

# AN AODV-Based Clustering and Routing Scheme for Mobile Ad Hoc Networks

Mieso K. Denko and Hua Lu  
University of Guelph, Guelph, Ontario, Canada, N1G 2W1  
denko@cis.uoguelph.ca , hlu@uoguelph.ca

**Abstract.** A mobile ad hoc network (MANET) is a collection of wireless mobile nodes forming a temporary network without the aid of any fixed communication infrastructure. Due to limited resources, frequent network partitions and unpredictable topological changes, proactive clustering schemes incur high overheads in this environment. In this paper, we propose an on-demand, distributed clustering algorithm for MANETs based on an Ad hoc On-demand Distance Vector (AODV) routing protocol. The use of on-demand routing protocol information for clustering reduces clustering overhead because no clusters are maintained unless they are needed. The clustering algorithm's stability was assessed using clustering metrics such as cluster head and cluster members lifetime. Based on this clustering scheme, a cluster-based routing protocol was proposed to add scalability to the AODV routing protocol. Using simulation, a comparison was made with a pure AODV protocol. Simulation experiments show that the scheme results in stable and scalable clusters and Cluster-AODV routing introduces less overhead than the pure AODV protocol without clustering.

## 1. Introduction

Mobile ad hoc networking is characterized by highly dynamic network topology and limited system resources. A number of routing protocols have been proposed for routing in MANETs [1, 4, 8, 10]. In MANETs, performance may decrease dramatically when the network's size is beyond a certain threshold. As a result, many routing algorithms perform well only when the network's size is small. To overcome resource limitations such as bandwidth and battery power, and to reduce routing overhead, the organization of the network into smaller and more manageable partitions is necessary [14]. The clustering architecture provides three useful features in a MANET environment: network scalability, fault tolerance and reduction of communication overheads. Most existing clustering algorithms use either geographical regions as clusters or form new clusters proactively even if their

function is not needed [2, 3, 9]. The algorithm by Chatterjee et al [11] creates clusters on demand. However, this algorithm does not use the information maintained by a routing protocol.

We argue that if the routing algorithm is used as a means of gathering clustering information, the clustering and routing overhead can be significantly reduced. The AODV is one of the reactive routing protocols most commonly used in MANETs. Although the AODV protocol performs well with mobile nodes, it incurs high overhead with an increase in the network's size, the nodal degree or the number of communicating source-destination pairs. By using AODV route construction and maintenance mechanisms, clustering architecture can be constructed on demand. Clusters are maintained when data are to be sent. Such an integrated routing and clustering scheme can improve throughput and reduce routing overhead. The two main contributions of this paper are: (a) we propose a clustering architecture based on an extended AODV routing protocol for cluster formation, maintenance and purging operations; and (b) we propose an adaptive Cluster-AODV routing protocol that uses AODV and clustering information for quick route discovery, maintenance and packet delivery.

The remainder of this paper is organized as follows. Section 2 presents cluster formation and maintenance mechanisms. Section 3 presents the proposed clustering architecture and routing schemes. Section 4 presents performance evaluation and finally, Section 5 presents conclusions and future work.

## **2. Cluster Management**

A clustering architecture provides network scalability and fault tolerance, and results in more efficient use of network resources. It can be used for resource management, routing and location management to reduce communication and computational overhead. In this section, we discuss cluster formation and maintenance mechanisms.

### **2.1 Clustering Algorithm Design Goal**

We intend to integrate clustering with routing functionalities. The main design goals of our clustering scheme are:

1. The algorithm should use a routing protocol's control messages for cluster formation with minimal overhead.
2. The algorithm must operate in localized and distributed manners and interoperate with nodes running only AODV.
3. The algorithm must incur minimal cluster formation and maintenance overhead and support on-demand cluster formation.
4. The algorithm should minimize network-wide flooding and be scalable.

Our proposed scheme constructs or updates clustering architecture only when clusters' service is needed. The on-demand nature emanates from the demand driven nature of the AODV the scheme is based on. Nodes that take part in clustering are known from topological information maintained in the CHs and individual nodes.

## 2.2 Cluster Formation

The main purpose of clustering is to use the network resources more efficiently, enhance availability, reduce overheads and provide scalable architecture [7, 8, 14]. The choice of a clustering algorithm affects the clusters' stability. Our proposal divides the network into several two-hop clusters. In each cluster, a node can play one of five roles: cluster head, ordinary node, secondary cluster head, undecided node or gateway. A gateway is a node that can directly communicate with two or more clusters.

In each cluster, a cluster head (CH) is elected and responsible for cluster maintenance and inter-cluster and intra-cluster communication. A Secondary Cluster Head (SCH) is also elected to avoid the CH from becoming a bottleneck [5]. The SCH stores backup routing and cluster information. This role is rotated among other ordinary nodes. Its election does not require extra overhead because any node that wishes to serve as a SCH notifies only the CH and the CH informs other members of the cluster. In addition to a routing table, every CH maintains two tables, an intra-cluster node table and a k-hop cluster table. The intra-cluster node table contains the IDs of all nodes within a cluster. The k-hop table stores the IDs of the CHs of all other clusters located in a 2-hop neighborhood. The CHs are coordinators in each cluster and they store shared information. Every node periodically broadcasts a hello message to maintain information about its neighbors. To reduce periodic broadcasting overhead, new nodes or undecided nodes can learn their nearest CH on demand by sending a cluster head request packet. These nodes then act as either an ordinary node or a gateway based on its current location.

## 2.3 Cluster Head Election

Several distributed algorithms were proposed for CH election in MANETs [2, 3, 7, 9, 11, 14]. Chiang et al [2] have shown that the Lowest ID (LID) algorithm performs better than the cluster head election algorithms based on Highest Connectivity (HC). The proposals in [11, 14] use multiple criteria for CH election. Because a cluster head is responsible for cluster maintenance and intra-cluster and inter-cluster communication, it is expected to function for a long period of time once elected. Nodal mobility and link failure are the main causes of cluster head re-election and cluster membership changes.

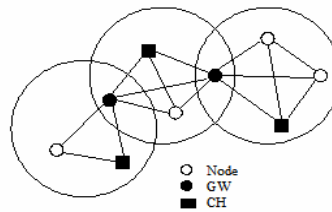


Fig 1. CH election

In [9], a mobility-based clustering algorithm for clustering was proposed. In this algorithm, a node is elected as a cluster head only when its mobility index is below a

certain threshold. The mobility index is computed based on cluster membership changes and the number of cluster head changes. In case of a tie, the node with the lowest ID is chosen. In our cluster election algorithm, the lowest ID clustering is initially used for cluster formation. Thus, a node is elected as a CH if it has the lowest ID. This forms initial node configuration. Later on, a node with a lower mobility index than its neighbors [9] is used as a criterion. For example, in Fig. 1, each node broadcasts its mobility information to its neighbors during the cluster head election phase. After collecting information from neighbors, each node checks whether it has the lowest mobility index. Once it confirms this, it sets itself as a cluster head and notifies its neighbors.

## 2.4 Cluster Maintenance

There are two parts to cluster maintenance: intra-cluster maintenance and inter-cluster maintenance.

**2.4.1 Intra-cluster maintenance.** In order to keep the neighbor table and CH information consistent, nodes broadcast and exchange hello messages periodically. A hello message contains information about a node's ID and roles. If no hello message is received from a neighbor during the ALLOW\_HELLO\_LOST interval, the neighbor is considered lost and is removed from the neighbor table. An ordinary node checks its neighbor table to verify whether a CH still exists. If a node finds that no CH exists, a new CH will be elected in the neighborhood. If a CH fails, local maintenance is carried out.

**2.4.2 Inter-cluster maintenance.** Each cluster head maintains a K-hop cluster table, which contains all k-hop CHs alive in a network. Each CH notifies other neighbor CHs that it is still alive by sending a HeadAlive message. A CH, say, CH1, receives a HeadAlive message from another CH, say, CH2. If CH1 finds out that CH2 already exists in its CH table, CH2's expiration time will be updated. Otherwise, a new CH entry of CH2 will be inserted and its expiration time will be set by adding the CH update time to the current time. If no HeadAlive message is received from a cluster head during a HEAD\_UPDATE\_INTERVAL interval, that cluster is considered unavailable. If no HeadAlive message is received during an ALLOW\_HEADALIVE\_LOST interval, the CH is considered unavailable and removed from CH table.

## 3. Implementation of the Cluster-AODV Clustering and Routing Schemes

A K-hop CH table maintains K-hop neighbor cluster information. We used  $k=2$ , where only two hop clusters are considered. The K-hop CH table maintains fields such as CH ID, CH status, cluster size, CH expiration time and number of times the hello message has been lost. The CH's expiration time is the current time plus the HeadAliveUpdate interval. A CH's status could be either "alive" or "not alive." A CH's status will be marked as "not alive" when its HeadAlive message is not

received by other CHs within its expiration time. A CH entry will be removed from a K-hop CH table if the number of times that the HeadAlive message has not been received exceeds three. In the following section we will present the various data structure added to the OADV to implement the proposed clustering architecture.

### 3.1 Hello Message and Neighborhood Maintenance

Before a hello message is sent, the sending node adds its status information to the message. The hello message sender could be a CH, a gateway, an ordinary node or an undecided node. The hello message extension includes source address, lifetime and current status.

We use a neighbor table to store the neighbor's ID, expiration time, status and the number of times the hello message has been lost. When a node receives a hello message from a neighbor, it will check to see whether this neighbor exists in the neighbor table. If the existence is confirmed, the neighbor's expiration time will be updated. Otherwise, a new entry will be added. When a hello message has not been received from a node in three subsequent hello intervals, it is considered not connected and will be deleted from the neighbor table.

### 3.2 Cluster Management Module

To implement a clustering algorithm based on AODV, we use the following three modules.

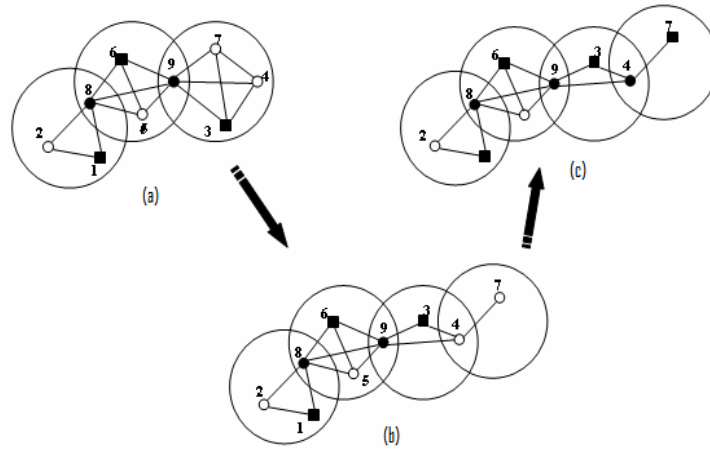


Fig. 2. Cluster re-election when no cluster exists in the neighbor table

**3.2.1 Cluster initialization module.** The cluster formation module is used to initialize clusters shortly after AODV has been initialized. The module is invoked in the CLUSTER\_FORM\_PERIOD, which is a predefined threshold. Before the initialization, every node sets its status to “undecided” and broadcasts a hello message to its neighbors. As soon as a node has received all the hello messages from its neighbors, it invokes the Cluster Head Election module to elect a CH. At this stage of cluster formation, an LID clustering algorithm is used. The new CH will then notify its neighbors and other CHs. When a node receives a CH existence notification, it sets its status to an ordinary node and broadcasts a hello message to its neighbors. If one node finds two different CHs in its neighbor table, that node becomes a gateway. After the Cluster Formation Module’s execution, every node will belong to at least one cluster.

**3.2.2 Cluster head election module.** A node will invoke the Cluster Head Election module to elect a cluster head when no CH exists or when multiple CHs come close to each other. The procedure involves three steps as follows: (1) A node checks to see whether it has the lowest mobility index among its non-gateway neighbors. It sets its status to CH and broadcasts its status to its neighbors; (2) If it has the lowest mobility index, it declares itself a CH and notifies other nodes. Otherwise, it sets its own status to ordinary node.

We monitor clusters to avoid both the formation of multiple CHs and the absence of a CH in the cluster. We illustrate this with examples. Fig. 2(a) shows the topology of a hypothetical clustered-network at time  $t_1$ . In Fig. 2(a), nodes 1, 3 and 6 are CHs. Nodes 8 and 9 are gateways while nodes 2, 4, 5 and 7 are ordinary nodes. Assume that at time  $t_2$  the network’s topology changes to Fig. 2(b), where node 7 has moved out from its original cluster. When node 7 checks its neighbor table, it cannot find a cluster head. Then node 7 invokes the Cluster Head Election module to elect a new CH. The election result is shown in Fig. 2(c), where node 7 becomes a new CH and node 4 becomes a new gateway.

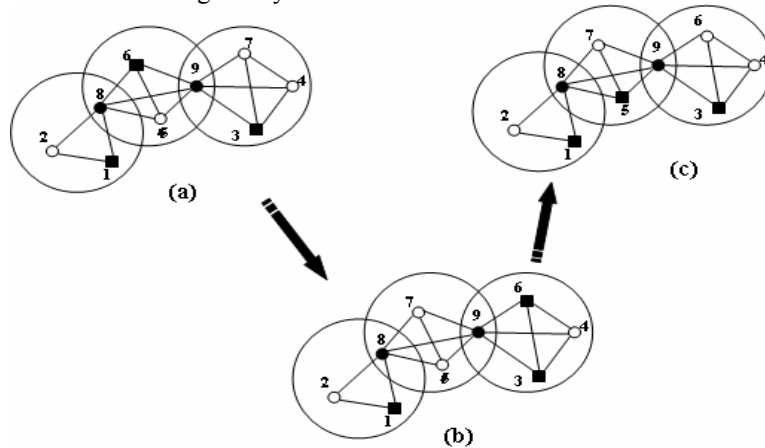


Fig. 3. Cluster re-election when more than one CH exists

Fig. 3 shows a situation where more than one CHs exist in one cluster as shown in (a). If the network’s topology changes from Fig. 3(a) to Fig. 3(b) at time  $t_2$ , nodes 6

and 7 exchange positions. Nodes 6 and 3 will find that two CHs exist in their neighbors, as shown in Fig. 3(b). Then both nodes 3 and 6 will invoke the Cluster Head Election Module to determine a new cluster head. After cluster election, node 6 abandons its CH status and sets itself as an ordinary node. Node 3 remains as a CH. Also, the cluster with nodes 5, 7, 8 and 9 elects node 5 as a new CH. The network's topology after CH re-election is shown in 3(c).

**3.2.3 Neighbor table purge module.** Because every node is mobile and the network's topology changes frequently, clusters formed in network may not be stable. Clusters' size and nodes' status may differ at any given time. In order to keep the cluster information fresh and updated, the neighbor table will be purged.

### 3.3 Cluster-AODV-based Routing

The AODV protocol sends many small packets compared to other reactive protocols such as DSR. Hence when the network's size increases, the degree of node also increases, causing network congestion. The use of clustering reduces this overhead by allowing localized route discovery and maintenance. The proposed Cluster-AODV scheme uses clustering architecture and AODV functionalities to perform routing. In this section, we will discuss the mechanisms used by Cluster-AODV to reduce routing overhead and allow scalability while achieving a good packet delivery ratio.

**3.3.1 Intra-cluster routing.** Intra-cluster routing involves routing within a cluster. Each node maintains routing information about its cluster. When a node does not have a route to a destination which is also in a cluster, however, it sends a Local Route Request (LRREQ) through the cluster. When there is no RREP due to route failure, local route maintenance is performed within a cluster.

**3.3.2 Inter-cluster routing.** Inter-cluster routing involves routing between clusters. The CH maintains 2-hop cluster topology. This 2-hop cluster topology is also maintained in a SCH to minimize the problem of one point of failure. When routes cannot be found within a cluster after the issuance of the LRREQ message, a CH uses a traditional RREQ message to search for a destination through a gateway to its 2-hop neighbor clusters. To reduce the overhead caused by the RREQ flooding packet, only gateways and CHs are involved in forwarding the RREQ. No ordinary nodes are involved in RREQ packets in the inter-cluster communication.

**3.3.3 Route maintenance.** Route maintenance is similar to cluster maintenance. When a route fails within a cluster, it is re-constructed locally using LRREQ and RREQ using the 2-hop topology information. When LRREQ fails, the AODV procedure is invoked and the traditional RERR is sent to the source node to reconstruct routes. The source node also follows a similar procedure to repair the failed routes, first locally and then with incremental scope.

The new node joining and the existing node leaving processes are carried out based on the AODV hello messages. When the CH exchanges neighbourhood information with members of its cluster, any new node in close proximity can register with the CH by using an RREQ message. When a node registers with two

CHs it acts as a gateway. When a node moves away from the current CH, it changes its role to an ordinary node, a gateway or undecided, and it will be removed from the old CH. The old members' routing entry is updated accordingly.

## 4. Performance Evaluation

### 4.1 Simulation Environment

The NS2 simulation tool [13] is used for performance evaluation. At the beginning of the simulation, 100 nodes were randomly placed within the simulation area of 1000m x 1000m. The transmission range was set at 250m. The random waypoint mobility model [12] was used for simulating mobility. The pause time was 40 seconds. Other simulation parameters are shown in Table 1.

Parameter	Default values
Node Speed (m/sec)	0-20
Transmission Rate	4 pkts/sec
Traffic Model	CBR, 30 sources
Packet Size	512 bytes
Simulation Time (sec)	1000

Table 1. Simulation parameters

### 4.2 Performance Metrics

We use three different metrics to evaluate the performance: Clustering stability normalized routing overhead and packet delivery ratio of the proposed clustering and routing scheme.

1. **Clustering stability.** In the first set of experiments, we investigated the clustering stability by monitoring the percentage of CH changes, the CHs' and cluster members lifetime, and variation in cluster size. The less frequently the cluster members change their membership and the longer they stay in the cluster, the more stable the clusters are. These metrics are used to evaluate the clustering architecture's suitability for robust route construction.
2. **Normalized routing overhead.** In the second set of experiments we investigated the normalized routing overhead, which is defined as the ratio of the total number of control packets to data packets. This metric is used to evaluate scalability.
3. **Packet delivery ratio (PDR).** In the third set of experiments we investigated the normalized routing overhead packet delivery ratio, defined as the ratio of the total number of packets successfully delivered to the total number of packets generated. This metric is used to evaluate the routing scheme's transmission reliability.



### 4.3 Discussion of Simulation Results

In this section we present the results of the experiments based on the above simulation parameters. Each data point in the graph represents an average of ten simulation runs. First, the Cluster-AODV was compared with LID and HC. Then the pure AODV routing was compared with the Cluster-AODV routing scheme.

**4.3.1 The stability of the clustering algorithm's effect.** The results in Fig. 4 show that the average cluster member residence time is between 90 and 130 seconds in the LID and Cluster-AODV clustering schemes. The minimum value for cluster member residence time was observed as 110 seconds for Cluster-AODV. This is due to the fact that the network's topology changes more frequently when the mobile nodes' speed increases. However, the difference in cluster membership time is similar in all simulated network sizes. Fig. 5 shows that the average CH lifetime is between 128 and 181 seconds in the LID and Cluster-AODV clustering schemes. The minimum value for CH lifetime was observed as 137 seconds for Cluster-AODV. The CH and CH members lifetimes is the longest in Cluster-AODV compared to the other two schemes.

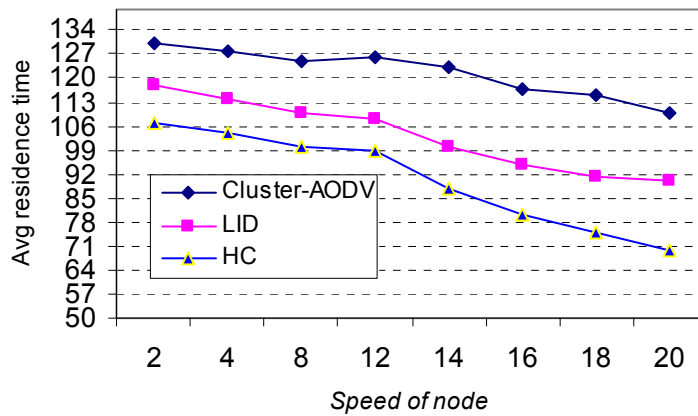


Fig. 4. Average cluster membership time (sec)

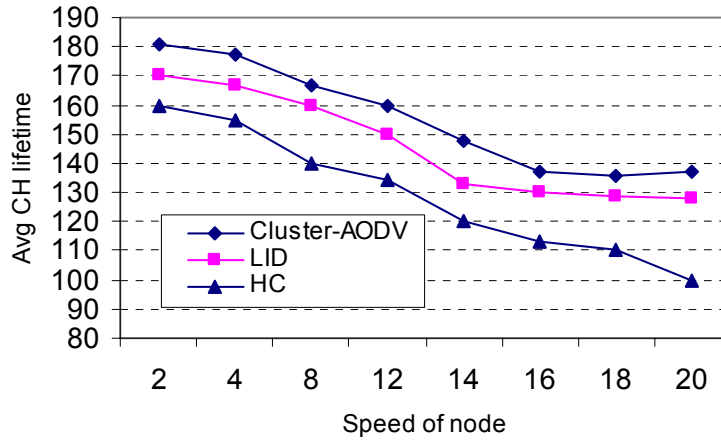


Fig. 5. Average CH lifetime (sec)

We have investigated the clustering algorithm's performance using the percentage of CHs elected and the cluster size's standard deviation. Larger cluster size results in a higher load at the CH, while smaller cluster size underutilizes the available resources. Fig. 6 shows the effect of transmission range on percentage of CHs in network with sizes of 150 and 200 nodes. The results show that when the transmission range increases, the percentage of CHs in the network decreases. The decrease in CHs is due to the fact that when the transmission range increases, a CH's coverage area increases so more nodes come into its transmission range. Because a CH can communicate only with its direct neighbors, the increase in network size results in more nodes being out of the existing CHs' reach, and hence requires more clusters to be formed. The percentage of CHs is similar for both network sizes at higher transmission ranges. The percentage of CHs varies only a little when the network's size rises from 150 nodes to 200 nodes at a higher transmission range. Fig. 7 shows that the variation in the number of nodes per cluster (cluster size) is similar at both network sizes and increases with the transmission range. This implies that the cluster's size increases at approximately the same rate for both network sizes.

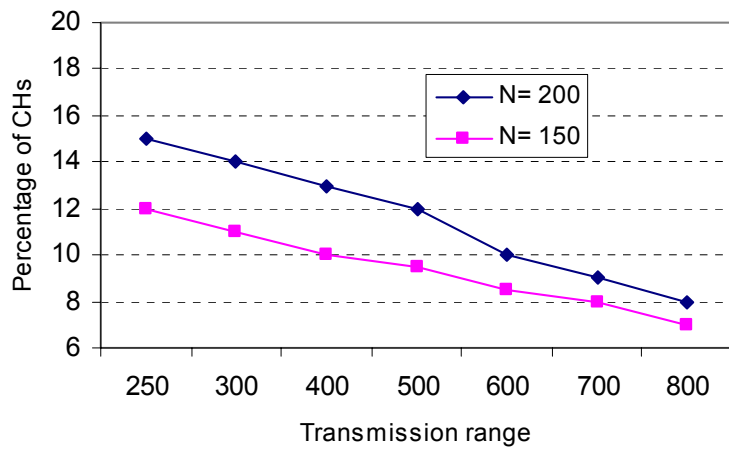


Fig. 6. Percentage of CHs

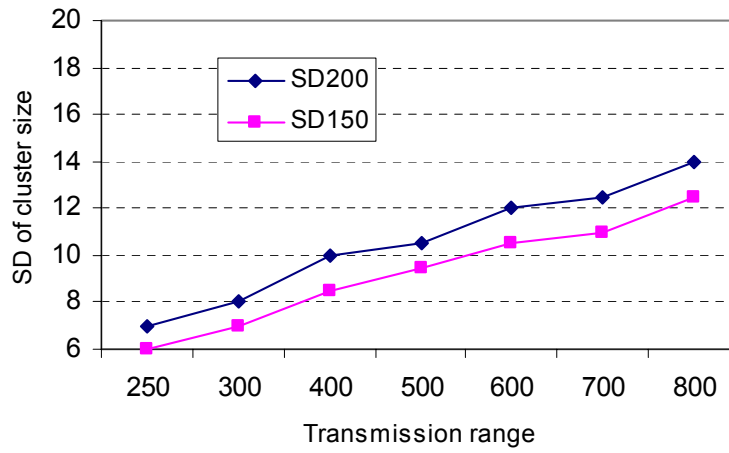


Fig. 7. Standard deviation of cluster size

**4.3.2 The analysis of normalized routing overhead.** To evaluate the overhead caused by the Cluster-AODV based routing protocol and the pure AODV routing protocol, we carried out experiments as a function of mobile nodes' speed. Networks of 150 and 200 nodes were used in the experiments. Figs. 8-9 show normalized routing overhead for AODV and Cluster-AODV based routing. Both results show that the routing overhead increases with the nodes' speed but the Cluster-AODV outperforms the pure AODV protocol at larger network sizes. Cluster-AODV has a lower overhead because it generates less control traffic overhead due to its localized and distributed control traffic handling, unlike the pure AODV protocol. Moreover, in the 200 node network, the Cluster-AODV has shown more stability than the pure AODV protocol. This confirms both the robustness and the scalability of the Cluster-AODV protocol.

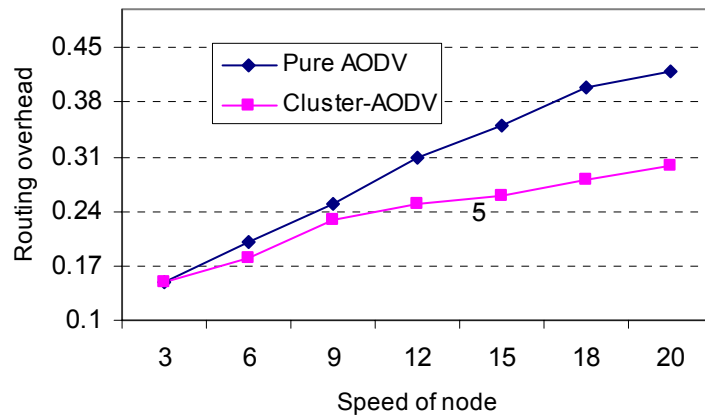


Fig. 8. Routing overhead (N=150)

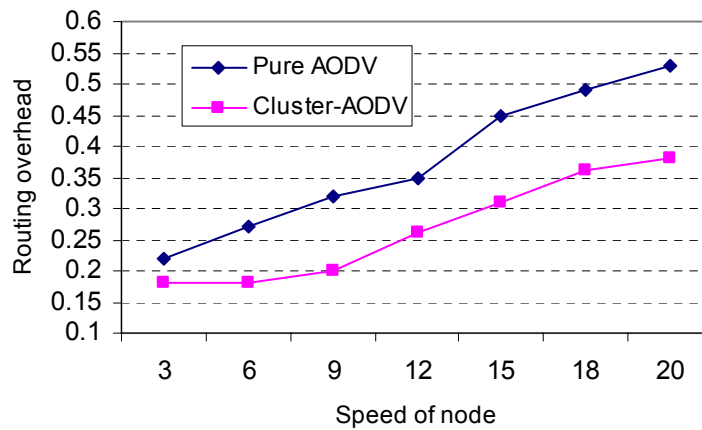


Fig. 9. Routing overhead (N = 200)

**4.3.3 The analysis of packet delivery ratio.** Fig. 10 shows that the packet delivery ratio is higher for the Cluster-AODV than the pure AODV protocol. The difference in the delivery ratios increases as the network's size increases, which shows the performance gained by the Cluster-AODV based routing scheme. This performance gain is attributed to the localized cluster-based route discovery and maintenance mechanism, and the effective use of AODV control messages in the Cluster-AODV protocol.

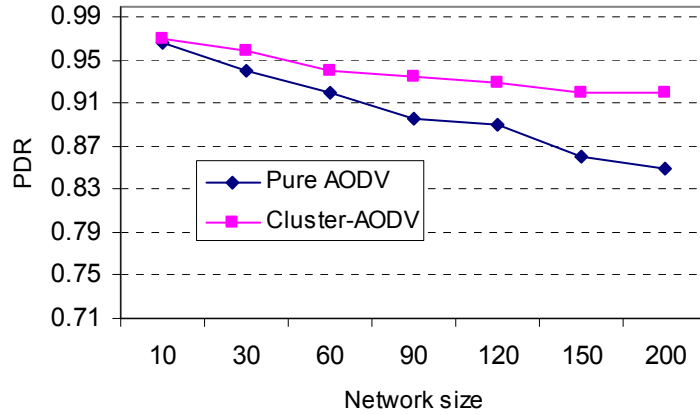


Fig. 10. Packet delivery ratio (N=200)

## 5. Conclusions and Future Work

This paper presents an AODV-based clustering and routing scheme for MANETs. The scheme is used for integrated routing and message delivery in clustered networks. A clustering architecture improves the network's scalability and fault tolerance, and results in a more efficient use of network resources. We evaluated the purposed clustering architecture using simulation experiments. The simulation results show that the algorithm builds stable clusters with low communication overhead due to its localized, distributed and reactive nature.

Our current study has built a basic framework for reactive clustering. Our future work includes building a cluster-based middleware for data dissemination and replica management among clusters. We also intend to extend the clustering architecture to support multihop clustering in MANETs.

## References

1. I.D. Chakeres and E.M. Belding-Royer, AODV Routing Protocol Implementation Design, Proceedings of the International Workshop on Wireless Ad hoc Networking (WWAN), Tokyo, Japan, March 2004, pp. 698-703.
2. C.C. Chiang, H.K. Wu, W. Liu and M. Gerla, Routing in Clustered Multihop, Mobile Wireless Networks With Fading Channel, Proceedings of IEEE Singapore International Conference on Networks SICON'97, pages 197-211, Singapore, Apr. 14-17, 1997.
3. C.R. Lin and M. Gerla, Adaptive Clustering for Mobile Wireless Networks, IEEE Journal on Selected Areas in Communications, Vol. 15, No. 7, pp. 1265-1275, September 1997.
4. DSR internet draft (2004) of Internet Engineering Task Force (IETF); <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt>.
5. H. Lu and M. K. Denko, Reliable Data Storage and Dissemination in Mobile Ad hoc Network, Proceedings of International workshop on Theoretical and Algorithmic Aspects of Wireless Ad Hoc, Sensor and Peer-to-Peer Networks, June 2004, Chicago, USA.
6. I. Stojmenovic I. and J. Wu, Broadcasting and Activity Scheduling in Ad Hoc Networks, in: Ad Hoc Networking, S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, eds., IEEE Press, 2004.
7. M. Gerla and J. Tsai, Multicluster, Mobile, Multimedia Radio Network, Wireless Networks, vol. 1, no. 3, 1995, pp. 255-65.
8. M. Jiang, J. LI, Y. C. Tay, Cluster Based Routing Protocol (CBRP) Functional Specification Internet Draft, draft- ietf-manet-cbrp.txt, work in progress, June 1999.
9. M.K. Denko, Analysis of Clustering Algorithms in Mobile Ad Hoc Networks, Proceedings of International Conf. on Wireless Networks, pp. 98-105, 2003.
10. P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot, T. Clausen, Optimized Link State Routing Protocol Internet Draft, draft-ietf-manet-olsr-04.txt, work in progress, June 2001.
11. M. Chatterjee, S. K. Das and D. Turgut, An On-Demand Weighted Clustering Algorithm (WCA) for Ad-Hoc Networks, Proceedings of IEEE GLOBECOM 2000.
12. J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu and J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, Proceedings of Mobile Computing and Networking (MOBICOM), 1998, pp. 85-97.

13. S. McCanne and S. Floyd, NS2; <http://www.mash.cs.berkeley.edu/ns/>.

14. F. Garcia, J. Solano and I. Stojmenovic, Connectivity based k-hop clustering in wireless networks, *Telecommunication Systems*, 22: 1-4, 205-220, 2003.