On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering

Mario Gerla, Taek Jin Kwon and Guangyu Pei

Computer Science Department
University of California, Los Angeles
Los Angeles, CA, 90095

Abstract – **This paper presents on-demand routing scalability improvements achieved using a "passive" clustering. Any on demand routing typically requires some form of flooding. Clustering can dramatically reduce transmission overhead during flooding. In fact, by using clustering, we restrict the set of forwarding nodes during flood search and thus reduce the energy cost and traffic overhead of routing in dynamic traffic and topology environments. However existing "active" clustering mechanisms require periodic refresh of neighborhood information and tend to introduce quite a large amount of communication maintenance overhead. In this paper, we introduce a passive clustering scheme which is mostly supported/maintained by user data packets instead of explicit control packets. The passive scheme is consistent with the on-demand routing philosophy. Simulation results show significant performance improvements when passive clustering is used.**

1. INTRODUCTION

Clustering in wireless ad hoc network has been investigated in order to enhance network manageability, channel efficiency [1,4], energy efficient communication [2], and to provide routing or multicasting scalability [3]. Clustering is indispensable for hierarchical routing or multicasting protocols rely on clustering for scalability. However clustering requires high refresh rate, and therefore even more overhead in realistic ad hoc environments because of unreliable links and nodes.

In this paper, we identify the limitations of conventional clustering in a realistic wireless ad hoc environment, and provide solutions which overcome such limitations by using a novel clustering protocol with a new clusterhead election rule. We call the clustering mechanism "*passive clustering*" because it does not require periodic, background protocol-dependent control packets or signals. Improvements of an on-demand routing (Ad Hoc On-Demand Distance Vector -AODV) will be presented as a example of passive clustering application to demonstrate the effectiveness of clustering in large scale ad hoc network.

2. CLUSTER LIMITATIONS

Unlike in a simulation environment, global information regarding node locations and adjacency relations is hard to collect in a realistic wireless ad hoc network. The major reason of the difficulties comes from unreliable and limited link capacity and from node mobility. Node locations and neighborhood information are keys for clustering; they do vary in time. Without the help of a special node- say "oracle"-which can listen or speak to all the nodes at the same time, adjacency (neighborhood) information can only be collected by exchanging beacons or hello messages. To ensure the correct collection of neighborhood information, previous clustering solutions assume repeated broadcasting of the neighbor list

In this period of neighbor learning and initial clustering, it is essential that there is no mobility for proper convergence. The quasi-stationary assumption must hold during the adjacency information collecting period, initial clustering, and the re-clustering or clustering maintenance period. The reason for the quasi-stationary assumption is obvious when we collect adjacency information. If there is motion, we may have to deal with stale neighborhood information. Moreover, mobility causes adjacency relation changes, which may trigger re-clustering throughout the network. Figure 1 demonstrates the case. The dotted circles represents the result of Lowest ID algorithm clustering [1] while node 1 is out of node 3's range. As node 1 becomes adjacent to node 3, node 3 abdicates clusterhead's role, and triggers a "chain reaction" causing a change in all clusterheads.

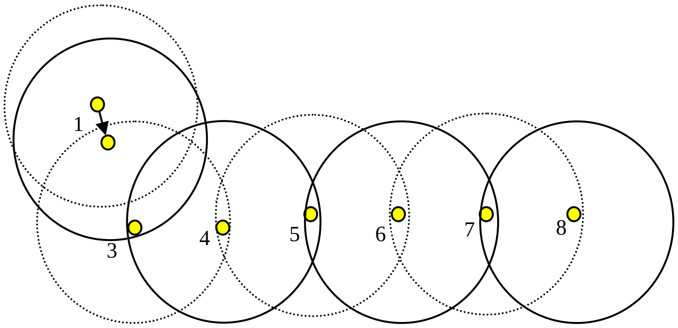

Figure 1. Chain effect when node 1 approaches to 3.

The quasi-stationary assumption is a necessity for stable operation in previous clustering mechanisms in which clusterhead election rules are weight driven (e.g. ID, degree) and are strictly enforced all the time. In section 3, we will address a new clusterhead election rule which overcomes these problems and does not require the quasi-stationary assumption.

In a wireless environment, it is hard to ensure that a node has collected all the neighborhood information (a complete neighbor list) even without node death/birth or mobility. By assuming uniform radio transmit power (symmetric links all the time) and perfectly coordinated communications (no collision, nor hidden terminal problems, etc) to assume perfect reception of entire neighbor lists, collecting the complete set of adjacency information is then feasible.

The isolation problem is another difficult problem to solve. The isolation problem refers to cluster disconnection while there is a feasible radio path. Figure 2 shows an example of isolation. Lowest ID algorithm ends up with solid circled clusters, which do not provide connectivity between node 5,6 and node 7,2. If node 6 were elected clusterhead, and node 7 became a gateway, the four nodes would be connected in the clustered structure. With weight driven clustering (like lowest ID), the problem can be solved only by using ad hoc extensions of the basic algorithm. For example, 6 and 7 become "split" gateways and jointly provide the required connectivity.

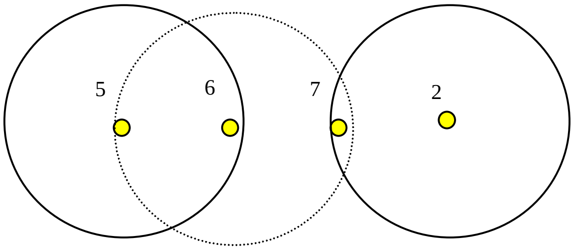

Figure 2. Isolation problem

In next chapter, we introduce a new clustering mechanism, which virtually eliminates all clustering line overheads, is free from quasi-stationary assumption, and can provide consistent solutions to the isolation problem.

3. PASSIVE CLUSTERING

Passive Clustering is a cluster formation protocol, which does not use dedicated, protocol specific control packets or signals. Conventional clustering algorithms require all participating network nodes to advertise neighbor information repeatedly. In an ad hoc network, collecting the neighbor information usually takes $O(k)$ communications where k is the possible maximum degree. This overhead is considered expensive in ad hoc wireless networks where bandwidth is limited.

Even worse, all the existing clustering schemes require an initial clustering phase prior to any network layer activities. With passive clustering, we can avoid both of the above disadvantages.

The key idea beyond passive clustering is to opportunistically exploit the neighborhood information carried by data packets. For example, by monitoring MAC level packets, which piggyback sender state, we can build a "soft state" cluster infrastructure. Note that massive exchange of cluster specific control packets is necessary for weight based clustering. With this new approach, no matter what routing protocol is used, the cluster infrastructure can be constructed as a by-product of user data packet exchanges.

Surprisingly, passive clustering can form better clusters than any conventional clustering schemes based on weight (e.g. ID, degree, and etc). This is because passive clustering utilizes data traffic information leading to solutions, which are relatively immune from logical isolation and lack of connectivity. Clustering stability is one of the major benefits of *passive clustering*; another is convergence time. To improve clustering stability and to speed up convergence, we developed a new clusterhead election rule, which is not based on weight. We call this rule "*first declaration wins*".

In the case of Lowest ID clustering, the lowest ID node (that can be a clusterhead) becomes one. However there is no guarantee that a clusterhead is always the lowest ID in the cluster - the lowest ID node may be a gateway.

This case occurs in every weight based clustering and enforcement of weight leads to a not very stable cluster structure because of continuous re-clustering attempts to satisfy the weight rules.

With the *first declaration wins* (FDW) rule, the node that sends a packet first, becomes the clusterhead and "rules" the rest of nodes in its clustered area (radio coverage). The new clusterhead election rule does not require re-clustering to maintain the weight criteria. As described in section 2, isolation problems can not be solved with existing weight based clustering methods. On the other hand, FDW can accommodate the following distributed isolation solution. *If there is no gateway in a cluster for a timeout period, the clusterhead resigns, and the rest of nodes in the cluster compete for the clusterhead position(s).*

When a node is ready to be a clusterhead, it declares its readiness by sending its clustering state claim in the MAC packets. Since *passive clustering* does not support explicit control packets or signals of its own, a clusterhead-ready node must postpone its claim until it has data traffic. After a successful transmission from a newly volunteered clusterhead, every node in the radio coverage can learn the presence of a clusterhead by monitoring packets from it. At this point, it records the clusterhead information and changes its clustering state appropriately. The readiness of being a clusterhead is determined by network activities as well as by the node state. After a clusterhead successfully asserts its state, it functions as a clusterhead, i.e. it collects the neighbor information by monitoring the network traffic. Because of the passive nature, the neighborhood information is kept in soft state - by time stamping, we preserve the freshness of information. Ordinary nodes or gateways keep a list of their clusterhead(s) in soft states. The timeout period has to be carefully chosen based on the node mobility and communication pattern. Non-clusterhead nodes can collect their own clusterhead(s) information by monitoring MAC level traffic. If a packet is from a clusterhead (after checking the status information in the packet), non-clusterhead nodes compare the sender ID of the packet with their own list and react accordingly.

4. THE PASSIVE CLUSTERING ALGORITHM

The passive clustering and maintaining procedure is simple and easy to implement, and fully distributed. When a node joins a network, it starts with INITIAL state. Like conventional LID clustering, passive clustering requires a "clustering" sub layer. Usually such clustering is included in the Network layer in the existing clustering. Passive clustering resides in MAC layer to utilize the source ID of MAC packets to collect neighbor information, and to advertise the node's clustering state (4 possible states). Clustering status information (2 bit for 4 states) is piggybacked on MAC packets. This is the extra overhead required by passive clustering. Also, the clustering sub layer monitors all the received MAC packets and strips off the cluster information if the node is the destination.

By embedding/stripping the clustering status information, we can use passive clustering without any modification of MAC protocol or network protocol. "First declaration wins" rule is applied for initial clustering, and clustering maintenance.

A node can not be a clusterhead if it has been "claimed" by another clusterhead, i.e. it has received a MAC packet from another clusterhead. It implies that the distance between any two clusterheads is at least 2 hops.

We summarize the passive clustering algorithm as follows,

1. There are 4 possible states and 1 internal state; INITIAL, CLUSTERHEAD, ORDINARY_NODE, GATEWAY, and CLUSTERHEAD-READY.
2. Below the MAC layer, a cluster sub layer (CSL) adds 2 bits of cluster state information on every outgoing MAC packet. If current state of the node is CLUSTERHEAD-READY, it changes state to CLUSTERHEAD before it tags state information. For incoming packets, the CSL strips cluster state information of the sender, and extracts the sender id information, performs passive clustering, then passes the packet to MAC layer.
3. At cold start, every node is in INITIAL state. There is no state change until a node receives a MAC packet. If the sender state is not CLUSTERHEAD, then its cluster state turns into CLUSTERHEAD-READY. The CLUSTERHEAD-READY node will be a clusterhead if it successfully transmit an outgoing packet before it receives any packets from another clusterhead. If the packet was from another clusterhead, i.e. sender state is CLUSTERHEAD, add the clusterhead information (id, and reception time) to the node's clusterhead list, and goes into ORDINARY_NODE state.
4. All the nodes in any state other than CLUSTERHEAD maintain neighbor clusterhead list. Whenever a node receives packets from a clusterhead, it updates/refreshes the clusterhead list. At the same time, it checks the number of alive clusterheads. The state of a non-clusterhead node is determined by the number of clusterheads in the list. When the number of clusterhead of a non-clusterhead node goes to 0, the node transits to INITIAL state. If the state of node is CLUSTERHEAD when the node receives a packet from another clusterhead, it goes into ORDINARY_NODE state.
5. Every node collects the neighbor information as the clustering procedure goes. It stores neighbors' id, state, and the idle time – if the idle time goes beyond the timeout threshold, the entry is removed.
6. Without employing explicit timer, a node examines the freshness of lists (clusterhead, and neighbor lists) whenever CSL is up – sending/receiving MAC packets.

The end result of the passive clustering with FDW algorithm is that the neighbors of the first node to transmit (in AODV, the neighbor nodes of the first source which starts a route request flooding for a far away destination) declare themselves as clusterhead(s). The first source becomes the root of a "clustering tree" in which the first layer of children are clusterhead(s), and then gateways with possible ordinary node siblings, then clusterhead(s), and so on. Note that clusterheads can be the only members of odd numbered layer on that tree.

We implemented a version of passive clustering with FDW rule in GLOMOSIM version 1.2.2 environment. The implementation was tested on CSMA and 802.11. It worked with both proactive and reactive routings (e.g. AODV). No modification of network layer protocol is necessary. If there are enough pairs communicating, FDW rule converges quickly.

The advantages of this approach can be summarized as

1. Clustering can be achieved without using protocol specific, explicit control packets or signals.
2. Passive clustering does not require the initial clustering phase to precede the data and communication phase.
3. Passive clustering does not require re-clustering to satisfy clustering regulation like LID, when the connectivity changes, because of mobility.
4. Clustering can be done without collecting complete neighborhood information.

5. ON-DEMAND ROUTING IMPROVEMENTS

In this section, we introduce a variant of Ad Hoc On-Demand Distance Vector (AODV) routing which uses passive clustering to reduce the routing overhead and improve performance. By monitoring the MAC traffic, the passive clustering algorithm will form a cluster infrastructure which can effectively control RREQ flooding.

Our AODV modification is summarized as follows:

(1) Each node periodically sends out HELLO messages every 1.5 seconds. The HELLO message is cancelled if the node has sent out a data packet during the last period. Note, every transmission which includes data packets, RREQ, RREP and REER will cause the cancellation of the previously scheduled timer for HELLO messages. The purpose of using the HELLO message option is two-fold. First, it is used to detect link breaks. Secondly, it prevents partial/stale cluster information.

(2) RREQ is forwarded only by clusterheads, gateways and all nodes which are in the initialization. This will drastically reduce the RREQ broadcast since the ordinary nodes do not need to participate in RREQ forwarding.

From the above steps, it is clear that clustering leads to a “reduced” forwarding set suitable for flood search. As long as the ad hoc network is connected, there is a path from an arbitrary source to an arbitrary destination using only gateways and clusterheads as intermediate nodes. Note however, that the set of forwarding nodes is not necessarily minimum (in terms of cardinality). Moreover, this set does not always guarantee the min hop path solution.

6. SIMULATION EXPERIMENTS

The simulation models used for the performance evaluation were implemented in the GloMoSim library [5]. The distributed coordination function (DCF) of IEEE 802.11 [9] is used as the MAC layer in our experiments. Radio propagation range for each node is 150 meters and channel capacity is 2 Mbits/sec. The size of the network is 100 nodes. The simulation area is 1000x1000 meter square. Each simulation executed for 20 minutes of simulation time. Traffic and mobility models are similar to those used in [7].

Traffic sources are CBR (Constant Bit Rate). The source-destination pairs are randomly picked. Data packets are all 512 bytes long. Control packet length is 32 bytes. Control packets are HELLO, RREQ, RREP and RERR. These packets contribute to the routing overhead. The random waypoint model [8] was used in the simulation runs. Aforementioned behavior is repeated for the duration of the simulation.

We compare AODV implemented in the GlomoSim library (version 1.2.3) which does not use either HELLO messages or clustering, with the modified version (AODV/PC) which uses HELLO messages to maintain complete cluster structure in the passive clustering. The goal is to determine the improvements obtained by using passive clustering.

The first set of experiments (Figure 3.A–C) report normalized routing overhead using different number of sources (from 20 to 500) with a moderate packet rate (0.4 packet/sec) and varying node speeds (from 0 to 10 m/sec). Note that the normalized routing overhead is defined as the ratio of routing packets transmitted per data packet delivered to the destination. A low normalized routing overhead is desirable for scalability of the protocols because the actual routing overhead tends to increase linearly with the offered load (i.e. number of source/destination pairs). We notice that AODV/PC reduces the routing overhead except for the situation with very low network load. Figure 3.A is the case that the traffic intervals are usually longer than clustering timeout. With zero mobility, the normalized routing overhead of AODV is almost nil. On the other hand, AODV/PC shows approximately 8 control packets per a data packet delivered because of the HELLO messages - one in 1.5 sec if there is no outgoing traffic. In this case, passive clustering does not contribute anything since there is no flooding for route request. From Figure 3.A, we can see the introduced hello message overhead. As mobility increases, so does the amount of control packets and network traffic. Even in such low network traffic environment, AODV/PC slows down the control overhead increases as mobility increases.

AODV/PC uses less control packets than AODV as the network traffic increases. Figure 3.B is the cross point situation, and Figure 3.C backs up the previous statement. More Hello messages are suppressed as network traffic increases. Moreover, we note that the higher the velocity, the higher the routing overhead reduction induced by clustering. In other words, the more route breakage, the more the gain of AODV/PC. At a speed of 8 m/sec, up to 35% reduction is observed. This behavior is explained by the fact that speed triggers more route breakage for both conventional and clustered AODV. Hence, the superiority of the clustered solution at higher speeds. In Figure 4.A-C, we report the mean end-to-end delay at various offered loads and for different speeds. Unsuccessful deliveries were excluded in delay calculation. The delay is slightly increased for AODV/PC with zero mobility (Figure 4.A) since only clusterheads and gateways participate in the broadcasting of RREQ packets. Due to these restrictions, the hop length of the end to end path can be slightly higher. In the unconstrained case (i.e., no clusters) more direct paths can be found by using also the ordinary nodes (excluded in clustered AODV). As the load and speed increase, however, we know that AODV suffers more routing overhead than AODV/PC. Thus, queueing delays are much higher and tend to offset the shorter path advantage. Furthermore, in the mobile environment, control overhead is indeed the one of main factor causing communication delay. In Figure 4.C, delay decreased after 700 kbit/sec in AODV which can be explained by saturated throughput (Figure 5.C).

Figure 5.A-C show the throughput versus the offered load. In this experiment, control overhead reduction was directly related to throughput increases. In stationary case, throughput increases is almost linear to the offered load in both AODV and AODV/PC. The Hello messages do not decrease the throughput because of the adaptive nature of proposed modification, i.e. we use Hello when there is no outgoing traffic–refer figure 4.A. In mobile cases,

AODV/PC reduces control overhead up to 35% compare with a conventional AODV, and in turn, increases throughput.

7. SUMMARY

We proposed a new passive clustering protocol, passive FDW, which is suitable for realistic wireless ad hoc network environment. Passive FDW does not introduce extra communication overhead for clustering, yet provides better, more stable and robust clustering structures than existing expensive clustering schemes can furnish. We introduced a new clusterhead election rule which was also helped in relaxing the quasi-stationary assumption. We have applied for the first time a clustering technique to on-demand routing, for the purpose of improving its performance. We have shown that the AODV scheme benefits from clustering, especially as mobility increases. A 35% reduction in routing control overhead was shown in a 100 node with 500 source mobile (8m/sec) network, indicating that a comparable throughput improvement can be achieved using clustering.

REFERENCES

[1] Lin, C.R. and Gerla, M. , *Adaptive Clustering for Mobile Wireless Networks*, IEEE Journal on Selected Areas in Communications, Vol. 15, No. 7, Sep. 1997, pp.1265-1275.

[2] T.J. Kwon and M. Gerla, "*Clustering with Power Control"*, In Proceedings of IEEE MILCOM'99, Atlantic City, NJ, Nov. 1999

[3] Chiang, C.-C., Gerla, M. and Zhang, L. ,"*Forwarding Group Multicast Protocol (FGMP) for Multihop, Mobile Wireless Networks*" ACM-Baltzer Journal of Cluster Computing: Special Issue on Mobile Computing, *vol. 1, no. 2, 1998.*

[4] Gerla, M and Tsai, J., Multicluster, mobile, multimedia radio network, ACM-Baltzer Journal of Wireless Networks, Vol.1, No.3, pp.255-265(1995)

[5] M. Takai, L. Bajaj, R, Ahuja, R. Bagrodia and M. Gerla, "*GloMoSim: A Scalable Network Simulation Environment,*" Technical report 990027, UCLA, Computer Science Department, 1999.

[6] A.Ephremides, J.E. Wieselthier and D.J. Baker, *A design concept for reliable mobile radio networks with frequency hopping signaling*, Proc. IEEE 75(1) (1987), pp.56-73

[7] S.R. Das, C.E. Perkins and E. M. Royer, "*Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks*", In Proceedings of IEEE INFOCOM 2000, Tel Aviv, Israel, Mar. 2000.

[8] 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,*" In Proceedings of ACM/IEEE MOBICOM'98, Dallas, TX, Oct. 1998, pp. 85-97

[9] IEEE Computer Society LAN MAN Standards Committee, "*Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification"*, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.

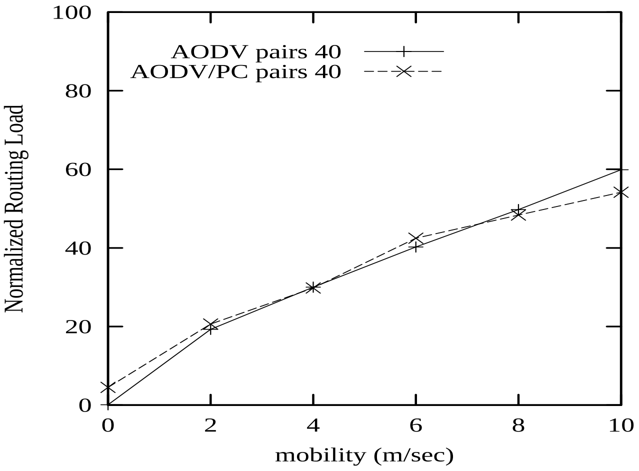

Figure 3.A.

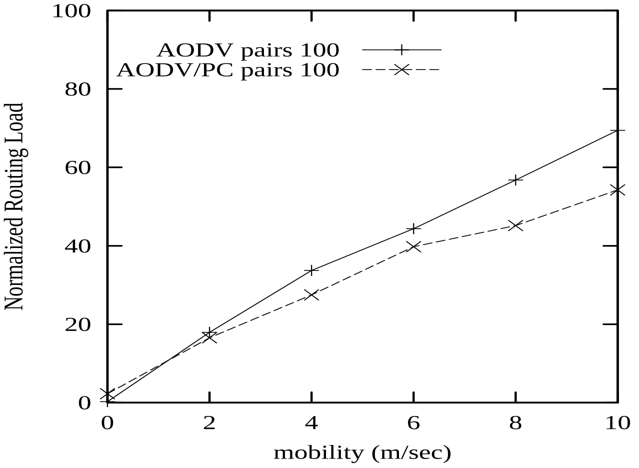

Figure 3.B.

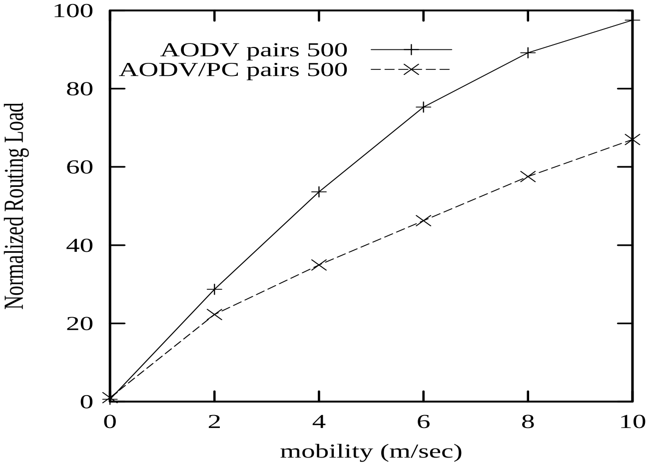

Figure 3.C.

Figure 3. Normalized routing overhead vs speed

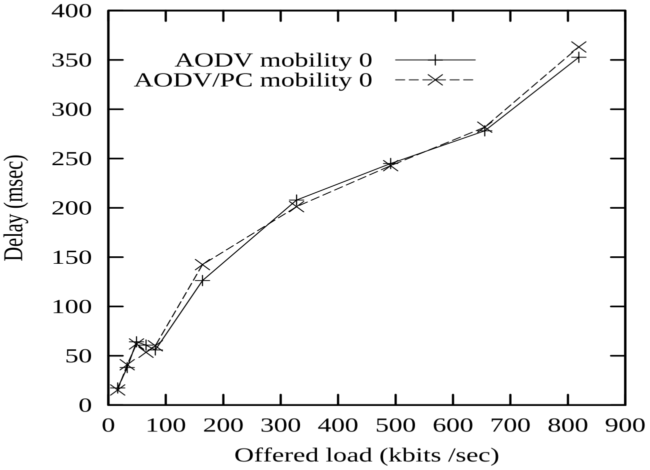

Figure 4.A.

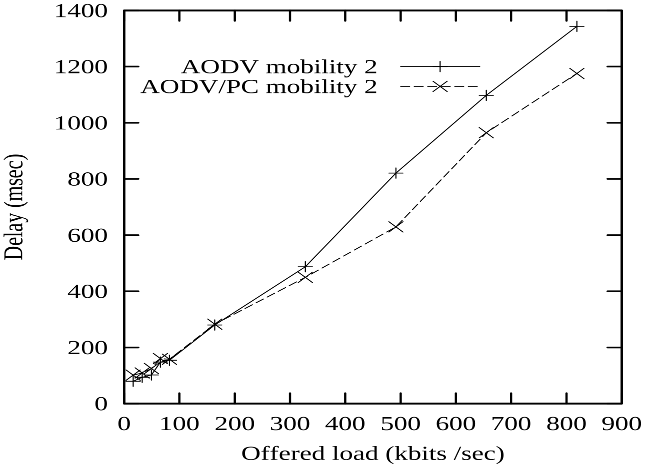

Figure 4.B.

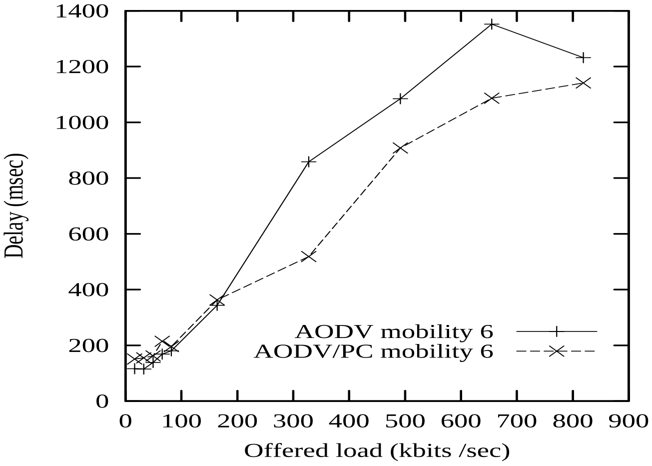

Figure 4.C.

Figure 4. Mean End-to-end Delay vs Offered Load

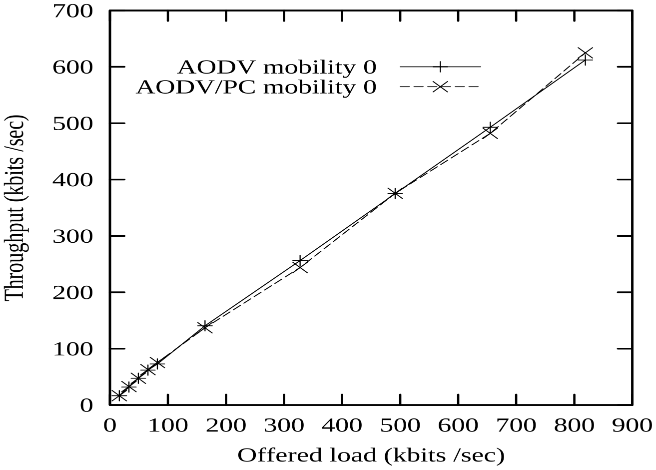

Figure 5.A.

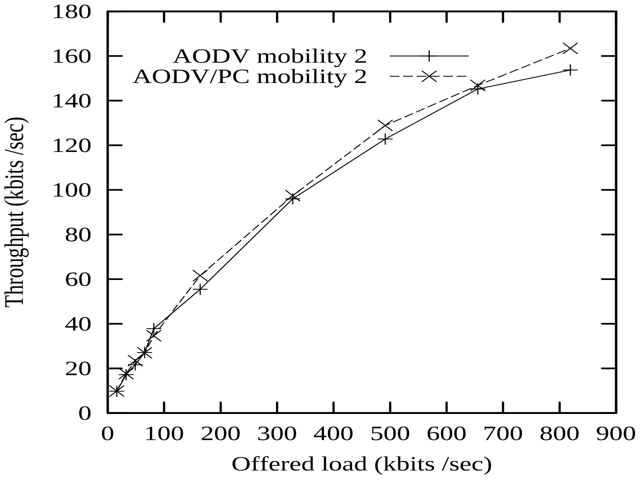

Figure 5.B.

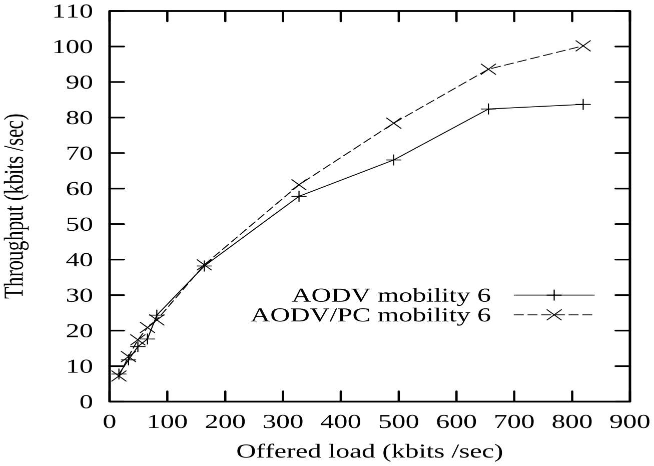

Figure 5.C.

Figure 6. Throughput vs Offered Load