Enhancing the Route Discovery Process of On-Demand Routing in Networks with Directional Antennas*

Marco Aurélio Spohn and Jose Joaquin Garcia-Luna-Aceves

University of California at Santa Cruz
Computer Science Department
Santa Cruz, CA 95064
{maspohn,jj}@cse.ucsc.edu

Abstract. The route discovery process of on-demand routing protocols is based on *route request* (RREQ) packets that are propagated using either an unrestricted broadcast (flooding) or an expanding ring search. We apply two enhancements to this basic process aimed at reducing the overhead incurred in the dissemination of *route request* (RREQs). First, we apply the Enhanced Dominant Pruning (EDP), which is a distributed connected dominating-set algorithm used for reducing the number of nodes that need to propagate RREQs transmitted on broadcast mode. Second, information regarding prior routes to a destination is used to unicast RREQs to a region close to the intended destination. Directional antennas are assumed, which provide higher spatial reuse than omni-directional antennas for unicast transmission. We show the benefits of our approach using the *Ad-hoc On-demand Distance Vector* (AODV) protocol as an example. Detailed simulation results show that using EDP and unicast RREQs provide substantial performance improvements over standard AODV running in ad hoc networks with directional antennas.

1 Introduction

Route discovery in on-demand routing protocols is based on *route request* (RREQ) and *route reply* (RREP) messages (e.g., AODV [1] and DSR [2]). A request is relayed until it reaches a node with a valid route to the destination or the destination itself, which triggers a reply message sent back to the originator. RREQs are propagated using either an unrestricted broadcast or an expanding ring search [3]. In either case, the resulting flooding operation causes considerable collisions of packets in wireless networks using contention-based channel access.

There are several techniques aimed at reducing the impact of broadcasting[4]. *Dominating sets* (DS) play an important role in some of these techniques. A DS is a set of nodes such that every node in the network is either in the set

---

* This work was supported in part by CNPq (Brazil), and the Baskin Chair of Computer Engineering

or is the neighbor of a node in the set. If the graph induced by the nodes in the dominating set is connected, then we have a *connected dominating set*. The problem of determining the *minimum connected dominating set* (MCDS) is known to be NP-complete [5]. Extensive work has been done on finding a good approximation of MCDS in terms of small approximation ratio. A protocol with a constant approximation ratio of eight has been proposed by Wan et. al. [6]. However, their approach requires that a spanning tree be constructed first in order to select the dominating nodes (forwarders), and only after that a broadcast can be performed. To improve the route discovery process we need an approach that is suitable for dynamic networks with mobile nodes, and is based on determining the CDS *on-the-fly*.

Lim and Kim [7] show that the MCDS problem can be reduced to the problem of building a *minimum cost flooding tree* (MCFT). Given that an optimal solution for the MCFT problem is not feasible, they propose heuristics for flooding trees, resulting in two algorithms: *self-pruning* and *dominant pruning* (DP). They show that both algorithms perform better than *blind flooding*, with which each node broadcasts a packet to its neighbors whenever it receives the packet along the shortest path from the source node, and that DP outperforms *self-pruning*.

A few enhancements to dominant pruning have been reported recently [8, 9]. Lou and Wu [8] propose two enhancements to DP: *total dominant pruning* (TDP), and *partial dominant pruning* (PDP). Spohn and Garcia-Luna-Aceves [9] propose *enhanced dominant pruning* (EDP), which is applied to AODV (using omni-directional antennas) to show its improvements compared to DP.

We apply two enhancements to the route discovery process of on-demand routing protocols aimed at reducing the overhead incurred in the dissemination of RREQs. First, we apply EDP to reduce the number of nodes that need to propagate RREQs transmitted on broadcast mode. Second, information regarding prior routes to a destination is used to unicast RREQs to a region close to the intended destination, so that broadcast RREQs are postponed as much as possible and occur only close to the destination, rather than on a network-wide basis.

Directional antennas are assumed, which provide higher spatial reuse [10] [11] than omni-directional antennas for unicast transmission. An advantage of using directional antennas is that they allow a larger number of simultaneous transmissions compared to omni-directional antennas .

The rest of this paper is organized as follows. Section 2 presents a review of dominant pruning and its enhancements. Section 3 presents our approach for improving the route discovery process in AODV. Section 4 presents detailed simulations showing the benefits of EDP when it is applied to AODV and compares it against AODV, and AODV with DP. In all cases we use directional antennas. The simulation results clearly show that AODV with EDP renders the best performance of all the AODV versions, which is a direct consequence of reducing packet collisions due to RREQs by means of EDP and the use of directional antennas. Section 5 concludes this work.

2   Dominant Pruning: Review and Enhancements

In DP [7] the sending node decides which adjacent nodes should relay the packet. The relaying nodes are selected using a distributed CDS algorithm, and the identifiers (IDs) of the selected nodes are piggybacked in the packet as the forwarder list. A receiving node that is requested to forward the packet again determines the forwarder list. The flooding ends when there is no more relaying nodes.

Nodes keep information about their two-hop neighborhood, which can be obtained by the nodes exchanging their adjacent node list with their neighbors. DP is a distributed algorithm that determines a set cover based on the partial knowledge of the two-hop neighborhood. Ideally, the number of forwarding nodes should be minimized to decrease the number of transmissions. However, the optimal solution is NP-complete and requires that nodes know the entire topology of the network. DP uses the *greedy set cover* (GSC) algorithm to determine the forwarder list of a packet (i.e., the list of nodes that should forward the packet) based just on partial knowledge of the network topology. GSC recursively chooses one-hop neighbors that cover the most two-hop neighbors, repeating the process until all two-hop neighbors are covered.

TDP [8] requires that the two-hop neighborhood of the sender be piggybacked in the header of the packet. This information reduces the size of the two-hop neighbor set that needs to be covered by the forwarders. The header size increases proportionally to the number of nodes in the two-hop neighborhood, which may become a problem in dense networks. PDP [8] enhances DP by eliminating the two-hop nodes advertised by a neighbor shared by both the sender and the receiver (forwarder). Simulation results assuming an ideal MAC layer with which no contention or collisions occur show that both TDP and PDP improve DP in a static environment. A dynamic scenario is also evaluated, and DP is shown to perform better than both TDP and PDP.

EDP [9] requires the second-to-previous (STP) list in addition to the forwarder list. EDP improves DP by reducing even more the number of redundant broadcasts, mainly due to the information provided by the STP list. Although EDP increases the packet header size with the STP list, EDP improves the performance of AODV when compared to standard DP.

3   Improvements to the Route Discovery Process of AODV

This section addresses the application of EDP to the route discovery process in AODV. The resulting variant of AODV is denoted by AODV-EDP. Our neighbor protocol uses hello packets to disseminate the one-hop neighborhood, which creates a picture of its two-hop neighborhood at any given node in the network.

To avoid pruning too many route requests in the presence of mobility and cross-traffic, we have chosen to implement the neighbor protocol as part of AODV. We extended the hello mechanism available in AODV to include the information about the one-hop neighborhood in hello messages, and we also rely

on the AODV mechanisms for evaluating the link status to neighbors. A hello packet advertises the node's sequence number (*mySeqNum*), the identification of its known neighbors (*neighbors[]*), and the corresponding neighbors' sequence number (*neighSeqNum[]*). We have chosen a hello interval of $1.5s$. To reduce the number of broadcast messages, a RREQ also advertises the one-hop neighborhood information, and the transmission of a RREQ by a node reschedules its pending hello message.

Algorithm 1: RREQ algorithm

Data   : n_i, destination D, $\mathcal{F}_S$, $\mathcal{F}_{S_S}$, U_i
Result   : Unicast the RREQ, or Broadcast the RREQ
begin

1   if *recently expired route to D and not retrying* then

2 | $NextHop \longleftarrow previous\_nextHop(D)$
3 | if *validRoute(NextHop)* then

4 | $\llcorner$ $result \longleftarrow Unicast$
5 | else

6 | $\llcorner$ $result \longleftarrow Broadcast$

7   else

8 | $\llcorner$ $result \longleftarrow Broadcast$
9    $\mathcal{F}_i \longleftarrow EDP(n_i, \mathcal{F}_S, \mathcal{F}_{S_S}, U_i)$
10   Update RREQ packet with $\mathcal{F}_i$
11   if *result == Unicast* then

12 | $\llcorner$ Unicast the RREQ packet to *NextHop*
13   else

14 | $\llcorner$ Broadcast the RREQ packet

end

Algorithm 1 presents the pseudo-code for the modified RREQ. As defined in [9], $\mathcal{F}_S$, $\mathcal{F}_{S_S}$, and U_i, are respectively the forwarder list, the STP list, and the set of neighbors to be covered. A route request (RREQ) is handled as follows:

– If the source of a RREQ does not have any previous knowledge about the route to the destination or is retrying the RREQ, it calculates its forwarder list using EDP, and broadcasts the packet (Lines 8, 9, and 14).
– On the other hand, if the source of a RREQ has knowledge about a recently expired route to the destination, and there is a valid route to the next hop towards the destination (Lines 2, 3, and 4), the node calculates the forwarder list using EDP (Line 9), but instead of broadcasting the RREQ packet, the

node unicasts the packet to the last known next hop towards the destination (Line 12).

- Upon receiving a route request, a forwarder that cannot respond to this request calculates its own forwarder list using the information provided in the RREQ packet (i.e., forwarder list, second to previous forwarder list, and source node) and broadcasts or unicasts the packet (depending on which one of the two first cases apply) after updating it with its own forwarder list.

Eventually, the RREQ reaches a node with a route to the destination or the destination itself. Our approach attempts to reduce the number of collisions and the delay of the route discovery by unicasting a RREQ towards the region where the destination was previously located. The success of this approach depends on how fresh the previous known route to the destination is, and how fast the destination node is moving out of the previous known location. If an intermediate node has completely removed any route to the destination, the RREQ is then broadcasted. The intended effect is to postpone the broadcast of a RREQ to the region closest to the destination. In the case that the unicast approach fails, or there is no previous route to the destination, the source broadcasts by default.

Because of topology changes, nodes may not have correct two-hop neighborhood information, which may result in forwarding lists that do not cover all nodes in the neighborhood. However, this is not a major problem when the request is broadcasted, because a node incorrectly excluded from the forwarder list may also receive the request and is able to respond in the case it has a route to the destination.

4   Simulations and Performance Results

To compare AODV-EDP against other protocols, we use traffic and mobility models similar to those previously reported in [12]. We implemented AODV-EDP in *Qualnet* 3.5, and compare it against AODV-DP (AODV with *Dominant Pruning*), and standard AODV with no hello messages. In the simulation scenarios, we vary both the number of nodes in the network as well as the number of flows.

Qualnet provides two models for directional antennas: *switched beam* with multiple patterns (circular array with 8 patterns), and *steerable* with multiple steerable patterns (triangular array with 4 different beam widths). The antenna model is receiver side only due to the omni-directional MAC protocol. In our simulations we have used the *switched beam* model for all the simulations and routing protocols. The radio model used is a $2Mbps$ IEEE 802.11 device. Terrain size and radio range are adjusted for each particular scenario.

Traffic sources are continuous bit rate (CBR). Only 512-bytes data packets are used. The source-destination pairs are chosen randomly among the nodes in the network. Flows last in average for $50s$ (following an exponential distribution), unless otherwise mentioned. Source nodes keep active flows during all simulation time (new destinations are randomly selected as needed). Nodes begin transmitting at $50s$ plus an offset uniformly chosen over a $5s$ period to avoid

synchronization in their initial transmissions. The simulation time is set to 600 seconds, and identical mobility and traffic scenarios are used for all protocols. Initially nodes are placed uniformly over a grid. Nodes move according to the random waypoint model with velocities between 0 and $20m/s$. Six pause times are tested: $0s$ (always moving), $50s$, $100s$, $300s$, $400s$, and $600s$.

Experiments are repeated for 10 trials with different random-number seeds. Results present a 95% confidence interval. Each data point represents the mean over the 10 runs discarding the lowest and largest results (quantile of one).

Four performance metrics are evaluated:

- *Packet delivery ratio*, the ratio of the data packets delivered to the destination to those generated by the CBR sources.
- *Average end-to-end delay* for data packets, including all possible delays caused by route discovery latency, queuing at the interface, retransmission delays at the MAC layer, and propagation and transfer times.
- *Routing load*, the number of routing packets transmitted per data packet delivered to the destination, where each hop traversed by the packet is counted as one transmission.
- *MAC collisions*, the number of collisions detected at the MAC layer.

Because the scenarios we have used to evaluate our approach differ from those presented in [8], and because we implemented our solution together with a neighbor and routing protocol, we do not know how our solution compares to TDP and PDP. The relation between the savings of pruning (too much, or too little) and the degree of broadcast redundancy achieved, can be different, depending on the physical environment under consideration. If we take into account that more packets being broadcasted translate into more contention and collisions, we could have a different picture, depending on the number of broadcasts that are avoided.

4.1   50-Node Scenario

The network is composed of 50 nodes spread over an area of $1500m$ x $300m$. The radio has a nominal transmission range of $250m$. The network is tested for three traffic models:

- 30 source nodes transmitting 4 *packets/s*, each flow lasting in average $50s$ (exponential distribution).
- 40 source nodes transmitting 3 *packets/s* (flows of $50s$ as in the previous scenario).
- 30 source nodes transmitting 4 *packets/s*, with very short flows (flows lasting in average $10s$ and $20s$).

In both scenarios, we have a total of 120 data packets being injected into the network every second. We show that, in all of the categories, AODV-EDP outperforms the other protocols. Figures 1 and 2 summarize the results for 30 and 40 flows, and Figures 3 and 4 summarize the results for 30 sources varying the flow duration.

30 flows. As pointed out in [12], the possibility of link failures is low with low mobility, but due to the node movement model (*random waypoint*) nodes usually get clustered. This situation is responsible for congestion in those regions in the presence of high traffic. This causes the link layer to report link failures even though the nodes are relatively static and a physical link still exists between the nodes. This is observed on Figure 1(a), where we notice a decreasing on the packet delivery ratio for some larger pause times.

Figure 1(b) shows the average end-to-end delay for 30 flows. AODV-EDP presents an almost constant mean latency, and is always the best for all pause times. Together with the packet delivery ratio, these results show that besides delivering more packets for most of the pause times, AODV-EDP delivers them faster than the other protocols. AODV-DP again shows that DP alone improves AODV. Clustering of nodes has a direct impact on the latency as well. Packets spend more time waiting on the queues, and usually need to be retransmitted due to increased congestion.

Figure 2(a) presents the routing load for 30 flows. As expected, AODV-EDP has a lower routing load compared to standard AODV, because it reduces the number of broadcast transmissions. AODV-DP reduces the control overhead compared to AODV, but not as much as AODV-EDP.

Figure 2(b) shows the number of collisions at the MAC layer for 30 flows. The number of collisions for standard AODV is noticeable larger than the other protocols, because a node always respond to the first received RREQ (if the TTL is valid, i.e., greater than zero). Because both AODV-EDP and AODV-DP reduce the number of necessary broadcasts, it translates in less collisions.
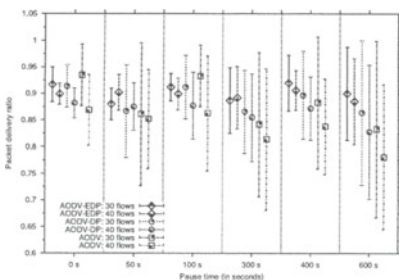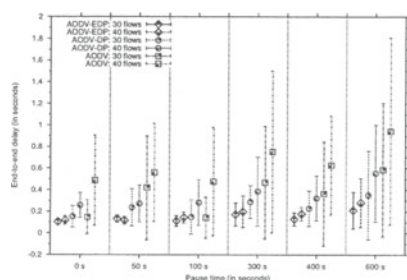
(a) Packet Delivery Ratio   (b) End-to-End Delay

Fig. 1. 50 Nodes, 30 and 40 flows: packet delivery ratio and end-to-end delay

40 flows. In this scenario we increase the number of flows but keep the same number of data packets being injected into the network (each source sends 3 packets/s). Figure 1(a) shows the packet delivery ratio. AODV-EDP presents
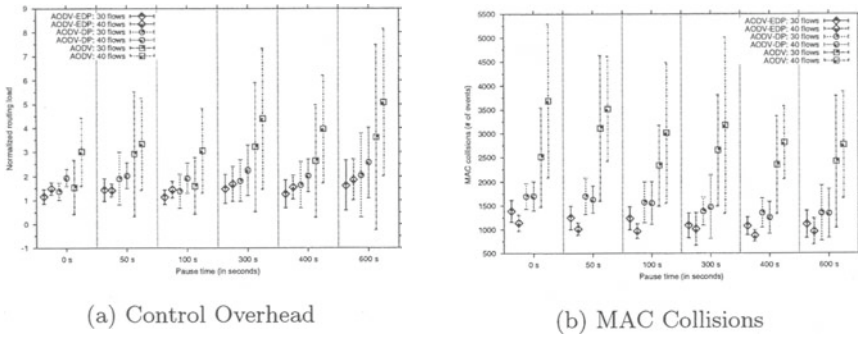
(a) Control Overhead

(b) MAC Collisions

Fig. 2. 50 Nodes, 30 and 40 flows: control overhead and MAC collisions

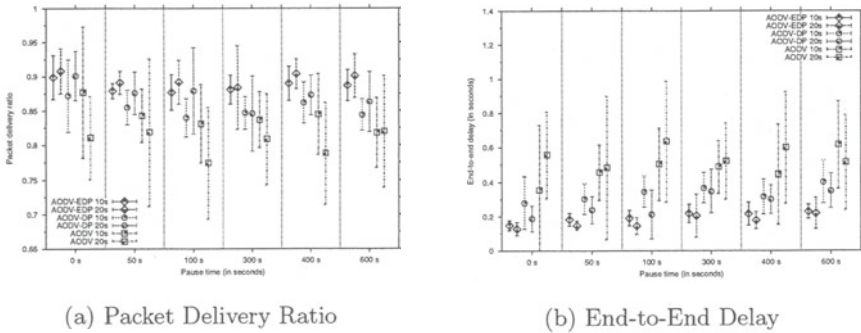(a) Packet Delivery Ratio

(b) End-to-End Delay

Fig. 3. 50 Nodes, flows of 10s and 20s: packet delivery ratio and end-to-end delay

an almost constant packet delivery ratio for all pause times, and it has a higher delivery ratio for all the pause times. The effect of clustering is noticeable on Figure 1(a). This result shows that by increasing the number of flows, more nodes in the network participate in active communications, what translates in more replies coming from intermediate nodes during the route discovery process. In these circumstances, it helps even more when a request can be unicasted instead of broadcasted.

Figure 1(b) shows the average end-to-end delay for 40 flows. AODV-EDP presents an almost constant mean latency, and is always the best for all pause times. In this scenario AODV-EDP again deliver more packets, and doing it faster than the two other variants. AODV-DP again shows that DP alone improves AODV. For all the pause times, AODV-DP presents less than half the latency produced by AODV. On its turn, AODV-EDP reduces even more the end-to-end delay, having almost all the time half the latency produced by AODV-DP. The impact of clustering of nodes in the latency of data packets is more noticeable only for AODV.
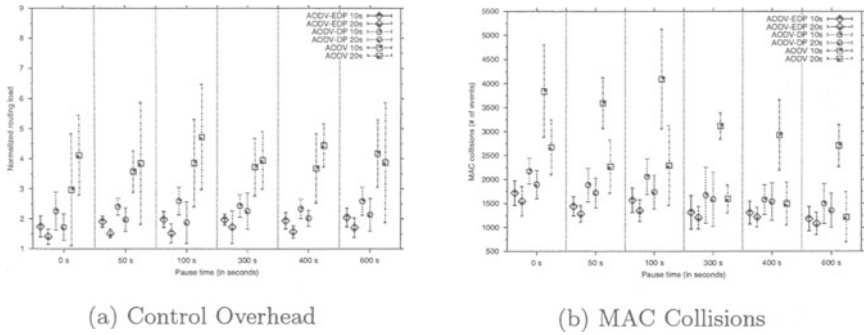
(a) Control Overhead   (b) MAC Collisions

Fig. 4. 50 Nodes, flows of 10s and 20s: control overhead and MAC collisions

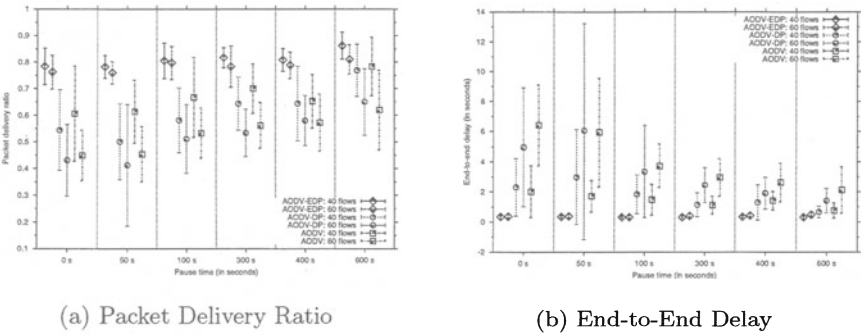(a) Packet Delivery Ratio   (b) End-to-End Delay

Fig. 5. 100 Nodes, 40 and 60 flows: packet delivery ratio and end-to-end delay

Figure 2(a) presents the routing load for 40 flows. As expected, there is an increase in the routing load because there are more flows (and destinations) in the network. It is more noticeable the improvements introduced by both dominant pruning techniques, but AODV-EDP performs better for all pause times.

Figure 2(b) shows the number of collisions at the MAC layer for 40 flows. Although a larger number of flows, for both AODV-DP and AODV-EDP we notice only a slightly difference (sometimes even less collisions) compared to the 30 flows scenario. But AODV incurs on more collisions than on the previous scenario. In all situations AODV-EDP outperforms the two other variants.

30 sources: varying flow duration. In this set of simulations we play with the flow duration. At any given time, there are at least 30 active flows, and every node in the network has a chance to be the source of at least one session . In fact, because we are dealing with flows of short duration, every node participates as a sender and as a receiver on several different sessions during the simulation time. Flows last in average 10s and 20s (exponential distribution). As mentioned
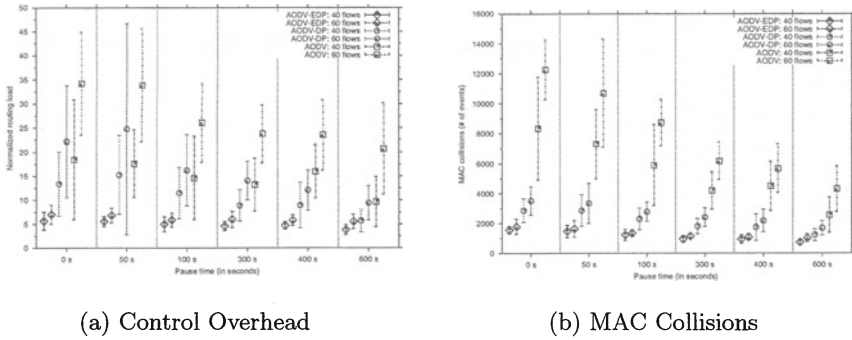
(a) Control Overhead   (b) MAC Collisions

Fig. 6. 100 Nodes, 40 and 60 flows: control overhead and MAC collisions

before, flows start at $50s$ of simulation time with a jitter of $5s$. For each flow duration, simulations are run for the same number of trials as in the previous scenarios.

The results presented in Figures 3 and 4 show that DP alone improves the performance of AODV for all pause times and for all flows. But AODV-EDP performs better than the other two protocols in all situations, and it also presents the smallest variance among the three protocols. Both AODV-DP and AODV-EDP present an almost constant performance for all pause times. As expected, we notice again a great reduction on the control overhead due to the pruning of redundant broadcasts. But we also notice that AODV performs as well as the other protocols regarding number of collisions in situations with large pause times and flows of $20s$.

4.2   100-Node Scenario

The network is composed of 100 nodes spread over an area of $2200m$ x $600m$. The radio has a nominal transmission range of $280m$. For traffic sources, we have two traffic models: 40 source nodes transmitting 3 $packets/s$, and 60 source nodes transmitting 2 $packets/s$. In both cases we have a total of 120 data packets being injected into the network every second. We show that, in most of the categories, AODV-EDP outperforms the other protocols. For all the metrics evaluated, AODV-EDP presents the smallest variance.

40 flows. Figure 5(a) shows the packet delivery ratio for 40 flows. AODV-EDP presents an almost constant packet delivery ratio for all pause times, as well as a higher delivery ratio for all pause times. AODV-DP performs worse than AODV specially in the high mobility scenarios, as the network gets more static the difference between AODV and AODV-DP becomes very small.

Figure 5(b) shows the average end-to-end delay for 40 flows. AODV-EDP presents an almost constant mean latency, and is always the best for all pause

times. Together with the packet delivery ratio, these results show that besides delivering more packets, AODV-EDP delivers them faster than the other protocols. Although AODV-DP performs better than AODV, AODV-DP delivers less packets than AODV.

Figure 6(a) presents the routing load for 40 flows. As expected, AODV-EDP has a lower routing load in comparison to standard AODV, but the difference among the protocols is a way larger than in the 50 nodes scenario. AODV-DP reduces the control overhead compared to AODV, but not as much as AODV-EDP. AODV-DP shows that DP alone improves the control overhead, but it does not improve as much as EDP.

Figure 6(b) shows the number of collisions at the MAC layer for 40 flows. As expected, the number of collisions for standard AODV is noticeable larger than the other protocols. AODV-EDP incurs 4 to 5 times less collisions than AODV for most of the pause times, and almost half of the collisions incurred by AODV-DP. AODV-EDP also presents the smallest variance, and an almost constant number of collisions for all pause times.

60 flows. Figure 5(a) shows the packet delivery ratio for 60 flows. AODV-EDP presents an almost constant packet delivery ratio for all pause times, as well as a higher delivery ratio. AODV-DP performs worse than AODV but the difference is smaller compared to the 40 flows scenario.

Figure 5(b) shows the average end-to-end delay for 60 flows. AODV-EDP presents an almost constant mean latency, and is always the best for all pause times. As in the previous scenarios, besides delivering more packets, AODV-EDP delivers them faster. AODV-DP performs better than AODV, but it also delivers slightly less packets than AODV. As expected, the latency increases compared to the 40 flows scenario, but not as much for AODV-EDP.

Figure 6(a) presents the routing load for 60 flows. AODV-DP performs better than AODV, specially for larger pause times. AODV-EDP is the best again, and when comparing the results against the 40 flows scenario, we observe that only AODV-EDP does not increase the control overhead proportionally as observed in the two other protocols.

Figure 6(b) shows the number of collisions at the MAC layer for 60 flows. Compared to the 40 flows scenario, AODV increases about 50% the number of collisions, while both AODV-EDP and AODV-DP increase around 15% the number of collisions. Both pruning techniques show to be effective on reducing redundant broadcasts, but EDP outperforms DP in all aspects.

5   Conclusions

We applied two enhancements to the route discovery process of on demand routing protocols. Redundant broadcasts increase the number of packet collisions, and consequently delay the response for RREQ's in the route discovery process. EDP is shown to reduce the number of broadcast transmissions when compared to standard DP. To reduce the interference (and the number of packet collisions)

we take advantage of directional antennas by unicasting RREQs to regions of the network where the destination may be located.

We show through extensive simulation results that AODV-EDP improves the performance in all aspects (i.e., the four metrics chosen) for all the pause times in the 50-node and the 100-node scenarios. The other protocols (standard AODV and AODV-DP) deliver fewer packets than AODV-EDP. AODV-EDP not only delivers more packets, but it does it faster than the other protocols. AODV-EDP also presents the smallest variance among the protocols, and almost constant results for all the metrics considered in the simulations (with some exceptions because of clustering of nodes due to the mobility model).

References

1. Perkins, C.: Ad-hoc on-demand distance vector routing. In: Second IEEE Workshop on Mobile Computing Systems and Applications. (1999)
2. Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks. In Imielinski, Korth, eds.: Mobile Computing. Volume 353. Kluwer Academic Publishers (1996)
3. Segall, A.: Distributed network protocols. IEEE Transactions on Information Theory **29** (1983) 23–35
4. Williams, B., Camp, T.: Comparison of broadcasting techniques for mobile ad hoc networks. In: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC). (2002) 194–205
5. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman, San Francisco, (1978)
6. Wan, P.J., Alzoubi, K.M., Frieder, O.: Distributed construction of connected dominating set in wireless ad hoc networks. In: Proceedings of INFOCOM. (2002)
7. H.Lim, Kim, C.: Flooding in wireless ad hoc networks. Computer Communications **24** (2001)
8. Lou, W., Wu, J.: On reducing broadcast redundancy in ad hoc wireless networks. IEEE Transactions on Mobile Computing **1** (2002)
9. Spohn, M.A., Garcia-Luna-Aceves, J.J.: Enhanced dominant pruning applied to the route discovery process of on-demand routing protocols. In: Proceedings of the 12th International Conference on Computer Communications and Networks (ICCCN). (2003)
10. Huang, Z., Shen, C.C.: A Comparison Study of Omnidirectional and Directional MAC Protocols for Ad Hoc Networks. In: IEEE Globecom 2002, Taipei, Taiwan (2002)
11. Wang, Y., Garcia-Luna-Aceves, J.J.: Collision Avoidance in Single-Channel Ad Hoc Networks Using Directional Antennas. In: Proceedings of ICDCS, Providence, Rhode Island, U.S.A. (2003)
12. Perkins, C., Royer, E., Das, S.R., Marina, M.K.: Performance comparison of two on-demand routing protocols for ad hoc networks. IEEE Personal Communications **8** (2001) 16–28