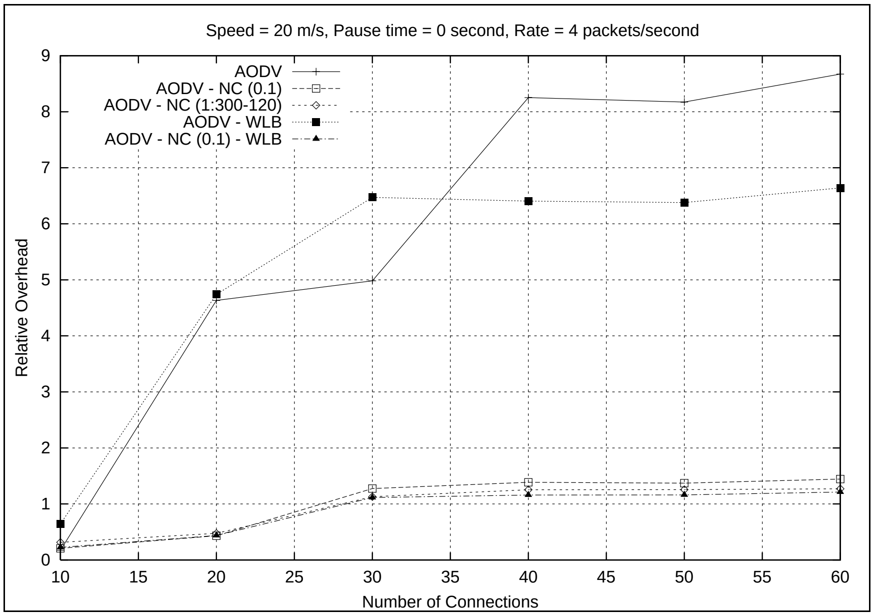


Figure 6.1.3.1. The relative routing overhead for the different number of connections with AODV and AODV-NC with load balancing

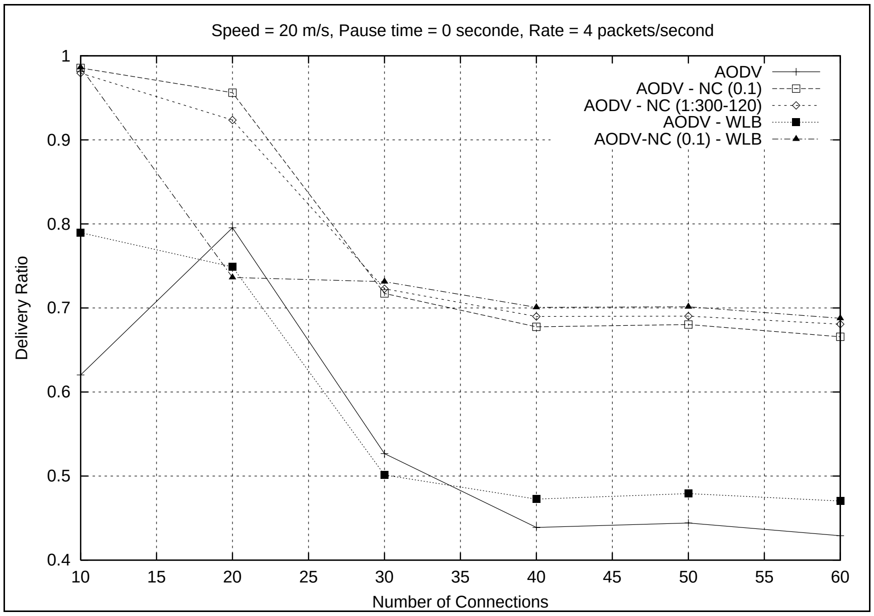


Figure 6.1.3.2. The delivery ratio for the different number of connections with AODV and AODV-NC with load balancing

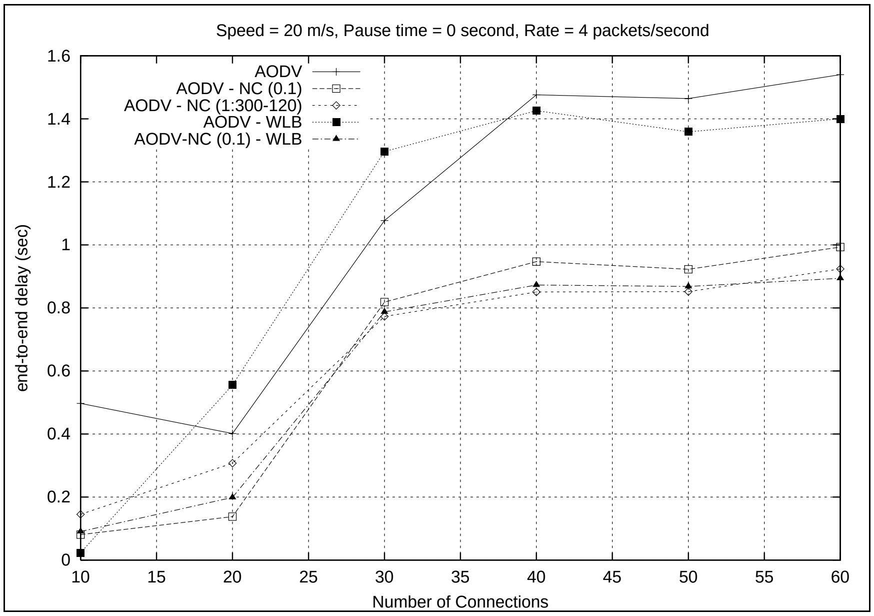


Figure 6.1.3.3. The end-to-end delay for the different number of connections with AODV and AODV-NC with load balancing

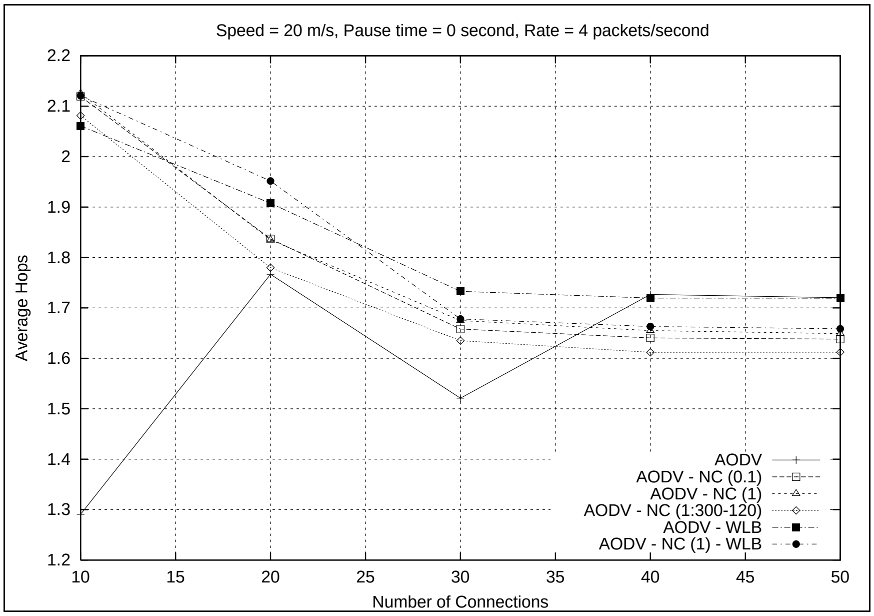


Figure 6.1.3.4. The average number of hops for the different number of connections with AODV and AODV-NC with load balancing

6.1.4. Various speeds from 1m/s to 20 m/s with pause time 0 second.

Figure 6.1.4.1., 6.1.4.2., 6.1.4.3., and 6.1.4.4. explore behavior of the protocols when the speed is growing from 1m/s to 20 m/s. All three proposed node caching protocols increase delivery ratio by 10-20% and reduce overhead by factor 10. The delay reduction for unfair protocols AODV-NC(0.1) and AODV-NC(1) is 3-4 times while the fair protocol AODV-NC(1:300-120) has mere 10-20% delay reduction. When there is no path to the destination through cached nodes then delay and overhead are obviously larger than that of original AODV. But due to the high hit ratio AODV-NC frequently uses cached nodes which significantly reduce overhead and delay. Also, due to lower overhead the message queue is not swamped with RREQs and AODV-NC is less likely to drop the messages that increase the packet delivery ratio resulting in the delivery ratio increase. At speed 1 m/s, AODV-WLB increases delivery ratio by 12% but results in increase in relative overhead and delay. However, AODV-NC(0.1)-WLB increases delivery ratio with decrease of relative overhead and delay. At other speeds such as 5 m/s, 10 m/s and 20 m/s, AODV-NC(0.1) shows better performance than AODV-NC(0.1)-WLB. It means that in high mobility scenarios, routing with AODV-NC combined with workload-based load balancing failed to find a path at the first attempt. It causes a node to send route request packets again. As a result, routing overhead as well delay increases.

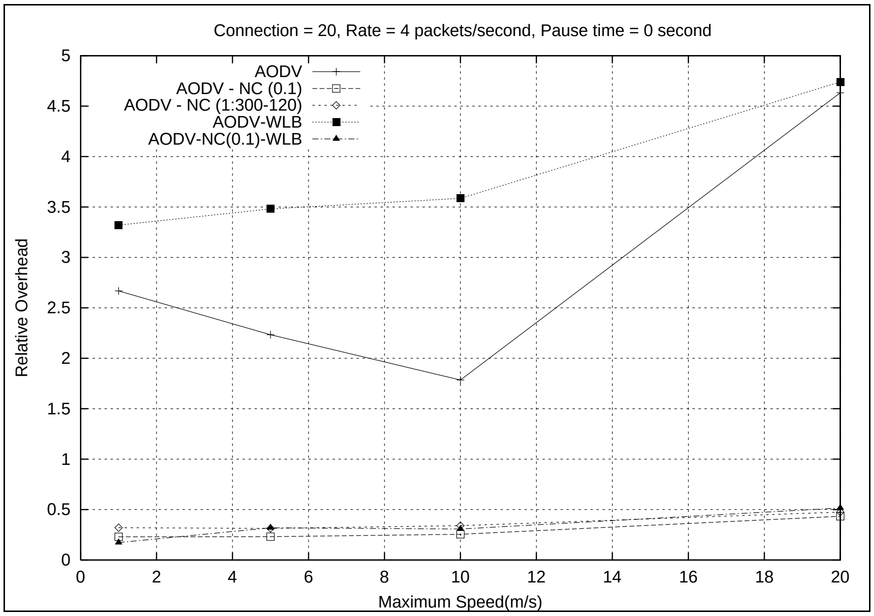


Figure 6.1.4.1. The relative routing overhead for the different maximum speeds of nodes with AODV and AODV-NC with load balancing

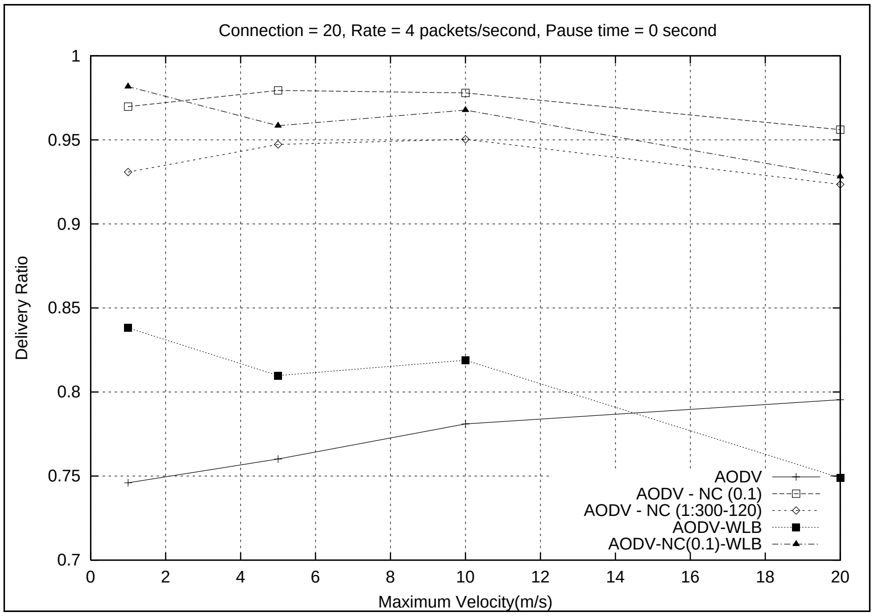


Figure 6.1.4.2. The delivery ratio for the different maximum speeds of nodes with AODV and AODV-NC with load balancing

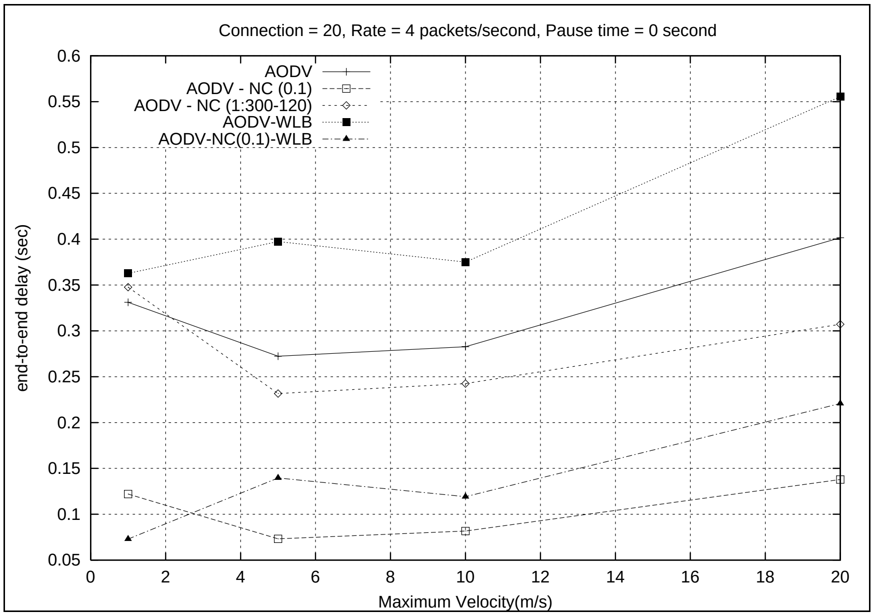


Figure 6.1.4.3. The end-to-end delay for the different maximum speeds of nodes with AODV and AODV-NC with load balancing

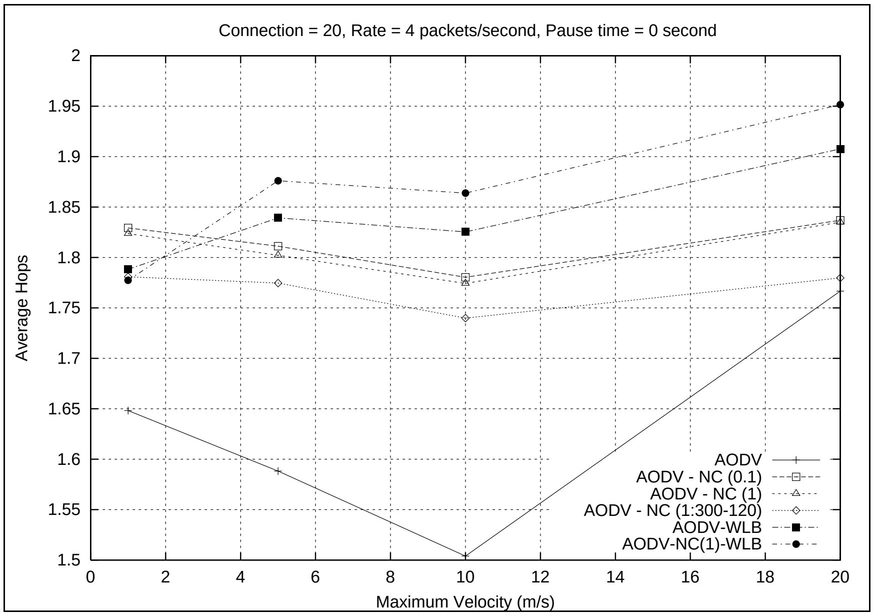


Figure 6.1.4.4. The average number of hops for the different maximum speeds of nodes with AODV and AODV-NC with load balancing

6.1.5. New routing efficiency metrics

We also compare the routing efficiency with the average number of hops and optimal hops. Optimal hops are calculated by NS2 during the simulation. In Figure 6.1.5.1., the first plot shows that the average number of hops and optimal hops depending on the cut off time. AODV-NC(1:300-120) delivers packets with the smallest number of hops while AODV uses the largest number of hops. On the other hand, AODV uses the lowest optimal hops among the rest of the protocols. The second plot shows the ratio of average hops to optimal hops in that AODV and AODV-WLB show higher ratio than other protocols while AODV-NC (0.1) shows the lowest ratio. It means that AODV-NC protocols find a shorter path than AODV. The last plot shows the distribution of delivered data packets per hops. In this plot, AODV-NC tends to send packets with smaller number of hops than AODV. AODV uses larger number of hops compared to other protocols. Between 6 and 9 hops, one can see the solid line (AODV) which is above dotted lines (AODV-NC protocols).

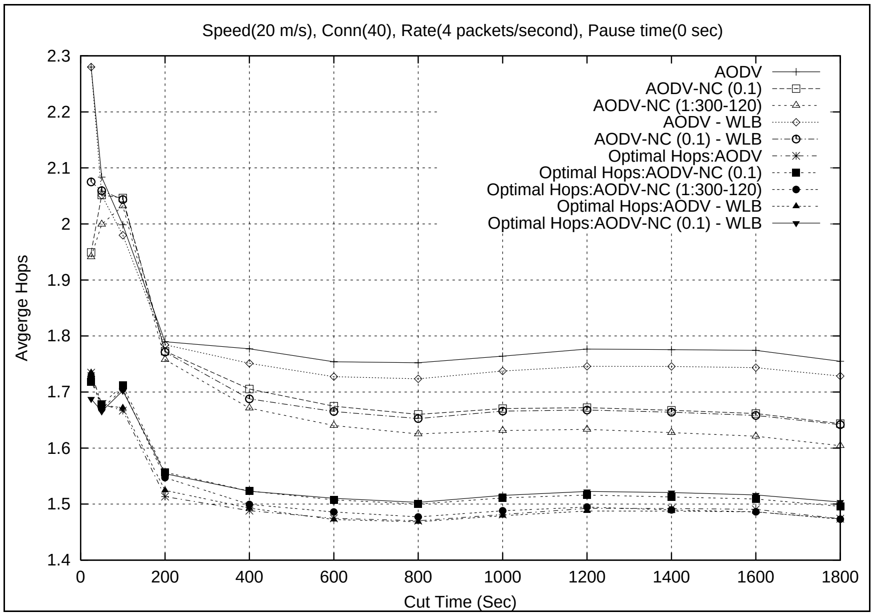


Figure 6.1.5.1. the average number of hops and the average number of optimal hops at different cut times with 40 connections

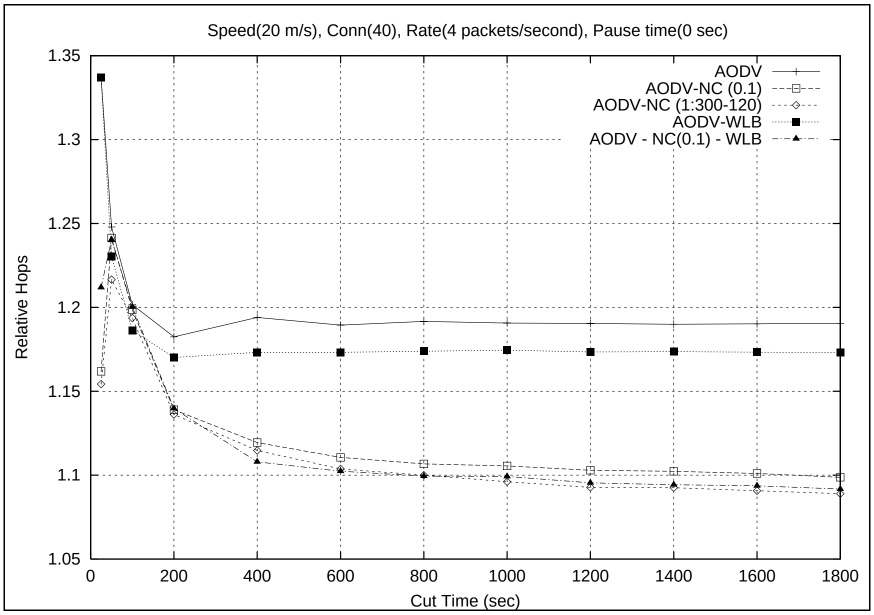


Figure 6.1.5.2. normalized hops at different cut times with 40 connections

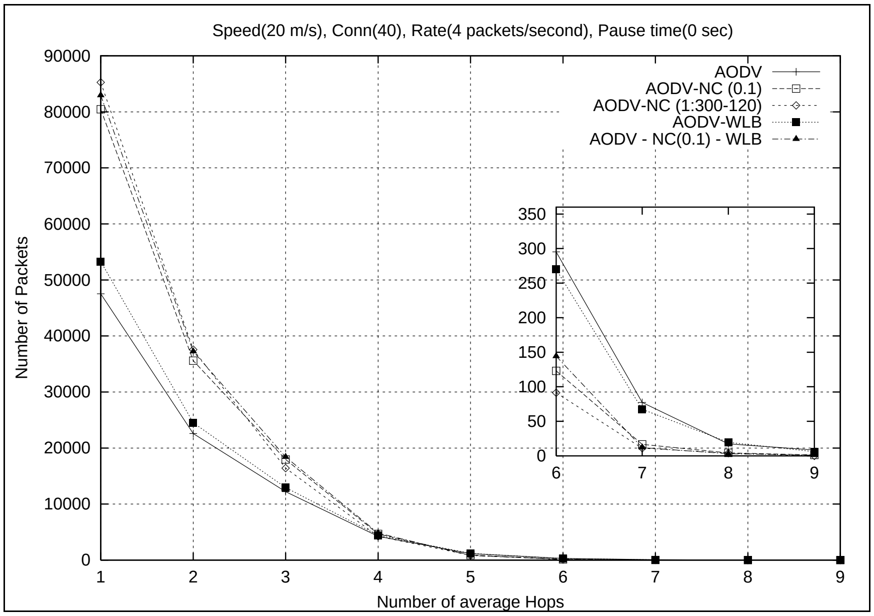


Figure 6.1.5.3. the distribution of average number of hops with 40 connections

6.2. Energy Efficiency

6.2.1. Energy consumption

Figure 6.2.1.1. shows energy usages of five routing protocols. AODV-NC (1:300-120) uses the least energy for a scenario with 20 connections while AODV-NC (0.1)-WLB uses the lowest energy usage for a scenario with 40 connections. This performance can be explained with relative overhead shown in Figure 6.1.3.1.. AODV-NC (1:300-120) and AODV-NC(0.1)-WLB show relatively lower routing overhead for connections 20 and 40 respectively. Since, these two protocols reduce routing request packets, it results in saving energy. AODV-NC (0.1)-WLB works better for high workload and requires less number of control packets therefore it is energy efficient in high workload scenario.

However, the differences among AODV and enhanced AODV protocols in regard to energy consumption are very small. In other words, energy consumption by itself is not suitable metric to compare energy efficiency. So, we measure the packet delivery throughput of protocols in the networks with limited energy nodes. For same energy consumption there is a significant difference in amount of packet delivery throughput for different protocols. In addition, we measure network lifetime using packet delivery throughput and simulation time.

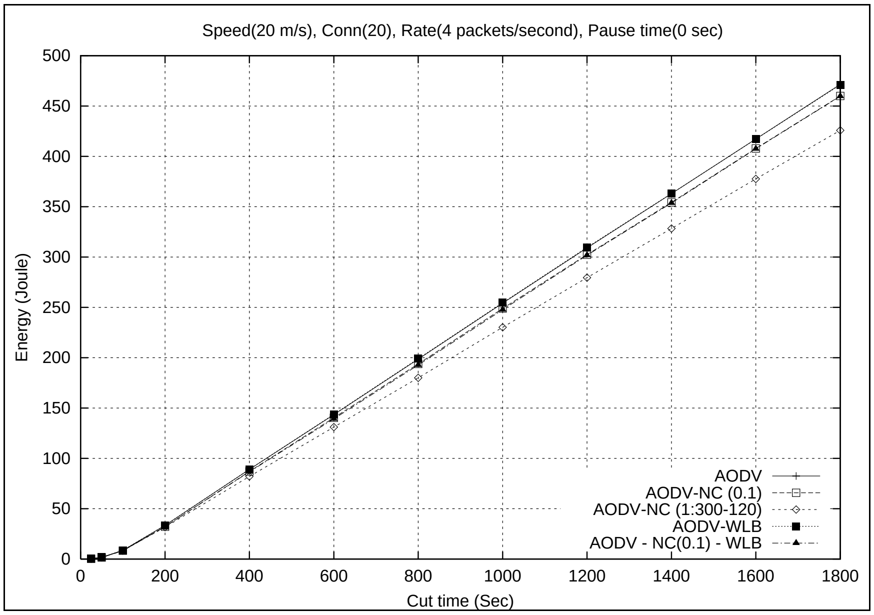


Figure 6.2.1.1. Energy consumption at different cut times with 20 connections.

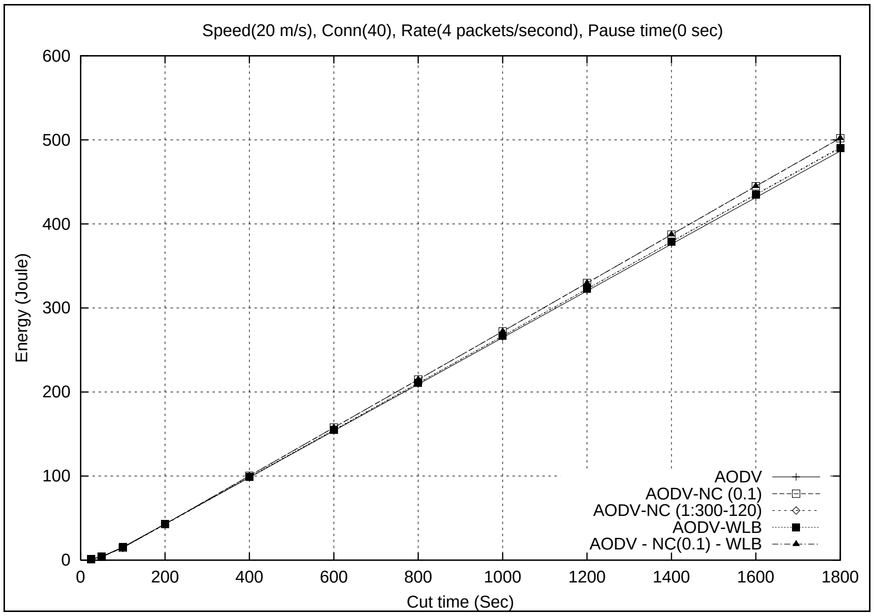


Figure 6.2.1.2. Energy consumption at different cut times with 40 connections.

6.2.2. Network throughput and lifetime

In Figure 6.2.1.1., we observe that the total energy consumption of all the protocols is almost the same however; the throughput of AODV-NC and AODV-NC-WLB is higher than the original AODV. The throughput measurement that is the metric of number of delivered packets per cut off time explains the efficiency of the protocol clearly.

In Fig 6.2.2.1., AODV-NC(0.1)-WLB records the highest throughput in every scenario. AODV-NC(0.1)-WLB delivers almost 30% more packets than AODV. It is obvious because AODV-NC(0.1)-WLB delivers 30% more packets than other protocols, as seen in Figure 6.1.3.1.. AODV shows the lowest throughput than the rest. Also, in this figure, we can see that the AODV-NC(0.1) is the first protocol in which a node looses all its power followed by AODV. It means that the lifetime of AODV-NC(0.1) is the shortest among five protocols. Because of overused nodes in AODV-NC, it results in short network lifetime. On the other hand, AODV-NC(1:300-120) keeps the simulation running approximately 1180 sec without having any node run out of power supply. It means that load balancing techniques in AODV-WLB, AODV-NC (1:300-120) and AODV-NC (0.1)-WLB extend the network lifetime as well as network throughput.

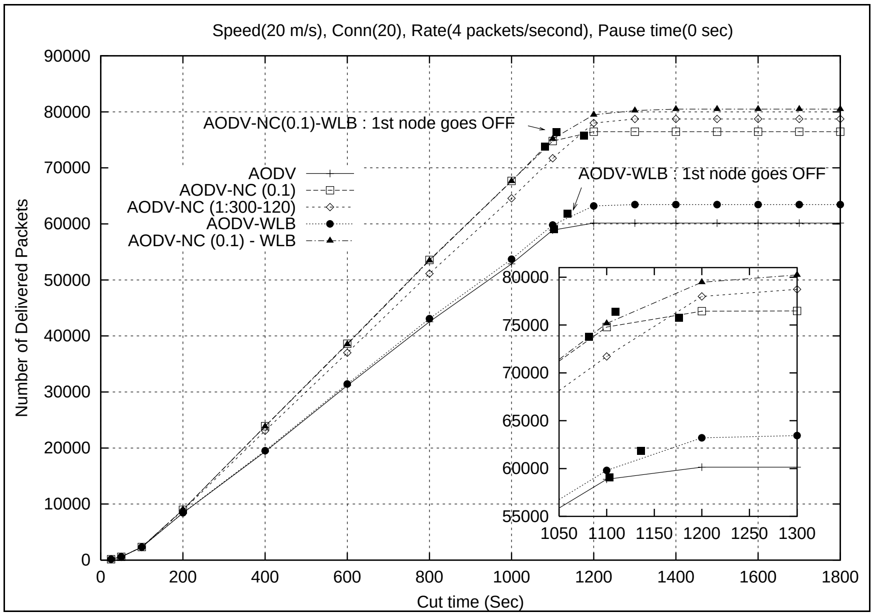


Figure 6.2.1.3. Network throughput and lifetime at different cut times with 20 connections

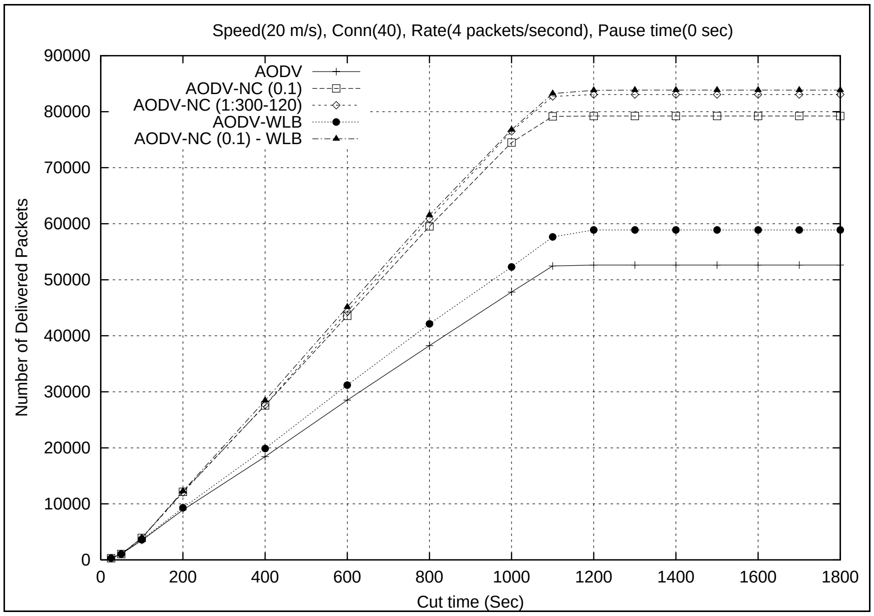


Figure 6.2.1.4. Network throughput at different cut times with 40 connections

6.2.3. Energy consumption of a packet and hop

Figure 6.2.3.1. and 6.2.3.2 shows the energy consumption per delivered data packet and the energy consumption per hop in the scenario of 40 connections which is relatively higher workload scenario in our simulation. AODV-NC protocols use less energy to deliver a data packet than AODV. Especially, AODV-NC (1:130-200) improves almost 35% in energy saving. Also, for the energy consumption per hop, AODV-NC (0.1)-WLB shows the least energy consumption. From two plots, we can derive that Load balancing technique is effective to save energy in AODV and AODV-NC.

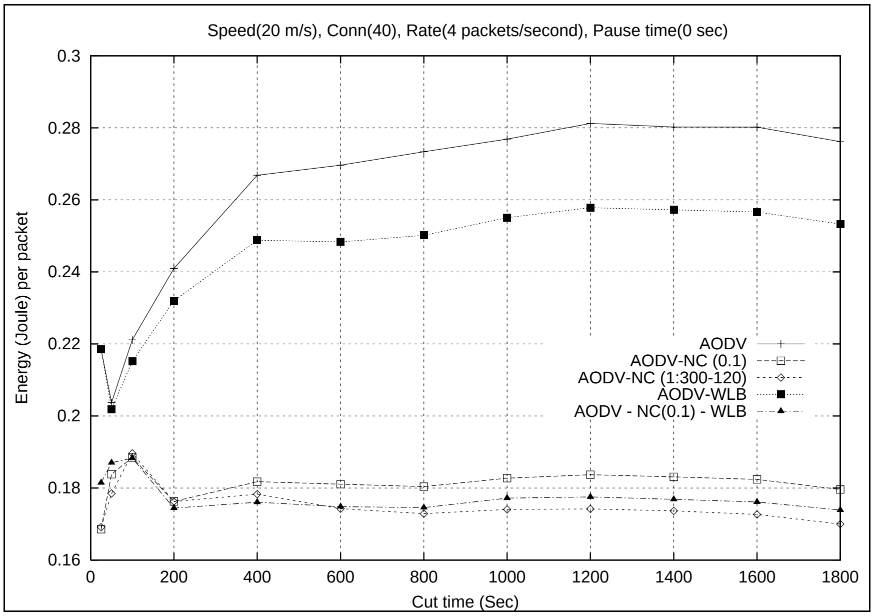


Figure 6.2.1.5. Energy consumption per a delivered packet at different cut times with 20 connections

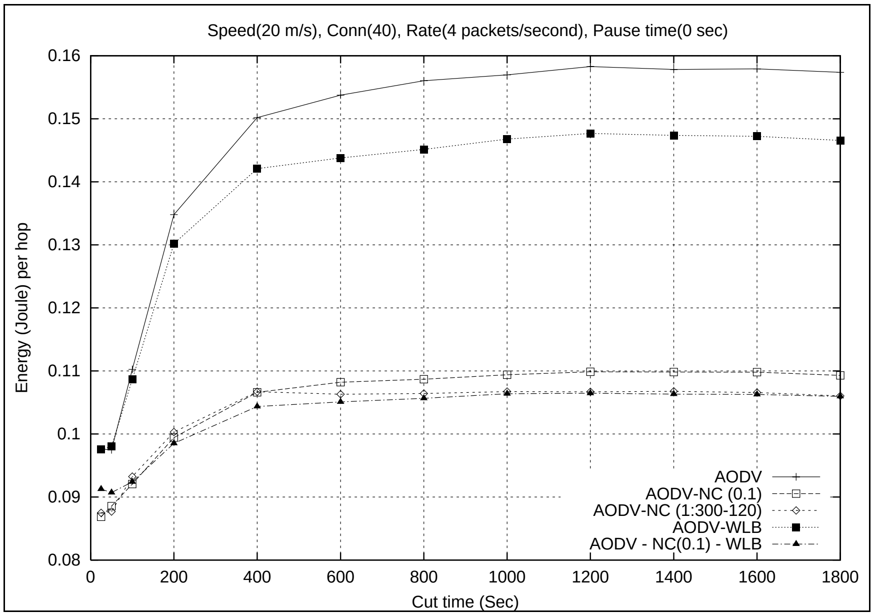


Figure 6.2.1.6. Energy consumption per a hop at different cut times with 20 connections

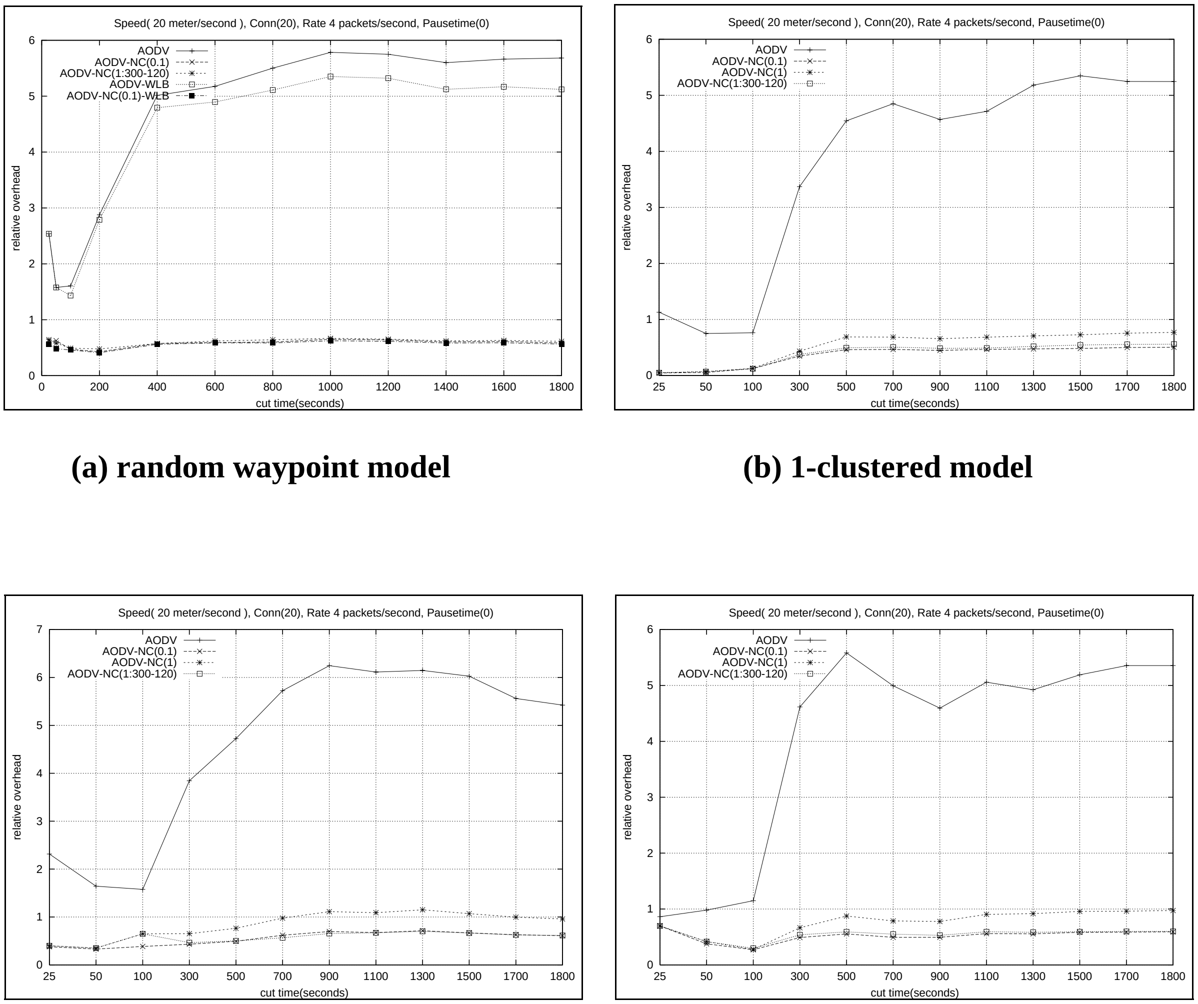
6.3. Clustered Movement

In this section, we observe the clustering impact of performance in routing protocols. We generated 3 kinds of cluster layout and movement scenarios using our generator shown in section 5.3.. For each of clusters, the node locations at the beginning are shown in Figure 5.3.1..

6.3.1. Cut time dependencies for different number of clusters.

First, we compare the performance of routing protocols with random movement model, 1, 2, and 3 clustered movement models over the different cut times for 1800 seconds. For convenience, we redisplay the same figures of cut time dependencies shown in previous pages again with small size. We set the maximum speed to 20 m/s with pause time 0 seconds and 20 connections for all scenarios.

In Figure 6.3.1.1., the relative routing overhead for the random, 1-clustered, 2-clustered, 3-clustered model are compared in a row. While 2-clustered layout model shows the highest routing overhead, 1 and 3 clustered models are similar with random model. Also, the delivery ratio of 2-clustered model in Figure 6.3.1.2. is also worst among them. And, an interesting fact is that 1 and 3-clustered model show the better delivery ratio than random model. In addition to delivery ratio, end-to-end delay and the number of hops are decreased in Figure 6.3.1.3. and 6.3.1.4.. this results are explained shortly. Because nodes are very close to each other in clusters, they can communicate fast. However, in 2-clustered layout, two clusters are partitioned into two locations, the nodes in the middle may struggle with heavy overload.



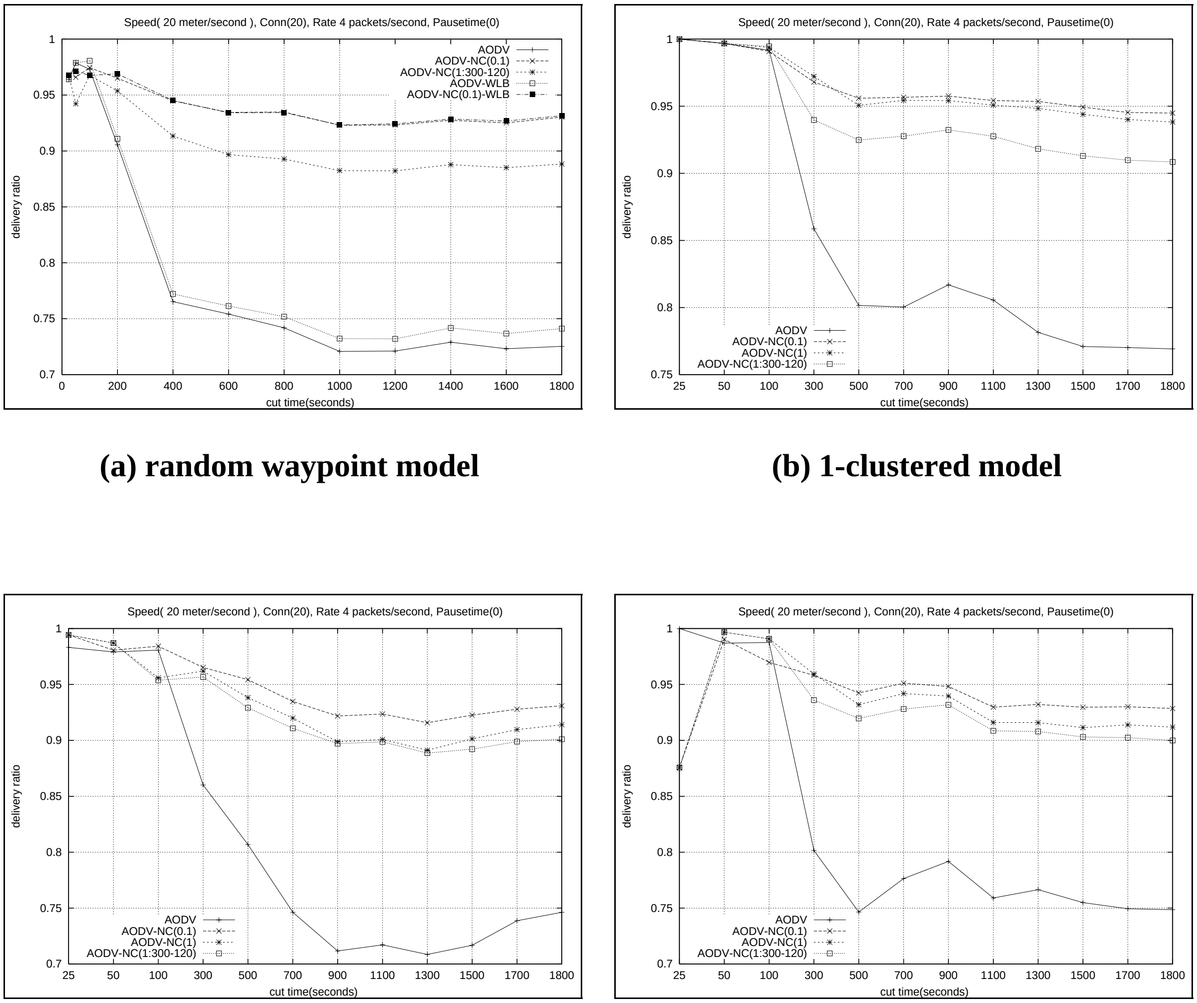
(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.1.1. The relative overhead for different movement models.



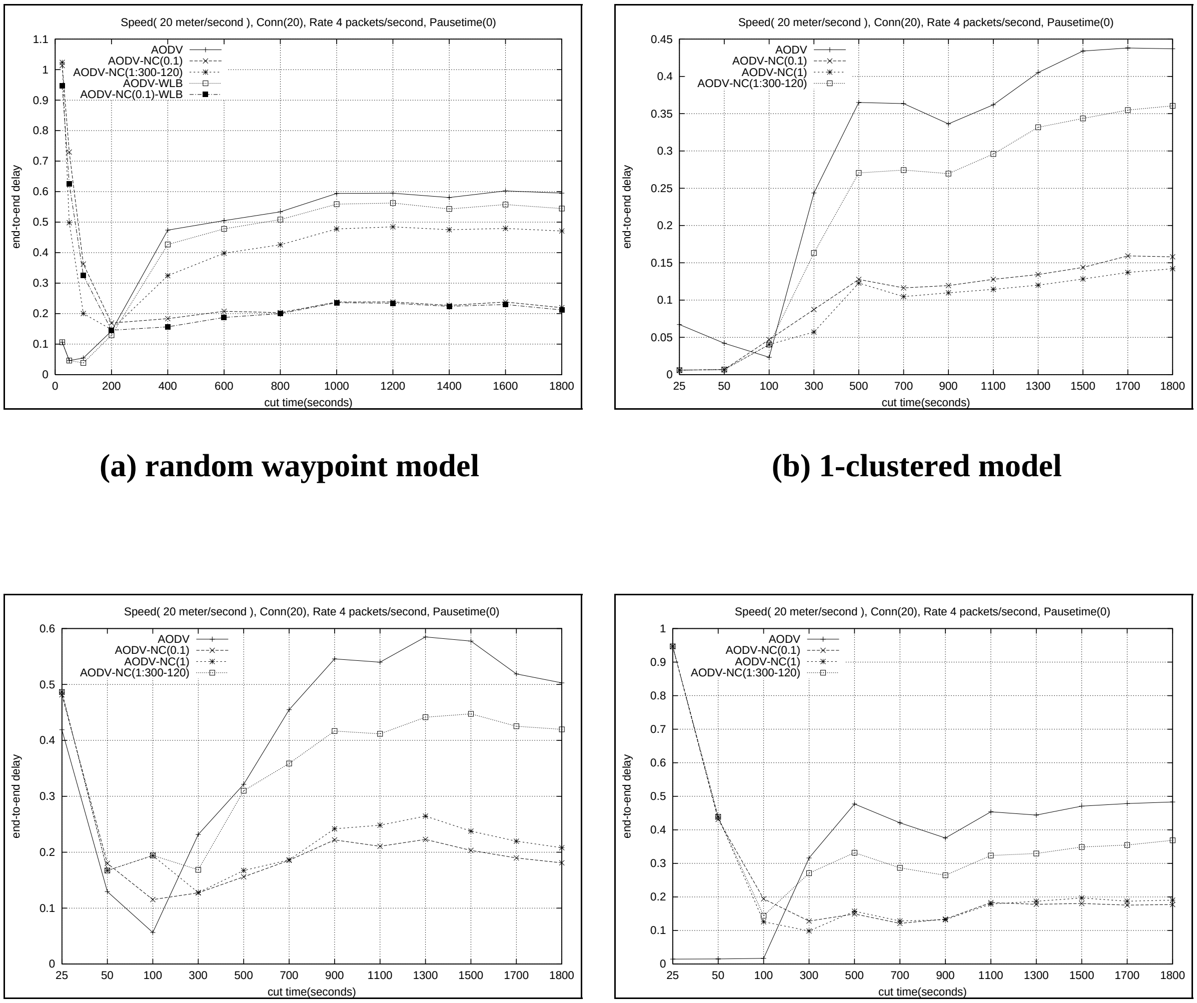
(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.1.2. The delivery ratio for different movement models.



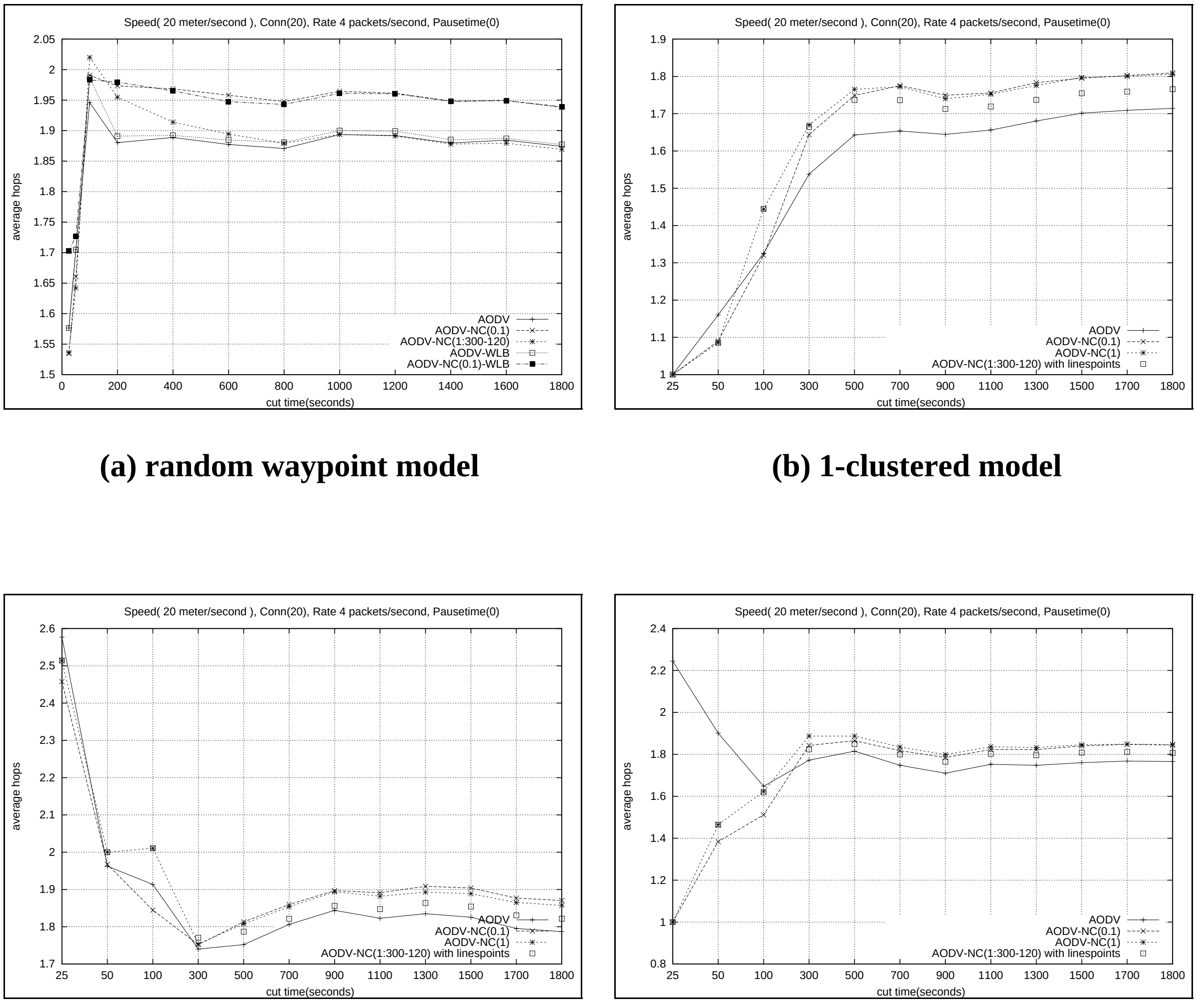
(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.1.2. The End-to-end delay for different movement models.



(a) random waypoint model

(b) 1-clustered model

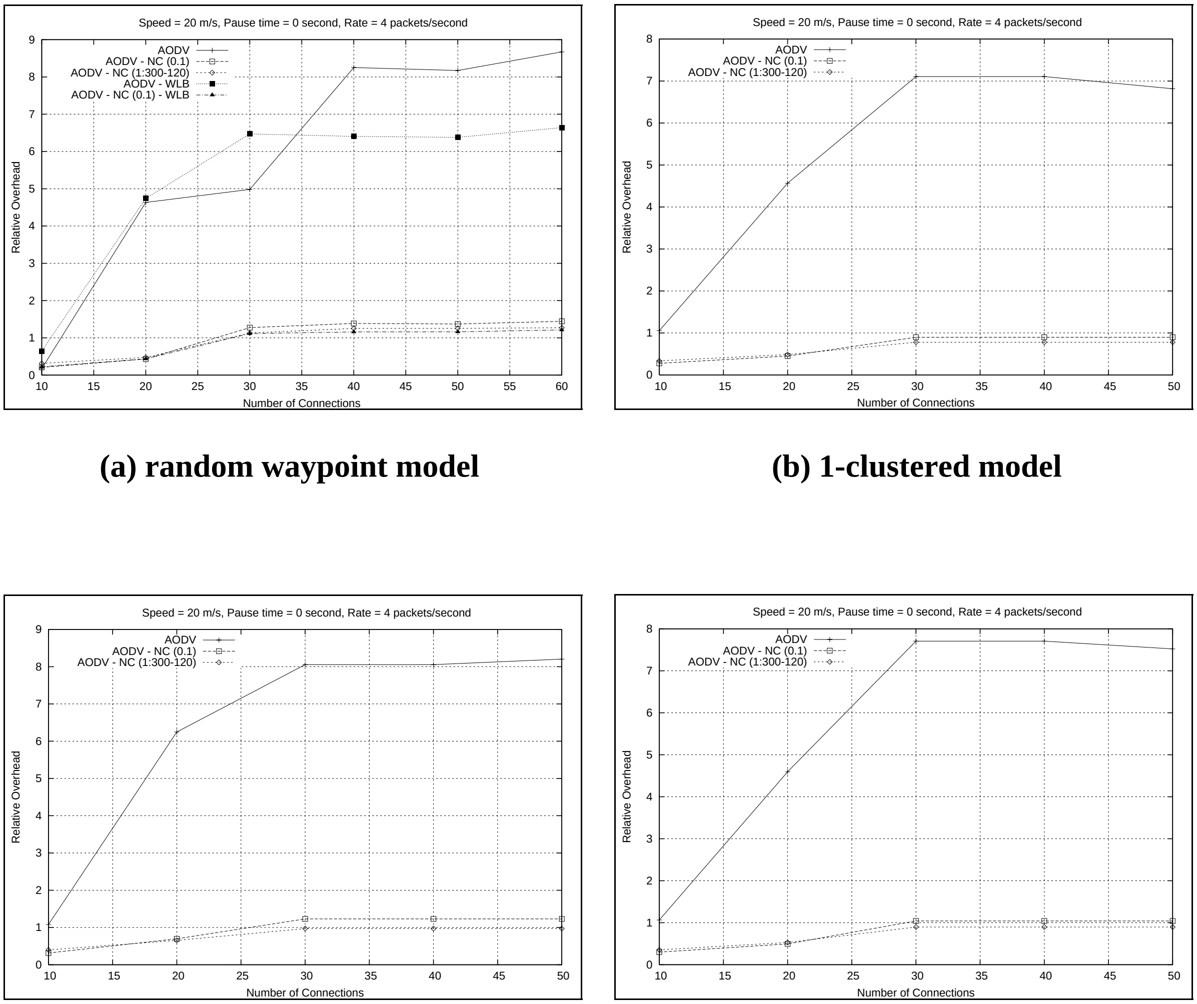
(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.1.2. The average number of hops for different movement models.

6.3.2. Various connections from 10 to 50

In this experiment, we consider the performance impact of node clustering for various communication overloads. We set the maximum speed of nodes to 20 m/s with pause time 0 and fixed the data packet sending rate as 4 packets/second. In Figure 6.3.2.1., random waypoint model shows the highest routing overhead among four models. Especially, the overload of AODV gets severe as connections increases. Also, AODV-NCs show the highest routing overhead in random model. However, the difference is very subtle. Moreover, AODV-NCs maintain the low overhead ratio compared with AODV for all connections. In clustered models, the relative routing overhead increases linearly to the 30 connections, but it is stabilized in connection 40 and 50. As seen in cut time dependencies in section 6.3.1, 2-clustered layout records the highest routing overhead in clustered models. In the delivery ratio, random model shows the lowest delivery ratio. 2-clustered layout also shows lower delivery ratio than other clustered layout. In Figure 6.3.2.3., delay shows the same results. However, the average hops in Figure 6.3.2.4., shows different results. 2-clustered layout finds the longest route to the destination than random waypoint model. The reason can be a partition between two clusters.



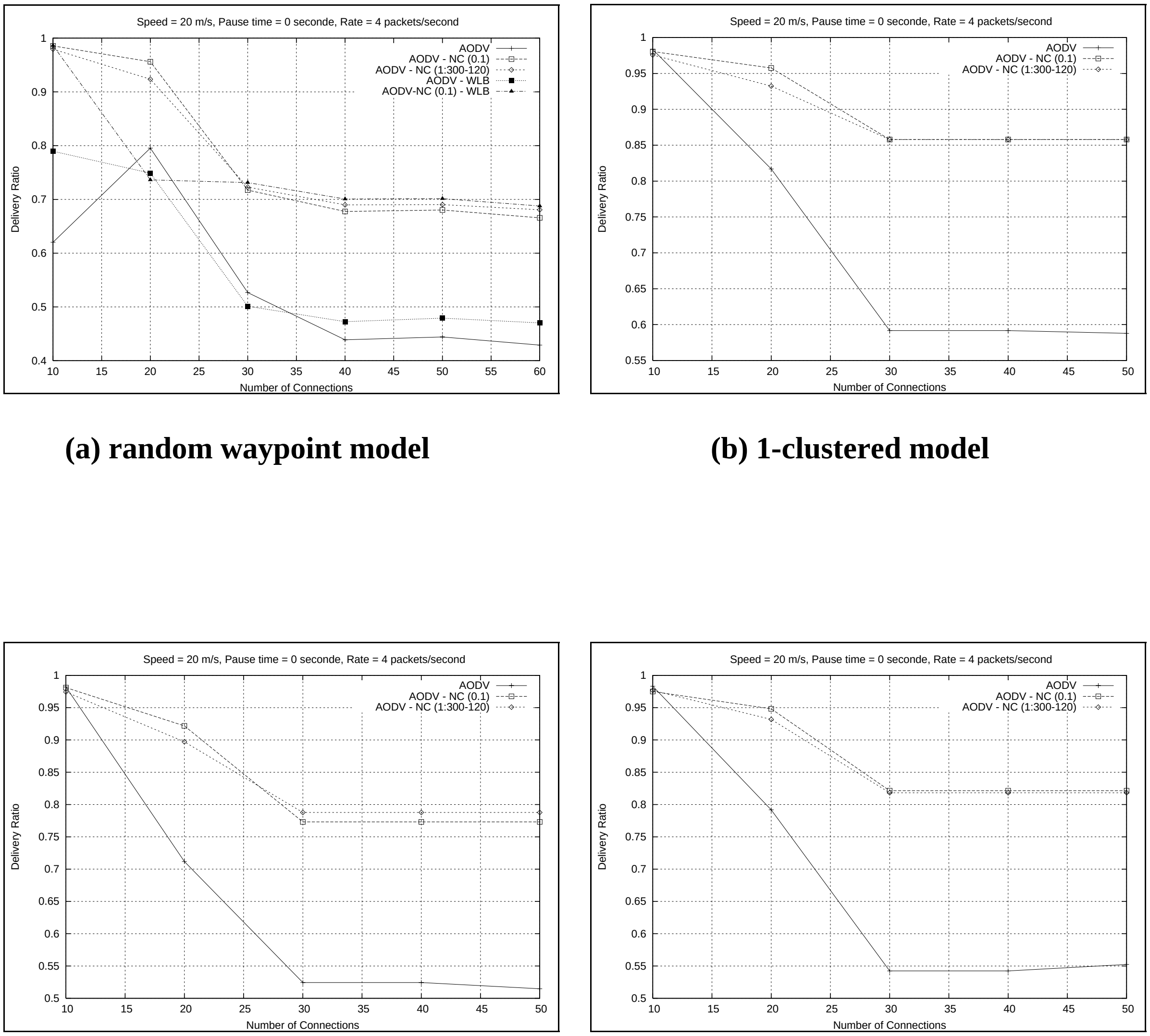
(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.2.1. The relative routing overhead over various numbers of connections for different movement models.



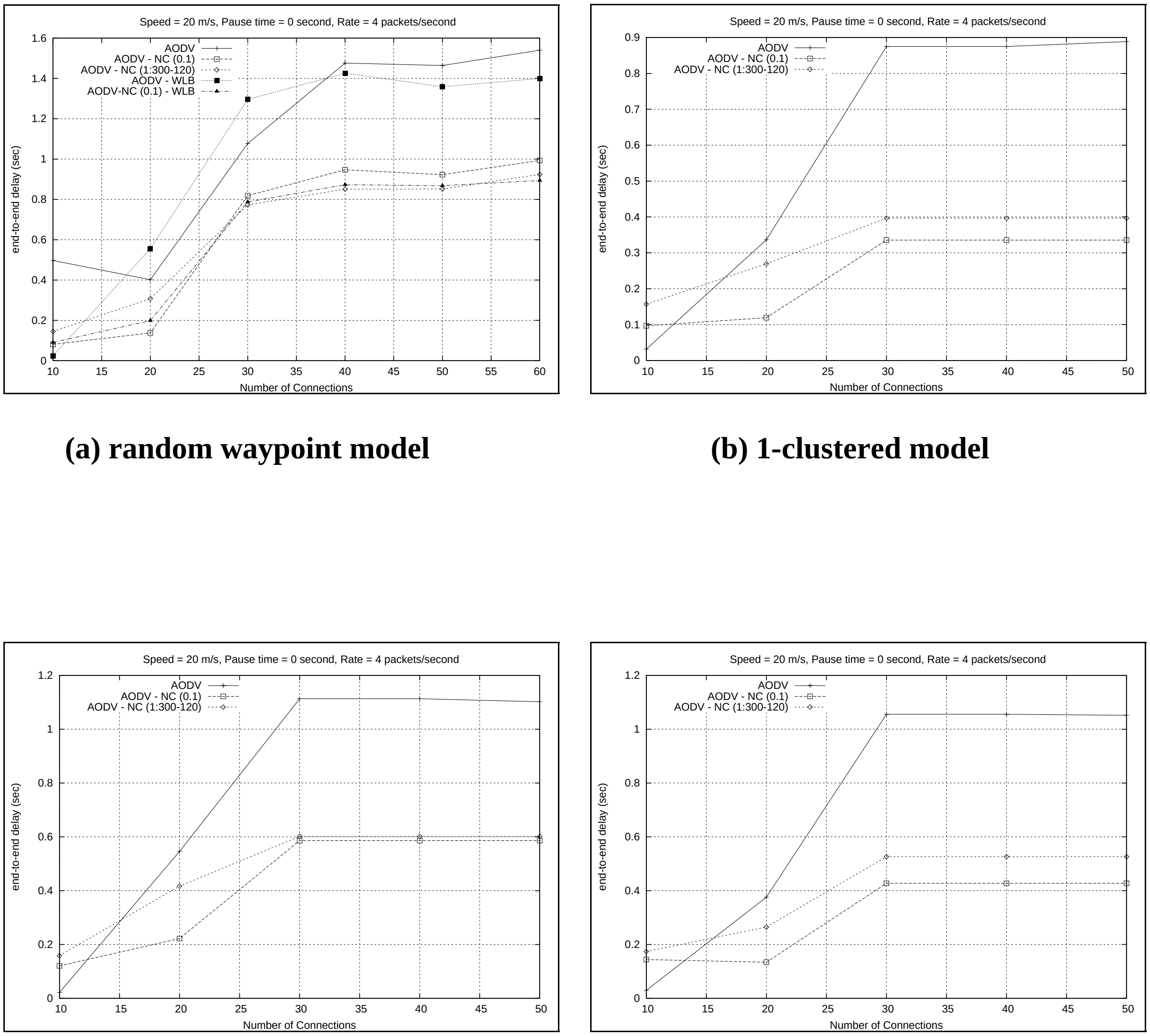
(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.2.2. The delivery ratio over various numbers of connections for different movement models.



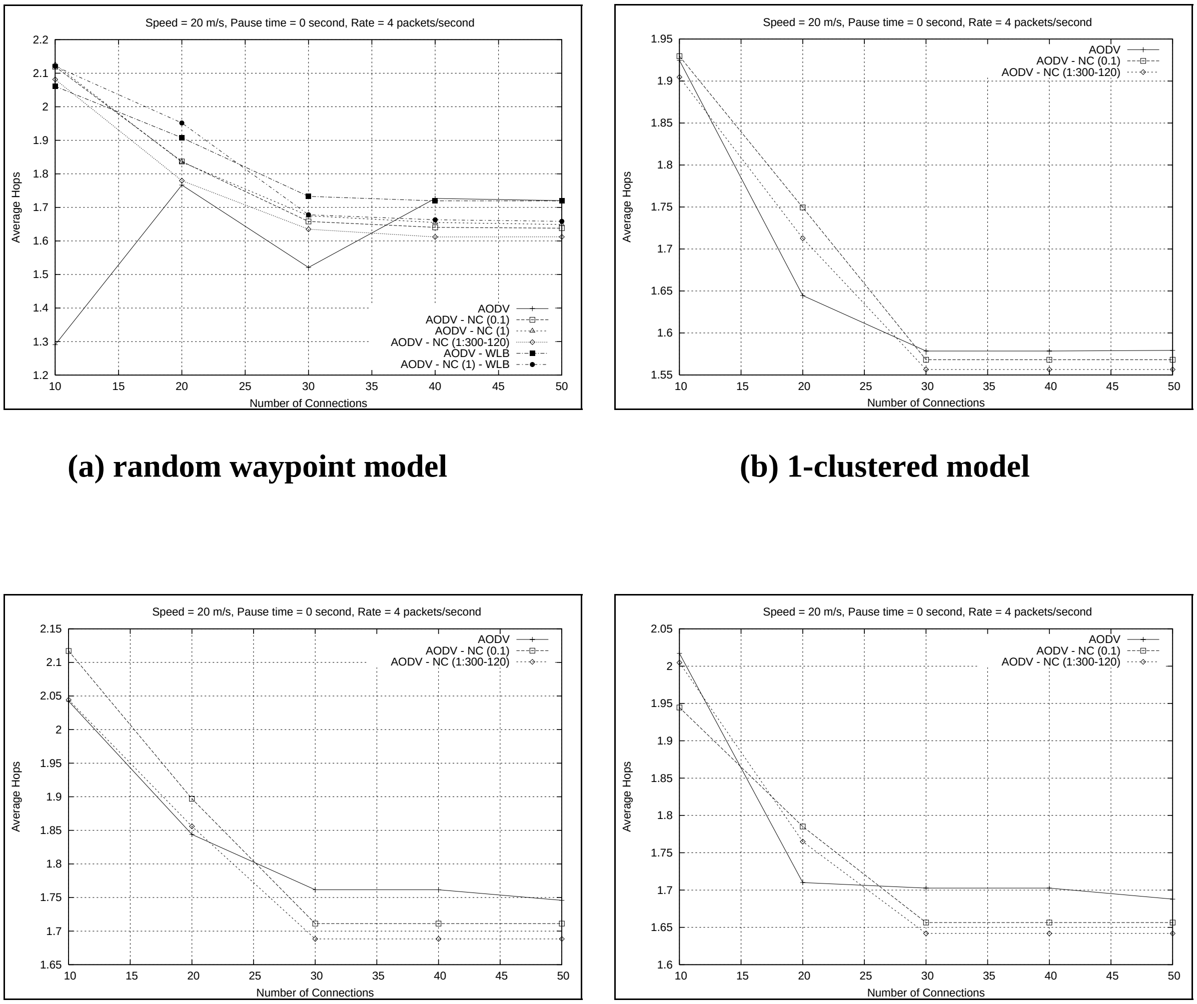
(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.2.3. The end-to-end delay over various numbers of connections for different movement models.



(a) random waypoint model

(b) 1-clustered model

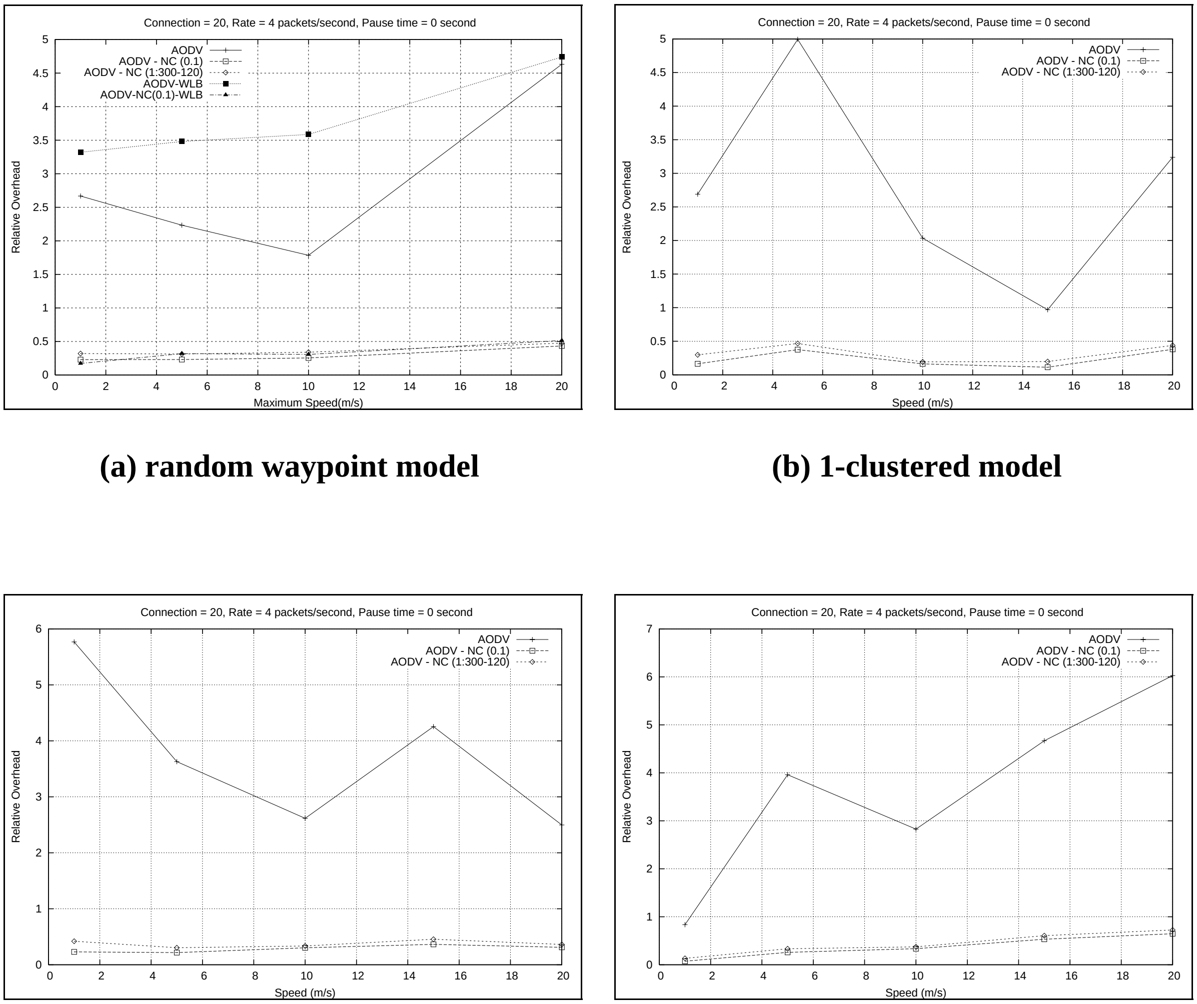
(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.2.4. The average number of hops over various numbers of connections for different movement models.

6.3.3. Various speeds of nodes

We fixed the number of connections to 20 and transmitted 4 packets per second. We varied the maximum speed of nodes from 1 m/s to 20 m/s. As we can see in Figure 6.3.3.1., the routing overheads of AODV in all scenarios fluctuate without any rules within wide range. However, AODV-NCs show the stable increase as speed increases. Also, it is difficult to find the performance difference based on the movement model. For delivery ratio in Figure 6.3.3.2., clustered models show the different results from various connections. In 2-clustered layout, delivery ratio is increased as node moves fast. In other layout, delivery ratios tend to decrease for high mobility. Since nodes change positions quickly in high mobility in 2-clustered model, intermediate nodes between two partitions are changed. On the contrary, the delivery ratio in 3-clustered decrease as speed goes up. Also, delay in figure 6.3.3.3. is increased. As a result, the number of clusters is an important factor which affects the performance of routing protocols when it is considered with node speed.



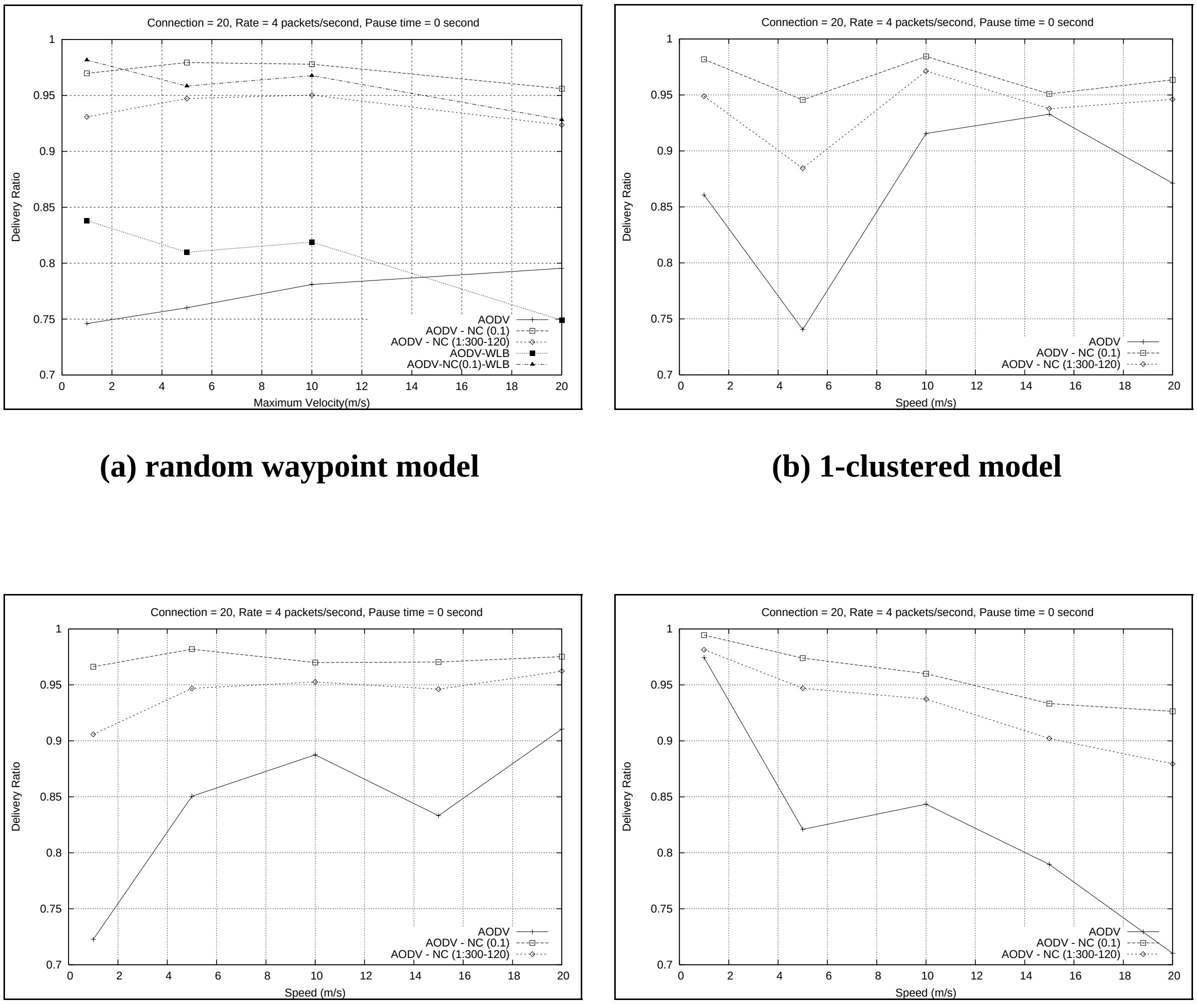
(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.3.1. The relative routing overhead over various speeds of nodes for different movement models.



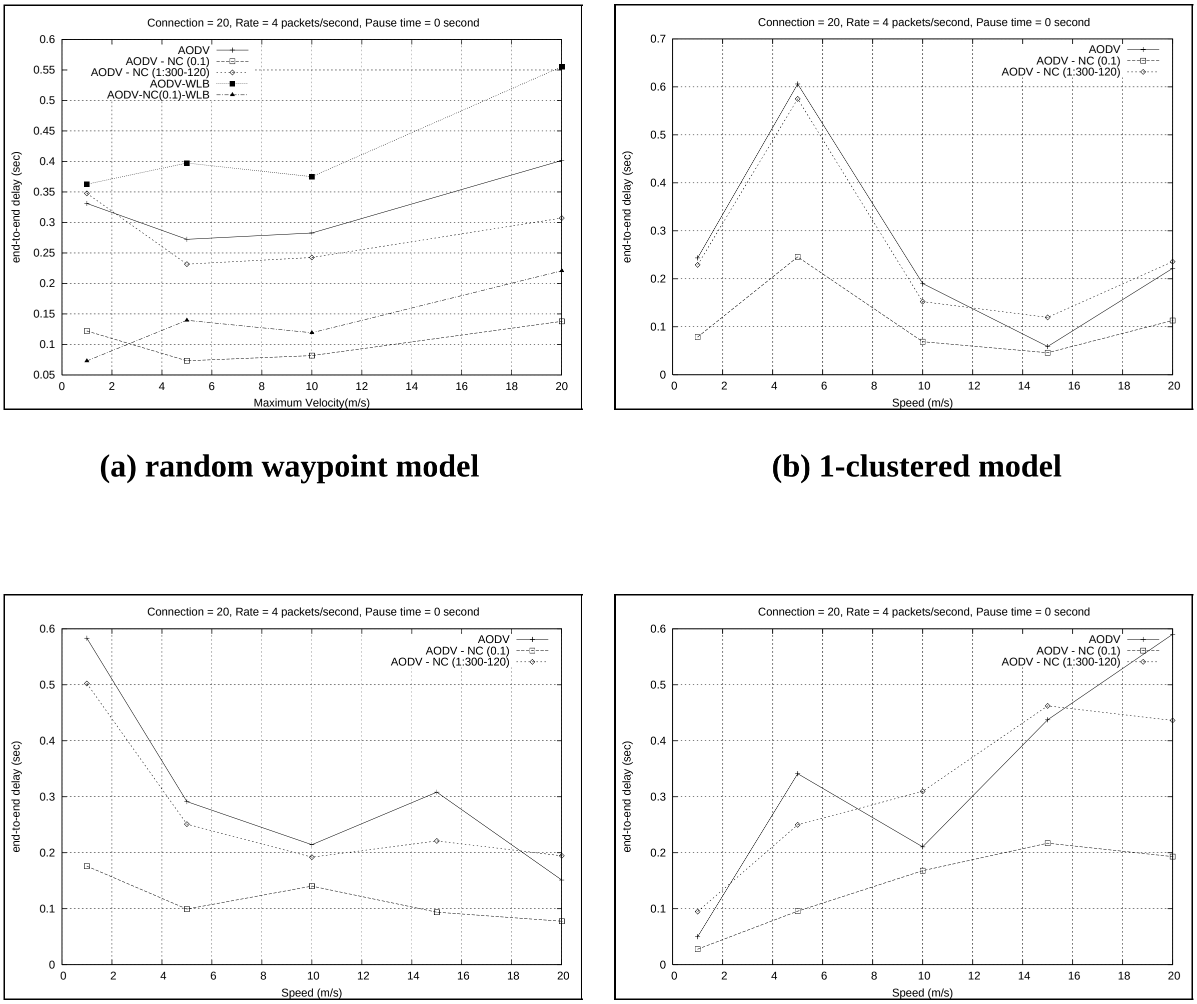
(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.3.2. The delivery ratio over various speeds of nodes for different movement models.



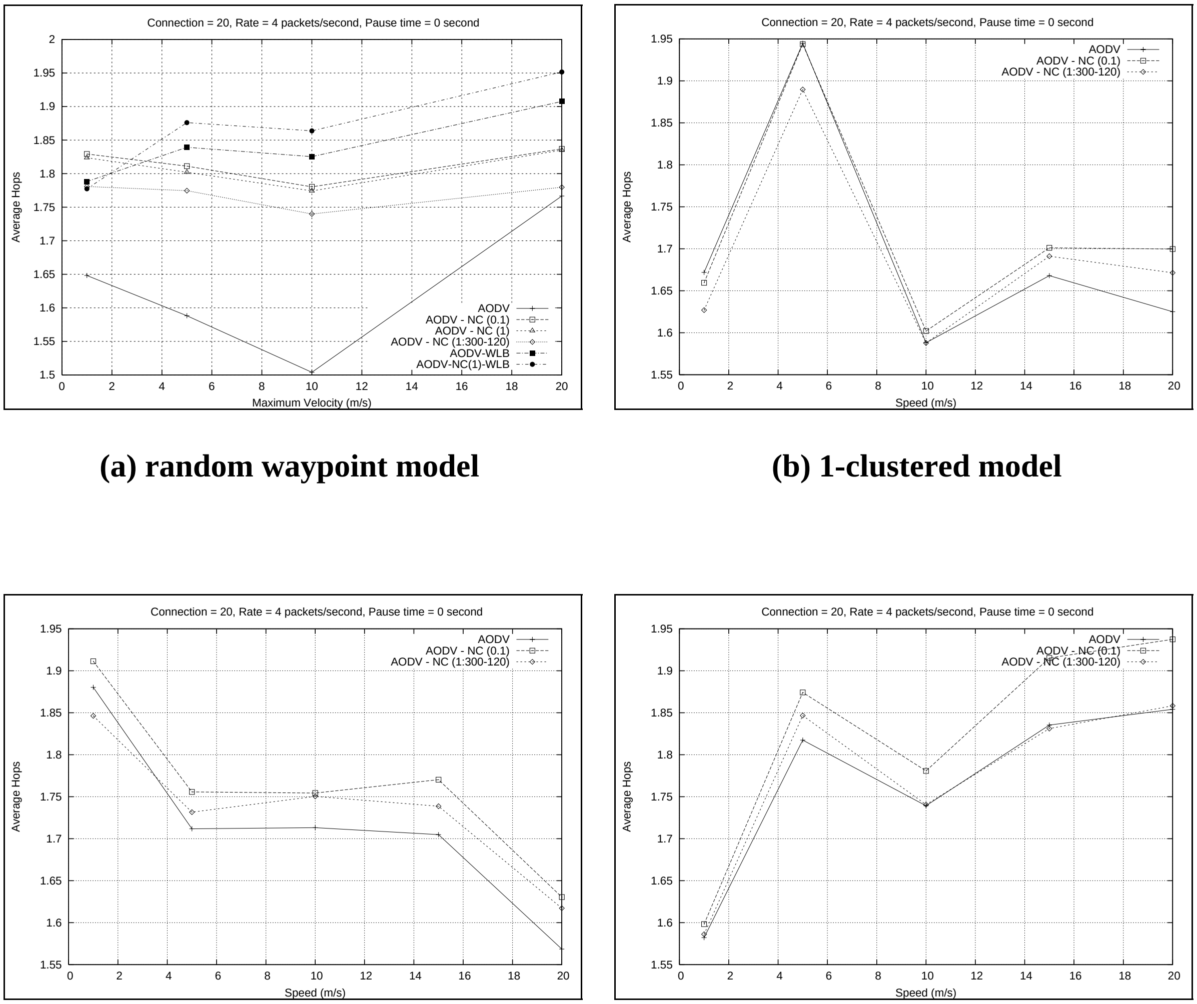
(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.3.3. The end-to-end delay over various speeds of nodes for different movement models.



(a) random waypoint model

(b) 1-clustered model

(c) 2-clustered model

(d) 3-clustered model

Figure 6.3.3.4. The average number of hops over various speeds of nodes for different movement models.

7. Modification in the Source Code of NS2

We modified the NS2 version 2.26 source code in order to integrate Node Caching algorithm with AODV. The source code, Tcl scripts and other documents can be downloaded from our website http://alla.cs.gsu.edu/~cscsjjx/.

7.1. Modification of AODV-NC, AODV-NC with load balancing

In order to implement Node Caching, we defined several attributes for the mobile node and the packet. Also, we have defined several TCL interfaces so that these attributes can be accessed by TCL script during runtime. Some of TCL interfaces were used for gathering results. Some of attributes had been defined for DSR-NC might be repeated in this section. We tried to use these attributes without additional definitions.

7.1.1. Attributes of the Node and the Packet

ns/common/node.h

We defined “tThreshold” for the time threshold and “tAge” to save the time when the node forwards a data packet. User has to set the value of “ tThreshold” through TCL script. It is used to decide the size of cached node. A node assigns the “tThreshold” value when it sends Route Request packet.

For AODV-NC (*H*: *n* –*t*), we defined three attributes: “MAX_FORWARD_PACKET,” “MAX_PACKET_PERIOD,” and “SLEEP_AWAKE_TIME.”

“MAX_FORWARD_PACKET” is the maximum number of forwarded packet during “MAX_PACKET_PERIOD.” The node forwarded the maximum number of packets within maximum period will not forward any constrained Route Request packet to neighbors during the

“SLEEP_AWAKE_TIME”. The values of attributes are also set through TCL script by a user.

For internal use, we defined “sleep_flag” which has value TRUE or FALSE. It indicates whether the node is sleep or awake status. Also, we defined many trace related counters to catch data generated during the simulation. These values are used through TCL script.

ns/common/node.cc

We bind attributes defined node.h with TCL variables.

ns/common/packet.h

We defined an attribute ‘tThreshold” in the packet header. A Route Request packet is delivered with the “tThreshold” value and the value is used when an intermediate node make decision of packet dropping or packet forwarding.

7.1.2. Modification in the AODV routing protocol

ns/aodv/aodv.h

We defined a new variable “thisnode” to identify the node itself.

ns/aodv/aodv.cc

In this file, we initialized all variables defined node.h and aodv.h with initial values at the command() function. And we added the module which performs node caching in the recvRequest() function. AODV, AODV-NC (*H*) and AODV-NC (*H*: *n* – *t*) are performed based on the user defined values. For example, if “tThreshold” is greater than simulation time, AODV protocol is applied. If “tThreshold” is less than simulation time, usually from 0.01 to 5, and “MAX_FORWARD_PACKET” is too large to be reached during the period, it goes to AODV-NC (*H*). If “MAX_FORWARD_PACKET” is set with a small value such as 100 or 300 or sleep_flag is set to TURE, AODV-NC (*H*: *n* – *t*) is applied.

For workload-based load balancing, we followed the algorithm described in []. To meet the convenience of test, we commented this module.

ns/aodv/aodv_rqueue.cc

We added a function which return the length of message queue for workload-based load balancing.

7.1.3.Data Collection from the Simulation Trace

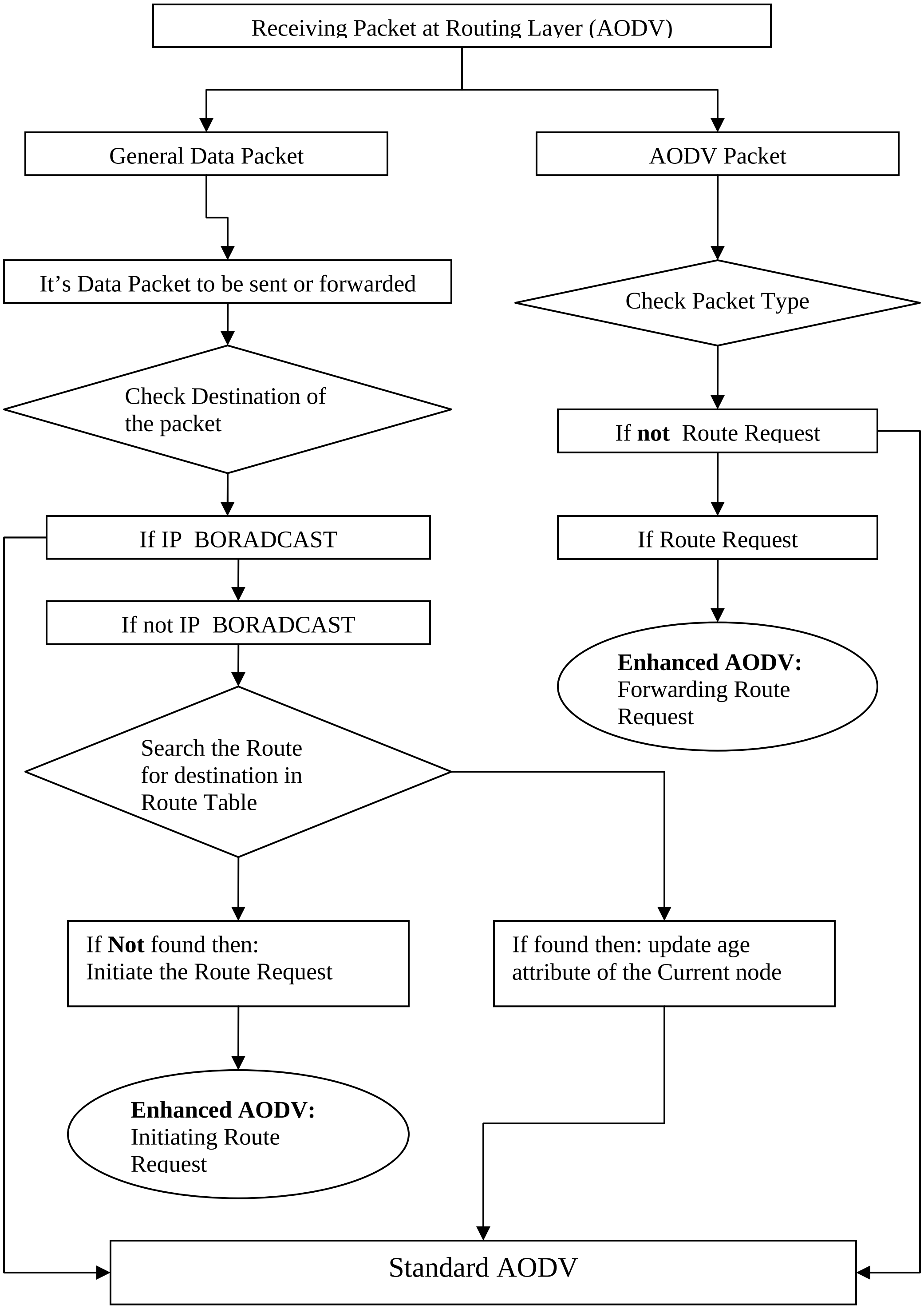
ns/trace/cmu-trace.cc

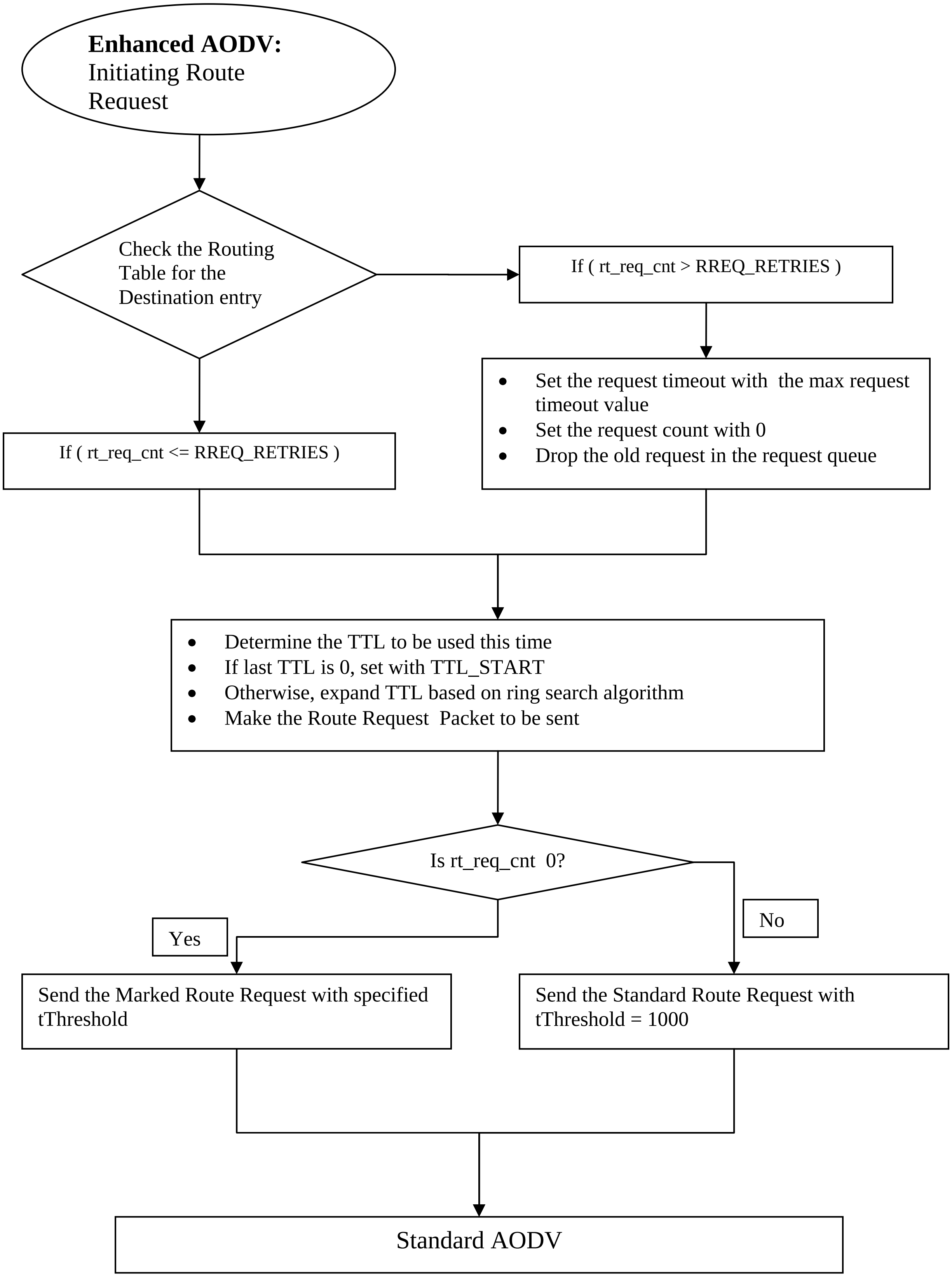
For data collection from the simulation, we defined several counter in node.h. Even though NS generates a trace file with detailed information of the simulation, it requires very large storage space. We had to keep several trace files to analyze performance for different scenarios. To avoid this problem, we made our performance comparison analysis on the fly by updating the associated counter which we need to measure. By this way, we were able to extract the results without analyzing huge trace files for every case.

Our modification is found in format () function.

ns/tcl/ns-default.tcl

Attributes defined in node.h are declared and assigned by default values.


Receiving Packet at Routing Layer (AODV)
General Data Packet
AODV Packet
It's Data Packet to be sent or forwarded
Check Packet Type
Check Destination of the packet
If **not** Route Request
If IP BORADCAST
If Route Request
If not IP BORADCAST
Enhanced AODV: Forwarding Route Request
Search the Route for destination in Route Table
If **Not** found then: Initiate the Route Request
If found then: update age attribute of the Current node
Enhanced AODV: Initiating Route Request
Standard AODV


Enhanced AODV:
Initiating Route
Request
Check the Routing
Table for the
Destination entry
If (rt_req_cnt > RREQ_RETRIES)
Set the request timeout with the max request timeout value
Set the request count with 0
Drop the old request in the request queue
If (rt_req_cnt <= RREQ_RETRIES)
Determine the TTL to be used this time
If last TTL is 0, set with TTL_START
Otherwise, expand TTL based on ring search algorithm
Make the Route Request Packet to be sent
Is rt_req_cnt 0?
Yes
No
Send the Marked Route Request with specified tThreshold
Send the Standard Route Request with tThreshold = 1000
Standard AODV

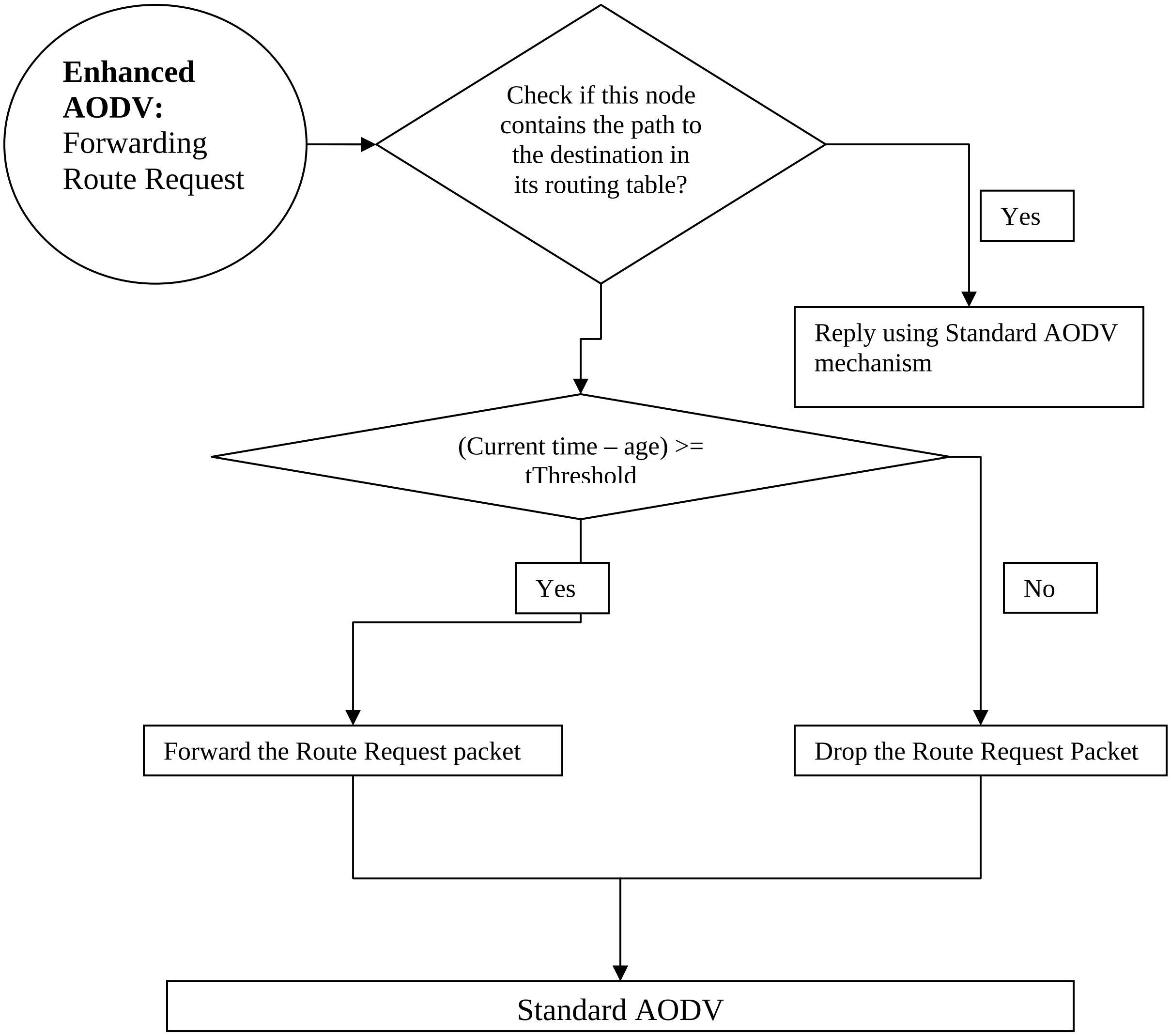

Enhanced AODV: Forwarding Route Request
Check if this node contains the path to the destination in its routing table?
Yes
Reply using Standard AODV mechanism
(Current time – age) >= tThreshold
Yes
No
Forward the Route Request packet
Drop the Route Request Packet
Standard AODV

Figure 7.1. Flow diagram of enhanced AODV

7.2. Clustered Movement Generator

We developed the *setdest_cluster* program which generates the clustered layout movement. It is based on the *setdest* program in NS2. Instead of generating the position and destination of nodes randomly, we place the nodes based on our algorithm described in section 5.

We add new functions: ClusterPosition(), ClusterMatching() and sort_degree(). Any other functions from *setdest* program are used without any changes. This generator requires the two more options than *setdest* program. One is the number of cluster and another is the alpha value of the power-law. The usage is as followed.

```
setdest_cluster  -n <nodes> -p <pause time> -s <max speed> -t simulation time –x <max X>
                 -y <max Y> -c <cluster number> -a <alpha>
```

Conclusion

In this thesis we introduce a novel node caching approach for constraining the route request protocol in ad hoc routing. We have implemented node caching enhancement AODV-NC which improves the original AODV in all three metrics – extensive simulation in NS2 show average decrease by 85% in relative routing overhead as well as average decrease by 63% in the delay, and average increase by 20% in the delivery ratio.

We compared our protocols with AODV-PA [6] on comparable test cases – while the both protocols have the same AODV delay, AODV-NC (1:300-12) improves delivery ratio of AODV-PA by 7% and has considerably larger routing overhead reduction (80% vs. 19%). Also, our improvement with AODV-NC(1:300-12) is in general larger than for the best-to-date protocol AODV-DS recently proposed in [10] which also considerably improves routing overhead (by 70%), but only slightly improves delivery ratio (by 5%).

We have also proposed a new measure of fairness of ad hoc routing protocols which depends on distribution of the forwarding load among nodes. The AODV-NC protocols are shown to be unfair and make certain overused nodes to exhaust their batteries prematurely. We suggest load balancing schemes that improve fairness and network lifetime of AODV-NC sustaining considerable improvement in routing overhead by 89%, delivery ratio by 17% and delay by 20%.

From the energy efficiency point of view, AODV-NC with adaptive workload-based load balancing showed the best network throughput and AODV-NC with non-adaptive load balancing showed the longest network lifetime by our new metrics. As a result, both non-adaptive and adaptive load balancing techniques combined with AODV-NC improved performance in energy efficiency as well as routing efficiency.

In the experiment with clustered layout model, we had interesting results of impact of node movement model. With two kinds of clustered layout model, AODV and AODV-NC showed the better performance in delivery ratio and delay than random waypoint model. However, with 2-clustered layout model, AODV and AODV-NC showed lower performance in overall metrics because of partitioning of clusters. Especially, performance degradation occurred in low mobility scenarios severely than in high mobility. Another important result with clustered layout model is that enhanced AODV protocols showed the stable experiment results than original AODV. AODV-NC maintained the routing overhead ratio under 1 in all simulation and the increase also was not prominent while AODV recorded fluctuate results for various scenarios.

We have shown that AODV-NC is an efficient and stable protocol by extensive studies and the performance evaluation of clustered layout movement with a large number of mobile nodes will be our future work.

Bibliography

[1] P. Sinha, R. Sivakumar, and V. Bharghavan, “Cedar: Core extraction distributed ad hoc routing,” in *Proc. of IEEE INFOCOM*, 1999.

[2] C. Perkins, Ed., *Ad Hoc Networking*. Addison-Wesley, 2001.

[3] K. Fall and K. Varadhan, “The vint project, uc berkeley, lbl, usc/isi, and Xerox parc,” 1997.

[4] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, “A performance comparison of multi-hop wireless ad hoc network routing protocols,” in *Proc. of the ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom98)*, 1998.

[5] C. Perkins, E. Royer, S. Das, and K. Marina, “Performance comparison of two on-demand routing protocols for ad hoc networks,”' in *IEEE Personal Communications*, February 2001.

[6] S. Gwalani, E. Belding-Royer, and C. Perkins, “Aodv-pa: Aodv with path accumulation.” in *Next Generation Internet Symposium –ICC*, May 2003.

[7] Y. C. Tseng, S. Y. N, Y. S. Chen, and J. P. Sheu, “The broadcast storm problem in a mobile ad hoc network,”, *Wireless Networks*, vol. 8, pp. 153 -167, 2002.

[8] B. Willams and T. Camp, “Comparison of broadcasting techniques for mobile ad hoc networks,” in *International Symposium on Mobile Ad Hoc Networking & Computing*, 2002, pp. 194-205.

[9] R. Castenada, S. R. Das, and M. K. Marina, “Query localization techniques for on-demand routing protocols for mobile ad hoc networks,”, ACM*/Kluwer Wireless Networks (WINET) Journal*, vol. 8, pp. 137-151, March 2002.

[10] M. Mosko, J. Garcia-Luna-Aceves, and C. Perkins, “Distribution of route requests using dominating-set neighbor elimination in an on-demand routing protocol,” in *Proceedings IEEE Globecom 2003 - Wireless Communications*, December 2003, pp. 1018--1022.

[11] Y. Ganjali and A. Keshavarzian, “Load balancing in ad hoc networks: Single-path routing vs. multi-path routing,” in *Proceedings of IEEE INFOCOM*, 2004.

[12] P. Pham and S. Perreau, “Performance analysis of reactive shortest path and multi-path routing mechanism with load balance,” in *Proceedings of IEEE INFOCOM*, 2003.

[13] D. Johnson and D. Maltz, “Dynamic source routing in ad hoc wireless networks,” in *In Mobile Computing, edited by T.Imielinski and H. Korth*. Kluwer Academic Publishers, 1996.

[14] Y. Lee and G. Riley, “ A workload –based adaptive load-balancing technique for Mobile ad hoc networks,” in *Proc. of IEEE WCNC*, 2005.

[15] V. Kavadia and P.R. Kumar, “Power Control and Clustering in Ad Hoc Networks,” in *Proc. of IEEE Infocom*, 2003

[16] A. Michail and A. Ephremides, “Energy efficient routing for connection-oriented traffic in wireless ad-hoc networks,” in *Mobile Networks and Apprications*, no. 8. Kluwer Academic Publishers, 2003, pp.517-533.

[17] C. Margi and K. Obraczka, “Instrumenting networking simulators for evaluating energy consumption in power-aware ad hoc network protocols,” in *International Symposium – MASCOT’04*, October 2004.

[18] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On Power-Law Relationships of the Internet Topology,” in *ACM SIGCOMM*, Cambridge, MA, September 1999.

[19] A. Medina, I. Matta and J. Byers, “On the Origin of Power Laws in Internet Topologies,” *ACM SIGCOMM Computer Communication Review*, Vol. 30, No. 2, April, 2000

[20] J.Yoon, M.Liu and B.Noble, “Random Waypoint Considered Harmful,” in *Pro. Of IEEE Infocom*, 2003

[21] C. Shah, “Distributed Algorithms in Ad hoc wireless networks”, master thesis, Georgia State University, 2003

[22] K. Calvert, M. Doar, and E. Zegura., Modeling Internet Topology. *IEEE Transactions on Communications*, pages 160–163, December 1997.

[23] C. Jin, Q. Chen, and S. Jamin., Inet: Internet Topology Generator. Technical Report Research Report CSE-TR-433-00, University of Michigan at Ann Arbor, 2000.