Making On-demand Routing Protocols Based on Destination Sequence Numbers Robust

Hari Rangarajan*
Email: hari@cse.ucsc.edu
* Computer Engineering Department
University of California at Santa Cruz
Santa Cruz, CA 95064.

J.J. Garcia-Luna-Aceves*†
Email: jj@cse.ucsc.edu
†Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304.

Abstract— **We show that the way in which the Ad-hoc On-demand Distance Vector (AODV) protocol handles destination-based sequence numbers can lead to looping of data packets, de-facto network partitions, and counting to infinity in the presence of link or node failures in ad hoc networks using an unreliable medium access control (MAC) protocol like the IEEE 802.11 DCF. The source of AODV's problems with sequence numbers is the use of a delete period after which nodes are allowed to forget invalid routes to destinations. We present a new approach for the handling of sequence numbers in AODV that eliminates the use of delete periods for destination-based sequence numbers, and show with simulation experiments that the new approach performs the same or better than AODV.**

I. Introduction

Considerable attention has been paid on developing loop-free protocols for MANETs. However, the robustness and efficiency of such protocols in the presence of failures and other abrupt changes in MANETs with unreliable MAC protocols, and nodes that must delete old routing information, has not been addressed in detail.

The current proposals in the MANET Working Group of the IETF (Internet Engineering Task Force) include the Optimized Link State Routing (OLSR) Protocol [5], the Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) protocol [8], the Dynamic Source Routing (DSR) protocol [7], and the Ad-hoc On-demand Distance Vector (AODV) [10]. These protocols are meant to operate in MANETs in which the MAC protocol is unreliable, like the IEEE 802.11 distributed coordination function (DCF), and nodes are allowed to delete old routing-table entries.

OLSR and TBRPF maintain routing information proactively and use mechanisms to maintain link-state information correctly that are similar to those used in routing schemes for wired networks. In this paper, we focus on on-demand routing schemes, which can incur less overhead in large networks in which each node needs to communicate with a small percentage of the rest of the nodes.

DSR uses source routed packets to avoid looping; when source routes become invalid, the data packets carrying such routes are dropped and route errors are sent to the sources of packets. Intuitively, as source routes become obsolete more quickly due to network dynamics, the performance of source routing degrades. To address this problem, we have proposed [9], an approach based on path information that does not require source-routed packets and is more robust than DSR as a result.

This work was funded in part by the Baskin Chair of Computer Engineering at UCSC and by the National Science Foundation under Grant CNS-0435522.

AODV uses destination-based sequence numbers to maintain loop-free routes on demand, while allowing packet forwarding to be based solely on the packet destination. The sequence number carried in a route request (RREQ) elicits only fresher route replies (RREP) with an equal or higher sequence number. On a link failure, a node increases its sequence number for a destination and invalidates the route. Route errors (RERR) are sent unreliably, based on the notion that increasing the destination sequence number invalidates the route entry of all upstream nodes. A few proposals exist that improve on AODV in terms of reducing the frequency with which the destination must be the node that answers a RREQ [6], [2].

Bhargavan et al [1] identified a failure condition in AODV caused by the rebooting of a node, and proposed the the use of the $DELETE\_PERIOD$ as a safety condition for loop-freedom in AODV in the presence of node reboots. The $DELETE\_PERIOD$ is the maximum time that a node can retain its successor for a route entry in the presence of unreliable communication, and is a property of the prevailing network conditions. The AODV specification uses this timer for managing the state of the routing table related to destination sequence numbers. Assuming that a node that reboots waits "long enough" before re-engaging in normal update activity, as prescribed by Bhargavan et al [1], the use of destination-based sequence numbers in AODV works correctly, as long as nodes maintain the last sequence number they learned for a given destination. However, to make the scheme practical in large MANETs, nodes must be allowed to delete old invalid routes after a finite time. This constitutes the motivation for the work presented in this paper, which provides the following contributions:

- Demonstrating that the way in which destination sequence numbers are used in AODV is prone to looping, de-facto network partitions, and counting-to-infinity.
- Introducing a robust solution to on-demand loop-free routing based on destination sequence numbers that elim-

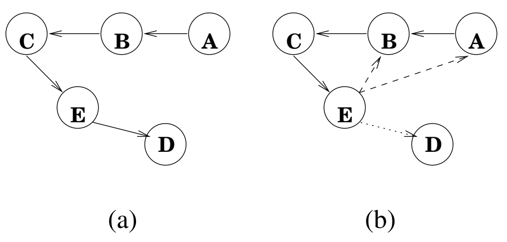

Fig. 1. Issues with $DELETE\_PERIOD$ in AODV

inates the need for using a $DELETE\_PERIOD$.

- Applying our solution to AODV as an example of how robust sequence-number handling can be attained without compromising the efficiency of a protocol.

Section II shows that AODV's handling of sequence numbers based on a $DELETE\_PERIOD$ can lead to looping, de-facto partitions and count-to-infinity behavior. Section III provides the details of our robust approach to destination-based sequence numbering. The basic idea behind the solution is simple: A node that may have "forgotten" the last sequence number it had learned for a destination must request that the destination answer its RREQ; this is the case of nodes without prior knowledge of the destination, nodes that reboot, and nodes that have deleted routing information for a destination. Based on this approach, we present a modification of AODV that we call AODV-RSN (for robust sequence numbering) in this paper. Section IV outlines why AODV-RSN eliminates the problems in AODV outlined in Section II and eliminates the need for using a pre-defined $DELETE\_PERIOD$. Section V shows through simulations that the proposed new approach to the handling of sequence numbers does not impact AODV's performance negatively, and allows nodes to participate in normal routing activities much faster than AODV after reboots. Section VI provides our concluding remarks.

II. Sequence Numbering Problems in AODV

We address AODV's problems with sequence numbering by way of examples in which AODV is assumed to operate on top of an unreliable MAC protocol similar to IEEE802.11 DCF.

In AODV, a node can lose its routing state (destination sequence number) for a destination in two possible ways: when a node reboots or after a node deletes a route entry to save memory. We now discuss the problems associated with losing such a state in AODV.

A. Looping

Figure 1(a) shows an example directed acyclic successor graph for a five-node network running AODV with nodes *A, B, C,* and E having flows to destination D. The nodes in the example have an active route entry with a valid destination sequence number for D, and Figure 1(b) shows the network at a subsequent time with the physical connectivity affected due to mobility. At this time, link $E-D$ goes down, and links $E-A$ and $E-B$ come up. Node E sends a RERR to advertise the unreachability of destination D, which propagates to the upstream nodes along the directed acyclic graph for D.

Consider the case of "unbounded" queueing delays or message loss. In the simplest case, if the RERR is never delivered to C, then node E can send a RREQ for D after deleting its destination sequence number, which happens after waiting for $DELETE\_PERIOD$. Nodes A or B can answer the RREQ resulting in a loop.

The AODV specification defines a value for the $DELETE\_PERIOD$ equal to K times $ACTIVE\_ROUTE\_TIMEOUT$, with the recommended value for K as five. However, the actual value of K is a property of the prevailing network conditions and cannot be deduced accurately. If the parameter 'K' is set too low, then it can result in loops. On the other hand, if 'K' is set too high, it can aggravate the problem during reboots, which we discuss next.

B. De-Facto Partitions

A node rebooting after a failure loses state about all destination sequence numbers. Assume that node E in our example of Fig.1(a) reboots and has to wait for a minimum of $DELETE\_PERIOD$ before it can participate in routing actions. Node E sends RERRs on receiving data packets from C, which propagate to A. New RREQs generated by A for D cannot reach D during E's reboot wait time, because E has to drop all RREQs during that time. Hence, the network appears to be partitioned, despite the physical links available in the network. This limitation can be critical when there are no alternate paths, and can force nodes to choose sub-optimal paths for the duration of their flow, because routes are not improved pro-actively. We note that nodes that are sources or destinations of flows after a reboot are effectively partitioned from the network during the reboot wait period. Considering that MANETs have to be deployed on-demand instantaneously, the reboot wait period forces nodes to stay out for an arbitrary time during network setup, which can be important for many operational scenarios.

C. Counting to Infinity in AODV

The previous two issues were due to the unreliability of the communication medium. However, we show now that the use of the $DELETE\_PERIOD$ impacts the ability of AODV to terminate on a partition or node failure.

According to the AODV RFC [10] and Internet draft [4], AODV deletes invalid route entries after a finite time equal to the maximum elapsed time after which a node can still send data packets to the next-hop specified in the routing table, called the $DELETE\_PERIOD$.

We show in [2], an example of how counting-to-infinity can occur in AODV. The basic problem can be summarized as follows: *A node A along a successor path P to destination D should never delete its invalid route table entry for D before guaranteeing that all its upstream nodes along path P have invalidated their active route entries for D.*

We note that a similar counting-to-infinity scenario can occur when nodes reboot after failures. On reboot, nodes running AODV wait for $DELETE\_PERIOD$ to elapse before engaging in routing operations; however, they forget their last-known sequence number for a destination. Hence, a counting-to-infinity scenario can be constructed that involves having all nodes along an upstream path rebooting.

In practice, counting to infinity in AODV can be avoided by waiting "long enough" before deleting invalid routes or rejoining normal operation after reboots. However, as the network size and its diameter change, what "long enough" means must also change. This is akin to the use of "hold down" timers used in the past trying to avoid the counting-to-infinity problem in the routing information protocol (RIP). Given that internodal coordination spanning multiple hops incurs too much overhead and that very long waiting periods are undesirable for protocol efficiency, a more elegant solution to the counting-to-infinity problem is desirable, which we present in the next section.

III. Robust Sequence Numbering in AODV

We propose a modification to the way in which destination sequence numbers are managed in AODV and call it AODV-RSN (for robust sequence numbering). In a nutshell, in AODV-RSN, the time period $DELETE\_PERIOD$ is eliminated, and a routing-table entry can be deleted without the necessity to wait for any arbitrary time period. Nodes can participate in routing actions immediately after a reboot. A node must request the destination to answer its RREQ if it has no routing-table entry for the destination, which can be due to the node (a) never hearing about the destination before, (b) rebooting, or (c) deleting deleted its routing information for the destination for any reason.

The following changes to the AODV specification are required for realizing AODV-RSN. We utilize the 'unknown sequence number' U-bit of the RREQ. The RREP requires the addition of a new 'Destination Initiated Reply' D-bit (which is borrowed from the reserved 13-bits), and a 32-bit field for carrying the flooding id. The parameters stored in the routing table do not require any modifications.

Initiating RREQs: A node generating a RREQ for a destination must set 'unknown sequence number' U-bit if it possess no valid sequence number for the destination in its routing table.

Relaying RREQs: A node on receiving a RREQ must relay the RREQ with the U-bit set in either of the following cases: (i) relaying node possesses no valid sequence number for the destination in its routing table, or (ii) the RREQ was received with the 'U-bit' set. For this destination, in-addition to caching the (source, flooding id) pair for RREQs, the address of the node, $reversehop$, that sent the RREQ, and a boolean value P which is set to true by default, must be cached.

Generating RREPs: A node having a active valid route for a destination can generate a RREP following normal AODV rules if the 'U-bit' is unset. If the 'U-bit' is set then the RREQ can only be answered by the destination. The RREP generated by the destination must have the 'D-bit' set and the destination must increment its sequence number. A RREP generated by any node must have the flooding id of the RREQ copied to the RREP.

Accepting RREPs: A node possessing a valid destination sequence number can accept RREPs following AODV rules, provided the node has cached the RREQ corresponding to the (source, flooding id) pair obtained from the RREP and P is true. In-addition, a node that has no destination sequence number state can accept a RREP only if the 'D-bit' is set.

Relaying RREPs: When relaying RREPs, a node must lookup the cached (source, flooding id) pair from the RREP, and find the corresponding $reversehop$. The RREP must be forwarded to the reverse hop. If the received RREP has the 'D' bit set, the relayed RREP must have the 'D' bit set.

Routing Table Entries: A node can delete its routing table state for a destination entry if necessary at any time, and is not required to store the destination sequence number for arbitrary periods of time to ensure correct protocol operation. The node must also set P to false, for all (source, flooding id) RREQ pairs cached for this destination.

The reverse hop cached for the (source, flooding id) pair and the relaying rules are necessary to relay the RREP along the same reverse path. This is a necessary condition for the convergence of the protocol. If the more recent specification for AODV [4] is used, then the reverse hop caching rules for relaying RREQs and RREPs are not required since the RREQs and RREPs carry the path traversed.

To ensure robustness, the following parameters related to routing table and cache state need to be modified as follows.

Flooding Id: The flooding id counter at a node needs to be derived from its real time clock(64 to 32-bit). This is essential because when a node reboots and participates immediately, it loses its state for the last used flooding id. If previously used flooding ids are repeated, the RREQs will not be forwarded. Loss of precision from a real-time clock is acceptable as long as the safe interval is very high compared to the time old RREQs and cached state can exist in the network. We note that this scheme needs to be even applied to 'soft-state' protocols like the Dynamic Source Routing (DSR) [7] protocol or any on-demand routing protocol that performs route searches using the (source, flooding id) pair.

Destination Sequence Numbers: The destination must generate its sequence numbers from a 64-bit real-time clock. This method was proposed in LDR [6]. This is necessary to prevent loops that can be caused when a destination reboots and cycles to a previously used sequence number.

IV. Correctness of AODV-RSN

We argue the correctness of AODV-RSN in terms of loop-freedom and termination in the presence of node and link failures.

The proposed modifications do not affect the loop-freedom property of AODV. The 'U'-bit forces RREQs to traverse all the way to the destination, and the RREP traverses a loop-free path due to the property that a RREQ flood identified by a (source, flooding id) builds a tree rooted at the source. Therefore, the extra handling for the 'U' and 'D'-bits cannot cause any loops. Additionally, a node with an unknown destination sequence number can only re-learn a sequence number greater than the one previously stored for this destination. After state-loss, for a node to accept a RREP with the 'D' bit set, it must have relayed or originated the RREQ because it requires to have cached the corresponding (source, flooding id) RREQ and P must be true. Because the RREP must have been initiated by the destination after this time, and old RREPs will be dropped because of lack of the cached state or P being false, the RREP processed must carry a destination sequence number greater than before state loss at the node.

When the network is partitioned, we have to prove that all nodes invalidate their routing table entries for the partitioned destination in the presence of link failures, and node state loss. Following the default RERR rules of AODV, the RERRs should eventually propagate upstream along the directed acyclic graph in finite time. During this time we argue that nodes cannot keep learning newer routes from upstream nodes, which can lead to count-to-infinity behavior. When nodes reboot or lose state, only RREPs with the 'D' bit set can be used to update their route entries, which is not possible in a partitioned network and hence these nodes can never learn a new route. On link failures or otherwise, nodes with valid sequence number entries can learn routes from a downstream node that has a higher sequence number than the one in the request. However, because only the destination can reset (increase) its destination sequence number, within a finite time all nodes should have the highest destination sequence number entry and future route searches cannot be answered by any node in this partition. Assuming a finite probability that RERR messages eventually be delivered to all nodes, then all nodes invalidate their route entries.

In a stable connected network, we have to show that a source sending RREQs for a destination is able to establish a route within finite time. This proof of convergence is similar to the one for LDR [6](pp.60, Theorem 5) considering just the cases with sequence numbers. A node relaying a RREQ with the 'U'-bit set is equivalent to relaying the RREQ with the highest known destination sequence number in the network (i.e., the one stored at the destination). The rest of the details are identical since a RREQ with 'U'-bit set generates a RREP with the highest sequence number and 'D' bit set which will be acceptable at all nodes relaying the RREP along the reverse path.

V. Performance Comparison

We compare the original specification of AODV with AODV-RSN under varying loads and mobility. Simulations are run in Qualnet[12]. The parameters are set as in [11]. Simulations were performed on two scenarios, a 50-node network with terrain dimensions of 1500m x 300m, and a 100-node network with terrain dimensions of 2200m x 600m. Traffic loads were CBR sources with a data packet size of 512 bytes. Load was varied by using 10 flows (at 10 packets per second) and 30 flows (at 4 packets per second). The MAC layer used was IEEE 802.11 with a transmission range of 275m and throughput 2 Mbps. The simulated time is 900 seconds. Node velocity was set between 1 m/s and 20 m/s. Flows have an exponentially distributed length with a mean of 100 seconds. Each combination (number of nodes, traffic flows, scenario, routing protocol and pause time) was repeated for nine (9) trials using different random seeds. To demonstrate the limitation of reboots in AODV, where nodes have to wait a $DELETE\_PERIOD$, we additionally simulate the 50-nodes,10-flows sample scenario. Here, every node reboots after 50 second intervals, periodically, during the entire simulation time. Although unrealistic, the purpose is to show the effects of wait times on performance. AODV-RSN retains sequence number state unless in the case of reboots, where state is lost. After reboots, the flooding id is retained in both AODV and AODV-RSN to simulate our real-time modification.

We address four performance metrics. *Delivery ratio* is the ratio of the packets delivered per client/server CBR flow. *Latency* is the end to end delay measured for the data packets reaching the server from the client. The *network load* is the total number of control packets divided by the number of received data packets. *Data hops* is the number of hops traversed by each data packet (including initiating and forwarding) divided by the total number of received packets in the network. This metric takes into account packets dropped due to forwarding along incorrect paths, and provides a measure of the quality of the routes.

Tables I and II summarize the results of the different metrics by averaging over all pause times for the 50-nodes and 100-nodes networks. The columns show the mean value and 95% confidence interval. Table III summarizes the results for the scenario that has nodes rebooting periodically.

In the 50-nodes, 10-flows and 30-flows scenario, the summarized results of the different performance metrics for AODV and AODV-RSN remain within confidence intervals. However, the latency and control overhead for AODV-RSN are marginally on the higher side. This can be explained due to the fact that RREQs relayed by nodes that have no knowledge of the destination sequence number require the destination to answer with a RREP.

The 100-nodes, 10-flows and 30-flows scenarios, show interesting results as AODV-RSN seems to have a more significant performance improvement over AODV, although the confidence intervals of packet delivery, latency, and data hops overlap. The most significant improvement is in the

TABLE I

PERFORMANCE AVERAGE OVER ALL PAUSE TIMES FOR 50 NODES NETWORK FOR 10-FLOWS AND 30-FLOWS

Flows	10	30	10	30	10	30	10	30
Protocol	Delivery Ratio		Latency (sec)		Net Load		Data Hops	
AODV-RSN	0.988±0.005	0.788±0.041	0.027±0.008	0.739±0.232	0.346±0.085	3.619±0.804	2.600±0.180	2.885±0.266
AODV	0.994±0.002	0.763±0.047	0.017±0.003	0.984±0.327	0.267±0.066	4.419±1.132	2.588±0.166	2.924±0.280

TABLE II

PERFORMANCE AVERAGE OVER ALL PAUSE TIMES FOR 100 NODES NETWORK FOR 10-FLOWS AND 30-FLOWS

Flows	10	30	10	30	10	30	10	30
Protocol	Delivery Ratio		Latency (sec)		Net Load		Data Hops	
AODV-RSN	0.982±0.005	0.670±0.034	0.048±0.010	0.845±0.150	1.050±0.244	11.069±1.697	3.938±0.374	4.488±0.350
AODV	0.989±0.004	0.662±0.087	0.034±0.006	1.184±0.485	0.915±0.243	17.079±15.952	3.841±0.344	4.693±0.429

tight confidence intervals for control overhead in AODV-RSN (11.06±1.69) compared to AODV (17.07±15.95). This result correlates with our illustration about AODV's termination properties. In the highly congested scenarios (30-flows, 120pps), after a route failure, the RERRs for a destination, can suffer delays or be lost before propagating to all nodes upstream. In the meantime, downstream nodes may delete this destination sequence number after $DELETE\_PERIOD$ and issue new route requests which can be answered by nodes upstream. Depending on how the RERRs are delayed, it is possible to form loops or undirected cycles, which cause additional delays due to nodes updating their routing tables to use routes that are no longer valid. We believe that this is one of the main reasons that AODV suffers from excessive request flooding in the highly congested scenarios, which has been noted in previous publications [9], [2]. AODV-RSN, however, has a safe destination sequence number reset and does not suffer from the above anomaly exhibited by AODV.

TABLE III

PERFORMANCE AVERAGE OVER ALL PAUSE TIMES WITH NODES REBOOTING EVERY 50-SECONDS

Metric	AODV-RSN	AODV
Delivery Ratio	0.920±0.021	0.713±0.009
Latency (sec)	0.056±0.015	0.026±0.005
Net Load	1.224±0.412	0.514±0.076
Data Hops	2.511±0.185	2.561±0.179

Table III summarizes the results for the scenario that has nodes rebooting periodically every 50-second interval for the 50-nodes, 10-flows scenario. Nodes running AODV cannot participate in any routing action for $DELETE\_PERIOD$ after a reboot, whereas nodes on AODV-RSN can participate immediately. AODV has a very low packet delivery of (0.713±0.009) compared to AODV-RSN (0.920±0.021), showing the effects of forced waits on node reboots. Nodes rebooting in AODV drop the packets if they are sources of flows or affect network connectivity.

VI. CONCLUSION

We have identified serious robustness problems in the current AODV specification that have gone undetected to date. We have shown that the use of a $DELETE\_PERIOD$ in the current AODV specification can result in loops, de-facto network partitions on reboots, and even count-to-infinity behavior. Our proposed solution, AODV-RSN, modifies AODV by eliminating the need for the $DELETE\_PERIOD$ by requiring the destination to answer RREQs from nodes that have no current state for the destination. Simulation results show that AODV-RSN fixes the correctness problems with AODV, without sacrificing performance. AODV-RSN can be operated in networks with "unbounded" queueing delays, without affecting the correctness of the protocol. The performance results for the scenario where nodes reboot emphasizes the advantage of having a routing protocol in which nodes participate in routing actions immediately upon reboots.

REFERENCES

[1] K. Bhargavan, C.A. Gunther, and D. Obradovic. "Fault Origin Adjudication," *Proc. Workshop on Formal Methods in Software Practice*, Portland, OR, Aug. 2000.

[2] J. J. Garcia-Luna-Aceves, H. Rangarajan, "A New Framework for Loop-Free On-Demand Routing Using Destination Sequence Numbers," The 1st IEEE International Conference on MASS, October 25-27, 2004, Fort Lauderdale, Florida, USA.

[3] S. Gwalani, E. M. Belding-Royer and C. E. Perkins. "AODV-PA: AODV with path accumulation," *ICC 2003*, vol. 26, no. 1, May 2003, pp. 527 - 531.

[4] C. E. Perkins, E. M. Belding-Royer and I. D. Chakeres, "Ad hoc On-Demand Distance Vector (AODV) Routing", IETF Internet Draft, draft-perkins-manet-aodvbis-01.txt, January 2004.

[5] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol," Request for Comments 3626, October 2003.

[6] J. J. Garcia-Luna-Aceves, M. Mosko and C. Perkins, "A New Approach to On-Demand Loop-Free Routing in Ad Hoc Networks," *Proc. PODC 2003*, pp. 53-62, Boston, Massachusetts, July 13–16, 2003.

[7] D. Johnson et al, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," IETF Internet draft, draft-ietf-manet-dsr-09.txt, April 2003.

[8] R. Ogier et al., "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF),"Request for Comments 3684, February 2004.

[9] H. Rangarajan and J.J. Garcia-Luna-Aceves, "Using Labeled Paths for Loop-free On-Demand Routing in Ad Hoc Networks," *Proc. ACM MobiHoc 2004*, Tokyo, Japan, May 24–26, 2004.

[10] C. Perkins et al., "Ad hoc On-Demand Distance Vector (AODV) Routing," Request for Comments 3561, July 2003.

[11] C. Perkins et al. "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *IEEE Personal Communications*, 8(1):16 – 28, Feb 2001.

[12] Scalable Network Technologies. Qualnet 3.5.2.