A Survey of Reliable Broadcast Protocols for Mobile Ad-hoc Networks

Einar Vollset
School of Computing Science
University of Newcastle
einar.vollset@ncl.ac.uk

Paul Ezhilchelvan
School of Computing Science
University of Newcastle
paul.ezhilchelvan@ncl.ac.uk

Abstract

Providing reliable multicast is a basic requirement to build more advanced distributed protocols such as total order or leader election, and much research has gone into providing this functionality for wired networks. However, the protocols developed for wired networks tend to be unsuitable for deployment on mobile ad hoc networks (MANETs), as these do not take into account the node mobility or increased sensitivity to network load and congestion.

Recently, a few reliable multicast protocols designed explicitly for MANETs have been proposed which take into account the characteristics of MANETs. This paper will given an overview of the proposed protocols and evaluate the relative merit of each.

1. Introduction

Mobile Ad-hoc Networks (MANETs) are networks of mobile nodes communicating over multi-hop wireless links without the support of any infrastructure such as base stations. A lot of research in recent years has concentrated on providing routing functionality, both multicast (ODMRP[5], CAMP[7], MAODV[10]) and unicast (AODV[9], DSR[4]), for this environment. However, until recently only a small amount of this research has focused on providing *reliability guarantees beyond best-effort*.

Providing this type of delivery guarantee, particularly in the multicast domain, is crucial if reliable group communication paradigms, such as agreement and total ordering, are to be developed for MANETs. This in turn is highly desirable, as it simplifies the development of fault-tolerant and reliable distributed applications for MANETs.

In this paper we describe the current approaches to reliable multicasting in MANETs, starting with a look at deterministic protocols which attempt to enforce "strong" reliability guarantees, followed by probabilistic protocols which provide guaranteed delivery with a certain probability. Each protocols operation will be outlined and the performance of each analyzed. Additionally, the general strengths and weaknesses of each class of protocol will be discussed.

2. Deterministic protocols

Deterministic protocols are multicast protocols which provide "all-or-nothing" delivery guarantees for the delivery of messages to a group of nodes in a MANET. Protocols offering this kind of delivery guarantee most often tend to try and detect and repair failures, either at the source, or locally where the route is broken. The next 4 subsections contains an overview of 4 such protocols, this is then followed by a discussion of the relative merits of this class of protocols.

2.1. Reliable Broadcast (RB)

The Reliable Broadcast (RB)[8] protocol was the first reliable broadcast protocol designed explicitly for MANETs. The authors argue that for low mobility MANETs, well known spanning tree algorithms could be used, as the network is essentially a static one. On the other extreme, where the mobility of the MANET is very high, the authors claim that: "there is no alternative to flooding". The RB protocol is aimed at providing exactly-once delivery semantics for MANETs, when the mobility is in between these two extremes, and also providing the ability to switch to flooding once the rate of topology change in the MANET is deemed too high. The protocol handles disconnections and network partitions as long as these are temporary in nature.

Protocol Operation The protocol assumes the existence of a clustering algorithm, and works by having each node wishing to broadcast a message do a blocking send to the cluster-head of the cluster it is currently in. The cluster-head then sends the message to all the nodes in its cluster, and waits for acknowledgments from each of the cluster members. Any nodes acting as gateway will then forward

the message onto the gateway or cluster-head of the cluster to which it is the gateway, and it will delay the acknowledgment of message received from the original cluster-head until the message has been successfully diffused in the nearby cluster. This in turn could involve a recursive wait, as the this cluster might be connected to another cluster. In this way, the protocol essentially constructs a routing tree structure among the clusters where messages and acknowledgments will travel. To deal with node mobility, the protocol will switch to flooding acknowledgments back to the cluster-head of the cluster with the originating node. This is required as acknowledgments from the destination nodes' cluster-heads is essential for the protocol to ensure that the message is "stable", i.e. the message has been received by all destinations.

In order to guarantee the liveness of the protocol, the authors assume that the following assumptions hold:

Eventually the network topology stabilizes for the time required to guarantee that:

1. if there are pending messages for a host pi, then pi receives at least one of these messages, and it succeeds to notify the reception before the topology changes again;

2. a host remains cluster-head for the time necessary to guarantee that the exchange of status information with the other cluster-heads is successfully completed and, if there are partially diffused messages, that at least one of them is known amongst the cluster-heads.

Performance evaluation of the protocol The authors only present very scare simulation information, but present both a proof of the correctness as well as an analysis of the complexity of the protocol. The proof of correctness is omitted here, but essentially proves that, given the liveness properties stated above, the protocol provides reliable broadcast both in static and dynamic network environments. The complexity analysis of the protocol identifies the message complexity, time complexity and memory complexity of the protocol, where message complexity is defined as the amount of messages that the protocol exchanges to broadcast one message, the time complexity is defined as the elapsed time between the generation of a message and the time in which it becomes "stable" and memory complexity is defined as the length of the unstable data structure. The preliminary results presented in the paper indicates that, if the topology changes do not affect the cluster-head and gateways, the complexity of the algorithm is O(n). As a consequence of flooding, it can degenerate into $O(n^2)$. Additionally the authors give a term for message complexity due to the change of cluster-head as well as the message complexity due to the movement of a node between clusters. These two terms rely crucially on the time it takes to elect a cluster head and the time it takes a node to disconnect from one cluster to reconnect to the next respectively.

A discussion of the protocol RB has the stated intention of being in between spanning tree and flooding protocols in terms of flexibility and efficiency. On the face of it, the protocol has a low message complexity (O(n)) when no cluster-head changes occur and switches to flooding when the routing structure provided by the clustering is lost due to mobility. However, it can be argued that in the face of medium to high mobility, the changes in gateways and cluster-heads will be fairly frequent, thus forcing the protocol to use flooding, timeouts and retransmissions, which severely degrades the performance of the protocol.

2.2. Adaptive Reliable Broadcast (ARB)

The Adaptive Reliable Broadcast (ARB) protocol[3] is a reliable multicast protocol that adjusts to the rate of topology change in the network. The protocol makes a similar assumption to RB with regard to the liveness of the network, that is:

> if there are pending messages for a processor p, then the network eventually remains constant for long enough so that p receives at least one of these messages and acknowledges the receipt.

Protocol operation The protocol works by constructing a core based shared multicast tree when nodes in the multicast group send JOIN messages to the core node. Whenever a sender wants to multicast a message to members of the group, it sends a multicast message to the core node of the given group, which initiates the dissemination of the message down the multicast tree. The acknowledgments from individual nodes travel in the opposite direction up the tree to the core node, and a message is said to be "stabilized" when the core node has received acknowledgments from all the group nodes. In order to reduce bandwidth consumption, the protocol uses acknowledgment aggregation. The knowledge of message stabilization is piggy-backed on subsequent multicast messages.

In case of fragmentation due to node movement, the concept of a forwarding region is introduced which is used to "glue together" the fragmented multicast tree. This gluing involves flooding of the forwarding region by nodes which which witness the topology change due to node mobility. In addition, there is a notion of nodes pulling the messages when they (re)join the multicast tree.

Performance evaluation of the protocol The authors present no simulation results or any complexity analysis of

the protocol. They do however present proof of correctness of the protocol.

A discussion of the protocol ARB is very similar in spirit to RB, and thus suffer from much the same problems with regard to the suitability of the protocol when faced with medium to high mobility. Although no simulation results have been reported for the protocol, it seems safe to assume that the protocols performance will degrade severely due to the tree structure used to forward messages. This tree structure is supplemented by the so called "gluing together" of pieces of the multicast tree when mobility fragments it. It seems obvious that as mobility increases, this gluing will become more and more frequent, thus in the worst case rendering the protocol to be a flooding protocol using timeouts and retransmissions, much as RB.

2.3. Reliable Multicast Algorithm (RMA)

The Reliable Multicast Algorithm (RMA) [2], is a multicast algorithm which utilizes the concept of link lifetime instead of hop-count to determine which route a packet should take. The protocol provides guaranteed delivery by using acknowledgments from the destination to the source. The burden of providing reliable delivery is placed on the sender, much like the other protocols in this section, however, the protocol designers claim that using the lifetime metric provides deterministic reliability with low overheads.

Protocol operation The protocol requires that all senders know the identities of all the members of the multicast group, and a send is performed by first the sender looking up each destination in the routing table. If the routing table contains the destination, combine that message with other destinations which have the same next-hop and post an MKNOWN message through that next hop. If any destinations is not known, combine all the unknown destinations into a MUNKNOWN message and broadcast. Wait for a specified amount of time or until acks from each node has been received. If acknowledgment is not received from all nodes, put out another MUNKNOWN message with the RETRANSMIT flag set. Repeat this a pre-specified number of times, or until all acks have been received. In addition, the sender will use the path information from the acks to update the routing table.

When a receiver receives a message, it sends a MACK along the path which the message came. If multiple messages arrive, only send a new MACK if the path of the second message is better. If a RETRANSMIT message is received, send a MACK along the path of the message if the source has not seen the message before, or if it has, broadcast a BMACK back to the sender.

Intermediate nodes upon receiving a message will, if the message is not a duplicate, update the routing table with the information in the message, and check if there exists a route to the destination. If a route exists, send the message on, if a route does not exist, broadcast the message.

Updating the neighbourhood set is achieved by using HELLO messages, and timeouts based on the predicted link lifetime with the neighbouring node. Updating the multicast group membership information is by explicit JOIN/LEAVE messages.

Performance evaluation of the protocol The protocol designers present experimental data intended to show the reliability and effectiveness of the protocol. The simulation environment has been the RELSIM simulator with the following test parameters. A simulation area of 1000x1000 meters containing 50 nodes with 200 meters transmission range was used. The mobility model chosen was the random waypoint model with 10 and 50 seconds rest time, and variable speed of the nodes from 5-50 m/s. No indication was given as to how long the simulation ran, or how many senders and receivers the simulation run involved. The protocol was compared to the MAODV protocol, with regards to packet delivery ratio and control- and data overhead. The simulation results indicate that "RMA has a packet delivery close to 1", and as speed increased above 5m/s, RMA has a lower data- and control overhead than MAODV (See [2] for full simulation results).

A discussion of the protocol The protocol aims to provide deterministically reliable multicast with low-overheads, and the simulation results seems to indicate a certain degree of success. The novelty of the approach is primarily the use of link lifetime as a metric for route creation, although the authors claim that the construction of a dynamic, undirected graph is novel, although this structure is similar to the one utilized in for example ODMRP. The link lifetime metric is a simple one, which basically aims to predict the future based on past behaviour, and although the introduction this new metric is an interesting one, the simulation results presented in the paper shows little improvement in control- or data overhead compared to a hopcount approach. These observations in turn prompts the question: how can deterministic reliability be achieved with such low overheads, when the only novel feature of the protocol seems to have little impact? Essentially, the protocol contains features of both ARB, with the concept of using broadcast if no route is known, and RALM, with collecting the acknowledgments from each member of the multicast group individually. The low overhead associated with the protocol may be attributed to the parameters chosen for the simulation, where the number of nodes, their wireless range and the size of the simulation area, indicates a very dense

network, as well as the fact that rather long rest times has been chosen for the random waypoint model. Studies have shown that using long rest times for the random waypoint model produces an essentially static network, thus explaining the low overhead.

An aspect of concern is the implication in the paper that JOIN and LEAVE messages are simply flooded throughout the network without any reliability guarantees. Such a scheme could easily ruin the deterministic delivery guarantees provided by the protocol.

2.4. Reliable Adaptive Lightweight Multicast (RALM) algorithm

The Reliable Adaptive Lightweight Multicast (RALM) algorithm [11] is a reliable congestion controlled protocol which uses a TCP-like error and congestion control by picking one multicast receiver at a time in a round robin fashion and reliably transmitting data to these multicast receivers.

Protocol operation The protocol assumes that the multicast receivers are known to the source, either through receiver discovery or by advance knowledge. The authors claim this is a reasonable assumption in certain scenarios. The protocol works as follows:

When a source has multicast data to send, it picks a receiver from the Receiver List, containing the known receivers. It then starts to send messages to the multicast group, with the chosen receiver (called the feedback receiver) included in the packet header instructing it to unicast a reply containing an ACK or a NACK and a sequence number. All other receivers in the multicast group simply process the message without acknowledging to the sender. If the feedback receiver determines that packets are missing, it will request the missing packets one at a time from the source. The authors' philosophy behind transmitting each lost packet one at a time is to slow down the transmission of the source when congestion is detected. Both new and retransmitted packets are broadcast, which implies that most of the multicast group members should receive the data packets. Once the feedback receiver has received all the packets, it unicasts an ACK back to the source, upon which the source picks a new receiver from the Receiver List and repeats the process until the list is empty.

The central new contribution of RALM is the introduction of a TCP-like window-based congestion control mechanism used to reduce overhead and increase efficiency. The congestion control mechanism works by the sender sending a number of packets to the feedback receiver before expecting a reply. This number of packets is determined by the window size. The window size is varied depending on the successful receipt of ACKs from the feedback receiver. In particular, if ACKs are received, the window size is increased either exponentially until the "slow start threshold" has been reached, or linearly after that. On the other hand, if a NACK is received or a timeout occurs, the window size is halved. The difference between TCP congestion control and the congestion control of RALM is that: a) one global window is maintained for all receivers and b) only the last packet of the current window needs to be acknowledged.

Performance evaluation of protocol The authors present simulation results comparing RALM to UDP and SRM running on top of ODMRP. UDP was chosen to compare RALM to a basic multicast protocol without reliability guarantees, while SRM was chosen to compare RALM to a protocol which only uses error control to achieve reliable delivery. The QualNet simulator was used with the following test parameters: A simulation area of 1500m x 1500m containing 50 nodes randomly placed was used. The maximum radio propagation range was set to 375 meters and a two-ray ground reflection model with free-space path loss for near sight and plane earth path loss for far sight was chosen. The underlying MAC protocol was IEEE802.11b DCF (Distributed Coordination Function) with channel capacity of 2Mb/s. The mobility pattern was the random waypoint model. Three metrics were measured and compared: packet delivery ratio, control overhead and end-to-end delay. These three metrics where measured in 3 separate simulations which varied the traffic rate, the number of sources and the mobility respectively.

In the simulation measuring the three metrics against variable traffic rate, all nodes in the system are stationary, with 5 multicast senders and 10 multicast receivers. The packet inter-departure time was varied between 100ms and 500ms. The simulation shows that RALM achieves 100% packet delivery ratio, and outperforms both UDP and SRM with regards to control overhead and end-to-end delay. However, for traffic rates over 200ms, RALM only slightly outperforms UDP and it is worth noting, as the authors do, that the traffic rate is only the initial traffic rate for RALM, as it has built-in congestion control, unlike UDP or SRM.

In the simulation varying the number of sources, the nodes are again kept stationary, this time varying the number of sources from 10 to 40. In this simulation, only the packet delivery ratio is presented and RALM outperforms UDP and SRM quite substantially, although the packet delivery ratio of RALM slips beneath 100%, something the authors blame on the experimental setup (Finishing the simulation before RALM has the chance to reliably deliver all packets to all receivers as the number of receivers increase).

In the final simulation, mobility is varied between 0m/s and 50m/s using the random waypoint model. No indication is given as to the length of the rest-times of the nodes. In this simulation 5 multicast sources and 10 multicast receivers were used, and RALM again achieves a 100% packet deliv-

ery ratio, although it must be noted that SRM and UDP in particular are not very far of this mark.

Discussion of the protocol The protocol's main contribution lies in introducing the need for congestion control to achieve reliable multicast delivery, and the simulations initially seem to support this view. Particularly the simulation showing the adverse effects of an increase in traffic rates on SRM, showing that not only does RALM maintain 100% packet delivery guarantees, but also manages to keep the end-to-end delay substantially lower than SRM. This indicates that congestion control is indeed a useful feature in a reliable multicast protocol.

That being said, control overhead and end-to-end delay does not give the full picture of a protocol's performance. One measure in particular which we feel should have been included is the data overhead, which we believe will be substantial in a multicast protocol where packets are reliably transmitted to each individual receiver in the multicast group.

Additionally the fact that the traffic rate and the number of sources simulations were performed with zero mobility and that no indication was given as to the length of the rest times used in the mobility simulation, leaves open questions as to the suitability of the protocol to scenarios where mobility is an integral part of the environment, something which must be assumed in a MANET setting. In particular, the simulations seemed to be geared towards scenarios where packet loss is mainly due to congestion, and not route failure, which favours the congestion control mechanism in RALM over the error control mechanism of SRM.

We also note that the end-to-end delay and the control overhead metrics were not included for the second and third simulation runs (i.e. variable number of senders and mobility).

2.5. A discussion of deterministic protocols

Current deterministically reliable multicast protocols suffer badly because of the tradeoff required between reliability and scalability/mobility. This is because either the protocol attempts to construct a routing structure on which messages and acknowledgments travel, as in RB, ARB and RMA, or as in the case of RALM, the sender is required to reliably communicate with each of the receivers in the multicast group. Clearly when the mobility and/or group size increases in both of these scenarios, the performance of the protocol is severely degraded, as the protocols eventually resort to using flooding and timeouts/retransmissions. This in turn means that in extreme cases, the network throughput could sink below a usable level, thus rendering it useless.

However, these protocols do provide deterministic delivery guarantees, unlike the protocols in the next section.

3. Probabilistic protocols

Probabilistic protocols are protocols that guarantee delivery with a certain probability. Although not as safe as guaranteed protocols, the probabilistic protocols typically have less restrictive assumptions and constraints associated with them as well as reduced overhead, thus making them in many cases the better (or even the only) choice for certain MANET settings.

The following subsections describes one addition to a multicast protocol, and one multicast protocol, which provides probabilistic delivery guarantees.

3.1. Anonymous Gossip

Anonymous Gossip (AG) [1] is an addition to increase the reliability of any on-demand protocol, by utilizing gossip. Gossip is the technique where nodes outside the normal message delivery phases exchange information on which messages they have received, thus increasing the reliability of the system. AG is added to MAODV, a multicast adaptation of AODV, but the authors claim that AG should work with any on-demand multicast protocol.

Protocol operation AG involves two phases, in the first phase the primary multicast protocol is used to unreliably multicast the message, m, to be sent to the group. In the second phase, AG is used to recover missed messages from other members of the group which might have received it. This phase consists of periodic rounds of the following steps being taken by the nodes:

1. Node A randomly chooses another member of the group, say B.
2. A sends B the information about messages it has received or not received.
3. B checks to see if it has received any of the messages listed by A
4. Then A and B could exchange messages which are not a part of each other's message history.

AG does not require the nodes to know the membership of their group, as: "maintaining even partial group membership is extremely expensive and would significantly reduce the throughput of the network". The protocol makes each node randomly select one of its neighbours to send a gossip message to. This node then either accepts the gossip request and starts gossiping with the node, or propagates the gossip request. Either way, the node accepting the gossip request unicasts a gossip reply to the initiating node. In the implementation of AG presented, only nodes in the multicast tree of MAODV are to be considered for propagating the gossip

message. The rationale for this is the fact that these are the only nodes which participate in routing any messages for a given multicast group, and also that propagation along the multicast tree prevents loops in the propagation.

Choosing which nodes to gossip with obviously has a huge effect on the reliability of the system, and gossiping only with a select few nearest neighbours could in theory leave the system susceptible to message loss covering a whole locality. However, as gossip is a periodic and constant activity, always gossiping with distant nodes would put a heavy burden on the network. The approach taken in AG is to gossip locally with high probability and gossip with distant nodes with low probability. The details on how this is achieved are given in the paper, but is very closely linked to MAODV, and thus omitted. In addition to the gossip strategy dependent on MAODV, the paper also introduces the concept of "cached gossip". This involves gossiping with nodes whom the node knows is part of the multicast group. Again, this is aimed at reducing the load on the nodes in the multicast tree.

Performance evaluation of the protocol The authors present a simulation of AG on GloMoSim, where a 200m x 200m fixed area was used. MAODV was implemented, and the mobility model used was random waypoint with rest times uniformly distributed between 0 and 80 seconds. Each simulation was run for 10 minutes, and the MAC layer protocol was IEEE 802.11b with 2Mb/s bandwidth. Every group member sends one gossip request per second, and each gossip message could request at most 10 lost messages. The size of the membership cache was set to 10, and each member could remember up to 200 lost messages and 100 of the last received messages. Three parameters were varied: transmission range, maximum speed and the number of nodes.

In the transmission range simulation, the transmission range was varied between 45 and 85 meters, with 6 different values for maximum speed. The paper presents graphs comparing the results of pure MAODV and MAODV with AG for 2m/s and 0.2m/s maximum speed. These graphs indicate that substantial improvements in the packet delivery ratio is achieved by adding AG, although the packet delivery ratio for AG never reaches 100% except where the max speed is 0.2m/s and transmission range is 75m+.

In the maximum speed simulation, the speed was varied between 0.1m/s and 10m/s with a fixed transmission range of 75m. In this simulation, as above, the addition of AG improves the packet delivery ratio substantially compared to pure MAODV, but only provides close to 100% delivery at speeds below 0.3m/s.

In the number of nodes simulation, the number of nodes was varied between 40 and 100 nodes. In one of the experiments the transmission range was varied so as to keep the average number of neighbours the same. In this experiment the packet delivery ratio declined steadily as the number of nodes increased. In another experiment, the transmission range was kept constant at 55m. In that experiment the packet delivery improved, then declined as the number of nodes increased. The authors blame this decline in packet delivery ratio on the increased congestion when a large number of nodes is used. Again AG outperformed pure MAODV.

A discussion of the protocol The authors present a novel way to achieve reliable multicast in MANETs by using anonymous gossip, and the simulations show that the packet delivery ratio is greatly enhanced compared to pure MAODV. However, 100% packet delivery is not achieved, and no indication has been given of the extra cost in message complexity the enhanced reliability comes with, although one would expect it to be less than that of the deterministic protocols. Additionally, it is unclear if the use of a pull mode of information exchange, where a node will locally determine which messages it has lost, is the best one. Finally, because the gossip done by AG is guided by the underlying routing structure of MAODV, AG looses the property of predictable behaviour, making it impossible to analytically predict its probabilistic delivery ratio, unlike the protocol in the following section.

3.2. Route Driven Gossip

Route Driven Gossip (RDG) [6] protocol uses a pure gossip scheme, which gossips uniformly about messages, negative acknowledgments and membership information without requiring an underlying multicast primitive (i.e. unlike AG it doesn't rely on a MANET multicast protocol).

Protocol operation The authors define their problem specification as being: "If some group member sends out a flow of M packets, a certain group member receives a fraction F of all M packets with probability f(F)." In this definition, F is known as the reliability degree and f the reliability probability distribution. The authors expect f(F) to be predictable based on simple information like packet loss ratio.

The protocol proposes using the concept of route driven gossip, rather than view driven gossip. The difference being that in view driven gossip the gossip is based on the view of the group membership by the source node. The authors argue that because in an on-demand setting (i.e. using DSR or AODV as the unicast routing protocol), needing to request a route to the node with which a node wants to gossip can greatly increase the network traffic. The route driven approach described below uses only the partial, random view of the group members as determined by the routing information provided by the routing substrate.

The protocol operation is divided into three tasks: join, gossip and leave.

The join task is initiated by any node intending to join a multicast group, and involves the joining node flooding the network with a GROUPREQUEST, intended to search for the existence of other group members. Any member node receiving such a request will add the joining node to its active view, the data structure containing member nodes to which a route is known, and respond back to the joining node. The joining node then constructs its active view based on the responses to its GROUPREQUEST.

The gossip and leave tasks are presented jointly, as the leave task involves using the gossip task. Each member of the group periodically generates a gossip message and gossips with F other nodes randomly chosen from its active view (any message is gossiped t times). The gossip message includes packets the node wishes to gossip about (i.e. gossip-push), messages the node knows to be missing (gossip-pull) as well as information on the group membership, and whether it wishes to leave the group (this is the leave task). A group member receiving a gossip message will: 1) remove obsolete members from its view, 2) add new members to its view, 3) update its data buffers with any new packets and 4) respond to gossiper-pull.

The paper includes an addition to the pure RDG protocol, namely one which is topology aware. This protocol, named TA-RDG, uses the topology information available from for example the unicast routing protocol (DSR for example could provide the hop count to the known group members), and this is used so that the protocol gossips with a near member with a higher probability than a far one, much like the approach taken with AG.

Performance evaluation of the protocol The authors present both an analysis of the probability of packet delivery at each node, and a simulation. The analysis derives a term for the single packet dissemination reliability, based on which the reliability probability distribution is derived. The simulation results were obtained using ns-2 with a 1000m x 1000m area and 100-200 nodes each with 250m transmission range. The mobility model is the random waypoint with a max speed between 2-20 m/s and average rest times of 40s. The simulation results closely match that of the analytical predictions, and show that the reliability of the protocol moves towards 100% as the number of gossip rounds increase. Additionally, a comparison with AG is made, in which RDG is shown to have a higher probabilistic delivery guarantee (between 90-100%) than AG (75-95%).

A discussion of the protocol The RDG protocol provides a practical solution for providing probabilistic reliability, allowing the protocol to be tuned to suit the environment. The authors provide little information on the overhead associated with the protocol, but one can assume, as with AG, that the overhead is limited compared to a deterministic protocol. However, choosing a large transmission range and high rest times for the random waypoint model does raise the question of the suitability of the protocol under more dynamic mobility scenarios, particularly without the use of the topology aware functionality hinted at at the end of the paper.

3.3. A discussion of probabilistic protocols

The probabilistic protocols have been seen as a way to "fight fire with fire" in that the way to combat the lack of determinism inherent in MANETs is to apply probabilistic protocols to them. There is an obvious benefit to the gossip protocols presented here, as both AG and RDG provides a high (relatively) probability of delivery, and RDG in particular, as it has been designed for large groups, seem to scale well. However, the fact that no eventual delivery guarantee can be provided by any of these protocols, is an obvious draw back. In addition, a few researchers has argued that AG suffers from long delays to recover from losses, something which probably also could be argued for RDG.

4. Conclusion

In the area of reliable multicast for MANETs there currently seems to be no silver bullet which provides us with deterministic delivery guarantees and rapid delivery of messages. The two approaches to the problem (deterministic and probabilistic) both have their advantages and disadvantages. The main ones seem to be bad tradeoffs necessary between reliability and scalability/mobility for deterministic protocols and no deterministic delivery guarantees for probabilistic protocols.

Another general observation which can be made from this survey is that experimenters often choose simulation parameters, such as long rest times for the random waypoint model, which has been shown to render the network topology almost static. We believe any credible simulation comparing these (or any new reliable protocols) should aim to provide a more taxing environment for the protocols, to properly show the overhead associated with each, as overhead is, in our opinion, one of the most crucial measures of the suitability of any MANET protocol.

References

[1] R. Chandra, V. Ramasubramanian, and K. Birman. Anonymous gossip: Improving multicast reliability in mobile adhoc networks. In *Proc. 21st International Conference on Distributed Computing Systems (ICDCS)*, 2001.

[2] T. Gopalsamy, M. Singhal, D.Panda, and P. Sadayappan. A reliable multicast algorithm for mobile ad hoc networks. In *Proceedings of ICDCS*, 2002.

[3] S. Gupta and P. Srimani. An adaptive protocol for reliable multicast in mobile multi-hop radio networks. In *IEEE Workshop on Mobile Computing Systems and Applications*, 1999.

[4] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[5] S. Lee, W. Su, and M. Gerla. On-demand multicast routing protocol in multihop wireless mobile networks, 2000.

[6] J. Luo, P. T. Eugster, and J.-P. Hubaux. Route driven gossip: Probabilistic reliable multicast in ad hoc networks. Technical report, EPFL, 2002.

[7] E. L. Madruga and J. J. Garcia-Luna-Aceves. Scalable multicasting: The core-assisted mesh protocol. *ACM/Baltzer Mobile Networks and Applications, Special Issue on Management of Mobility*, 6(2):151–165, 2001.

[8] E. Pagani and G. P. Rossi. Reliable broadcast in mobile multihop packet networks. In *Proceedings of MobiCom*, 1997.

[9] C. Perkins. Ad hoc on demand distance vector (aodv) routing, 1997.

[10] E. M. Royer and C. E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Mobile Computing and Networking*, pages 207–218, 1999.

[11] K. Tang, K. Obraczka, S.-J. Lee, and M. Gerla. Congestion controlled adaptive lightweight multicast in wireless mobile ad hoc networks. In *Proceedings of ISCC*, 2002.