

Ad Hoc Routing Protocol Verification Through Broadcast Abstraction

Oskar Wibling, Joachim Parrow, and Arnold Pears

Department of Information Technology, Uppsala University,
Box 337, SE-751 05 Uppsala, Sweden
{oskarw, joachim, arnoldp}@it.uu.se

Abstract. We present an improved method for analyzing route establishment in ad hoc routing protocols. An efficient abstraction for Propagating Localized Broadcast with Dampening (PLBD) is developed. Applying this result we are able to verify networks and topology changes for ad hoc networks up to the limits currently envisaged for operational mobile ad hoc networks (MANETS). Results are reported for route discovery in the Lightweight Underlay Network Ad hoc Routing protocol (LUNAR) using UPPAAL and we provide an outline of how similar verifications can be conducted for DSR.

Keywords: Mobile ad hoc networks, routing protocols, formal verification, model checking, UPPAAL, LUNAR, DSR.

1 Introduction

Delivering data in an ad hoc network with mobile nodes requires new protocols. Traditional routing protocols are incapable of routing data packets efficiently in this type of situation, motivating emergence of new protocol proposals. Validation of these new protocols is principally through simulation. Simulation often fails to discover subtle design errors, and therefore formal verification is a promising approach.

In this work, we verify correct operation of the LUNAR [1] protocol route establishment in realistic general scenarios using a network diameter of up to eleven hops. We further describe how the route discovery phase in the DSR [2] protocol can be verified in a similar way. We have aimed for the modeling to be fairly straightforward and for the verification procedure to require a minimum amount of user interaction. The verification properties are formulated at a high and easily assimilated level, e.g. “route possible to set up”.

The operation responsible for most of the complexity in the verification of a LUNAR network scenario is Propagating Localized Broadcast with Dampening (PLBD). PLBD is used in the route discovery phase where a node tries to find a path to another node in the network. Each PLBD phase that is initiated at a node contains a globally unique identifier in order for nodes to keep track of which PLBD:s they have seen. As the name implies, the broadcast propagates through the network, which causes many message exchanges between nodes.

This in turn yields many possible interleavings and leads to exponential growth in verification complexity with regard to increasing number of nodes as well as topological changes in the network.

We show that in any network topology where nodes are positioned so that at a certain time it is *possible* to transmit a message over a link chain between two nodes, the PLBD reaches the intended receiver. Furthermore, we show that if there is at least one such path available then there is always a fastest path. This means that a PLBD initiated by a sender will reach the receiver first along this path. Moreover, all the nodes along the path are the first to receive the PLBD. In LUNAR, network nodes only react to the first specific PLBD they receive; subsequent ones are dropped. Therefore, we model reactions on the first PLBD packet of each type and can safely ignore the rest.

Using this technique, we can model LUNAR with timed automata and perform verifications for realistic network sizes using the UPPAAL [3] tool. Variations on the PLBD operation are also used for route discovery in other ad hoc routing protocols; one example is the DSR protocol. The DSR variant of PLBD differs in that other nodes than the intended receiver can respond to a route discovery, if they happen to possess a cached route to the destination. Therefore, instead of studying just the fastest path, we need to study a number of disjoint paths. This increases the verification complexity, but the saving is still substantial in comparison to studying all possible packet interleavings.

The remainder of this paper is organized as follows. Section 2 covers preliminaries needed to assess the subsequent sections. Section 3 describes our new verification strategy in general and Section 4 provides more detail regarding the actual modeling of LUNAR. Verification results are presented in Section 5. Section 6 describes how the DSR protocol can be verified in a similar way, and Section 7 gives an overview of related efforts in verifying MANET protocols. Finally, Section 8 contains conclusions and describes opportunities for future work in the area.

2 Preliminaries

2.1 Previous Work

In previous work [4] our result was to formally verify important properties of ad hoc routing protocols acting in realistic network scenarios. The scenarios we used are repeated in Figure 1 for clarity. We studied the LUNAR protocol since it combines simplicity with the key properties of more complex ad hoc routing protocols. The protocol was modeled by seeing each node in the network as a separate entity. Each propagating broadcast then hugely increased the complexity because of the possible interleavings of messages. When using this approach we quickly ran out of memory due to verification state space explosion. We were unable to verify networks with more than six participating nodes and very simple topology changes. Here, we refine our method and extend it to verifying significantly larger networks.

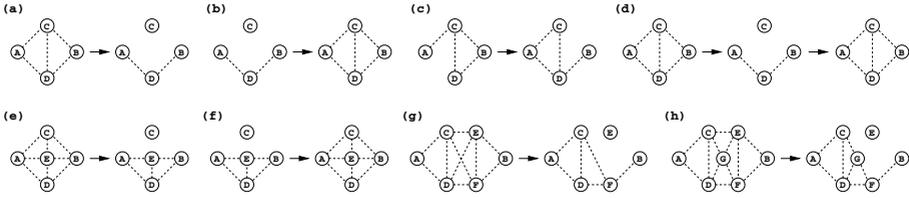


Fig. 1. Example classes of topology changes used in previous work

2.2 The Propagating Localized Broadcast with Dampening (PLBD) Operation

The reason for the state space explosion in the verification of LUNAR is the propagating localized broadcast operation (aka flooding) which works as follows:

- The broadcast is referred to as “localized” since each broadcast in a wireless network only reaches direct neighbors of the transmitting node and not nodes outside transmission range.
- Each node that initiates such a broadcast, tags the broadcast packet with a unique identifier (called the “series selector” in LUNAR).
- At each receiving node the broadcast identifier is compared with a local list to see if this particular packet has been seen before. If that is the case, the packet is just ignored. This mechanism prevents broadcast loops arising in the network.
- If the packet has not been seen before, and the receiving node is not the intended destination, the identifier is stored after which the packet is re-broadcast. Each neighbor, who has not seen the packet before will receive it together with neighbors that may already have seen the packet.
- The intended destination node also stores the identifier when and if it receives the packet, in order to be able to discard subsequent copies it might receive.

In our verification, we assume that this mechanism works, i.e. PLBD can be used as a primitive operation. In making this choice we run the risk of failing to detect subtle operational errors in the PLBD operation, potentially causing failure of the routing protocol. However, this risk is minimized by the analysis of the PLBD process needed to formulate the model.

When using PLBD, the only possible paths that packets can follow from a source to a destination are disjoint. That is, if the destination node receives a number of copies of the same PLBD, these must all have been transmitted through completely disjoint transmission chains. If two transmission chains should coincide at a node, then, since we assume that only one packet at a time can be delivered to each node, one of the packets would have been dropped and the other propagated. Different variants of PLBD have been proposed and used in other protocols [5]. The goal of these is to minimize the number of rebroadcasts needed to reach all connected nodes.

2.3 Brief Description of Ad Hoc Routing and LUNAR

A mobile ad hoc network (MANET) is a transient network that is set up to serve a temporary need, e.g. the exchange of files at a conference. It is assumed that nodes are mobile and that their location can change frequently. Therefore, node connectivity can also vary heavily. In the networks we study, multiple hops are possible between a source and a destination. This, in contrast to a fully connected network, means that nodes outside direct transmission range can be reached by traversing other intermediate nodes. In order to realize this, a routing protocol must be running on each node to forward packets from source to destination.

We study a basic version of the LUNAR protocol and use our earlier pseudo code description [6] to aid us in the modeling. The situation in which we wish to verify correct operation arises when one network node, S , has an IP packet to deliver to another node, D , but there is no route to that node available. In this situation the LUNAR route formation process is initiated at node S , which sends out a route request (RREQ) for the sought node using PLBD. On every retransmitting node, return information for the reply is temporarily stored. If the RREQ reaches D , that node will initiate a unicast route reply (RREP) destined for the node from which it received the RREQ. This node, as well as subsequent ones use the stored return information to re-address the unicast RREP for its next hop. On every step of the way back to node S , relays are also set up for label switching of wrapped IP packets later traveling along the found route. If node S does not receive a RREP within a certain time, it will issue a number of retries (using new PLBD identifiers). After that, the protocol will not take action until there is another IP packet that needs to be delivered.

2.4 General Assumptions

We use the following assumptions throughout this work.

- Unique id:s. It is possible for each network node to generate unique identifiers in isolation from the other nodes. In practice this can be implemented by appending the MAC address of a node to a monotonically increasing sequence number.
- Sequential delivery. Each node in the network can only receive and handle one message at a time. This means that we assume that relatively standard hardware is used in an actual implementation with no parallel processing of messages sent on different channels. Packets thus arrive at each network node in a strict time order.
- Bidirectional links. Only bidirectional links are possible in the network. Since 802.11 requires a bidirectional frame exchange as part of the protocol [2] this is not a significant limitation. It is, however, relevant since it affects the caching strategy of DSR.
- No persistent memory on nodes. If they go down, they lose their current route caches etc.

3 A Refined Modeling Strategy

3.1 Correct Operation of an Ad Hoc Routing Protocol

The definition of correct operation of an ad hoc routing protocol is taken from our previous work [4].

Definition 1. Correct operation of an ad hoc routing protocol

If there at one point in time exists a path between two nodes, then the protocol must be able to find some path between the nodes. When a path has been found, and for the time it stays valid, it shall be possible to send packets along the path from the source node to the destination node.

We said that “a path exists between two nodes”, meaning that the path is valid for some time longer than what is required to complete the route formation process. A route formation process is the process at the end of which a particular routing protocol has managed to set up a route from a source node to a destination node, possibly traversing one or more intermediate nodes.

In the following, we will need a more detailed definition of path existence which pertains only to the unidirectional case. The reason is to be specific about what nodes are connected at different time periods for use in the proofs that follow.

Definition 2. Existence of a unidirectional path

Assume nodes $X_0 = S, X_1, \dots, X_N = D$ (where $N \geq 1$). At time τ_0 a unidirectional path exists from network node S to D if, for all $n \in [0, N - 1]$, between times τ_n and τ_{n+1} node X_n has connectivity and can transmit to node X_{n+1} . Furthermore, between these times, node X_n does not have connectivity to any of the nodes $X_m : m \in [n + 2, N]$.

We require that $(\tau_{n+1} - \tau_n) = T_n : n \in [0, N - 1]$ where T_n is the time required by node X_n to transmit a message to any (or all) of its neighboring nodes (i.e. over one hop), plus the time for the receiving node(s) to handle the packet and prepare for possible retransmission.

Note that we do not limit ourselves to unicast transmission. In the case of LUNAR, the first phase of the route formation process is to send a route discovery along a path from source to destination. We only require this path to be unidirectional. For LUNAR, our previous definition of path existence thus implies Definition 2. Therefore, if the preconditions of Definition 1 hold, then we know there is a unidirectional path at that point in time. In our verification, we will (as before) also make sure that these conditions hold.

3.2 Focusing on the Packet Transformation

We here describe a remedy for the state space explosion problem whilst still being able to model check scenarios of interesting proportion.

In LUNAR, two types of message transfer are used: unicast and PLBD. This is the case for other (reactive) ad hoc routing protocols as well, although some in

addition use regular broadcast e.g. for neighbor sensing. Here, instead of being node centered, we focus on the packet. The idea is that every full (route setup - initial IP packet delivery) session begins with the source node, S , sending out a PLBD packet containing a RREQ for a particular destination, D . When this packet hits one or more other nodes, it can be viewed as being transformed into new packets. Once one of the rebroadcast packets reaches D (provided there is connectivity), this node will generate a RREP unicast packet destined for the node from which it received the RREQ. The RREP then traverses back through the network to S , all the time rewriting addresses. When the last RREP reaches S , it can send its initial IP packet along the found path.

The transformation is probably most easily seen in the case of the unicast chains: one packet comes into a node and another leaves. In the case of broadcast, we would like to be able to ignore all receiving nodes except one, which can then act as a message transformer.

3.3 Disregarding Unimportant Broadcast Receivers

To be able to motivate our claim that we can, at each step, disregard packets received by all broadcast receivers but one, whilst still being able to show important properties of the protocol we will need Theorem 2 below. First, however, we need a theorem that guarantees that we find a path if we use PLBD.

Theorem 1. Existence of a PLBD path

In a finite mobile network, if there at time τ_0 exists a possible unidirectional path between two nodes S and D , according to Definition 2, then a PLBD initiated from node S at this time will reach node D . The PLBD path is then the inverse of the following sequence of nodes: Node D , the node that broadcast the PLBD to D , and so on all the way back to node S .

Proof. See Figure 2 for an illustration. A PLBD with unique identifier β initiated by node S at time τ_0 will reach the direct neighbors of S . According to Definition 2 these direct neighbors either contain node D , or some intermediate node X_1 . If node D was reached directly, it cannot have seen the PLBD with identifier β before and will receive the packet.

If node D was not in the direct neighbor set of S , then we know (according to our definition) that one of the other nodes in this set, say X_1 , will receive the PLBD and, at time τ_1 be able to retransmit it to its neighbors. These neighbors

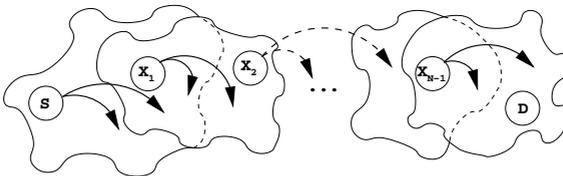


Fig. 2. Localized broadcast traversing network

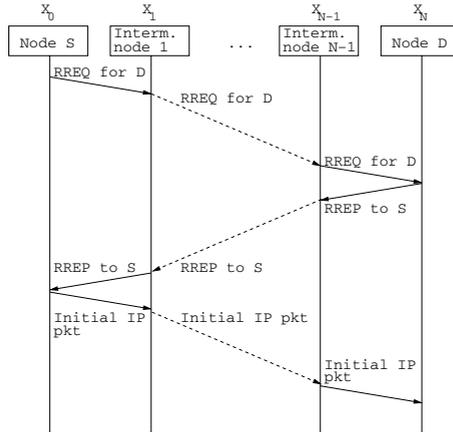


Fig. 3. LUNAR MSC with RREQ PLBD only along fastest path

may partially overlap the neighbor set of node S , but, according to Definition 2 the set will either contain node D or at least one other node, X_2 , that has not previously seen this PLBD (with id β).

Continuing in this fashion, since the network is finite, we will eventually have transmitted to the final connected node(s) that had not yet heard the PLBD with identifier β . According to our definition, node D will then also be among the nodes that have received this PLBD.

Definition 3. Fastest path

A path ξ between two nodes S and D is faster than another path ρ at some point in time, if, at this time it is possible to deliver a packet from S to D faster along path ξ than along path ρ . A path χ is the fastest path between two nodes at some point in time, if, at this time it is faster than all other paths between the two nodes.

Theorem 2. Uniqueness of a PLBD path

If there at one point in time exists at least one PLBD path, from one network node to another one, then during the same time there must exist exactly one PLBD path.

Proof. Because the number of loop free paths in a finite graph is finite, the number of paths between two nodes, S and D , in a finite network is also finite. Then, if packets are sent from node S along all (disjoint) paths, one of them will be the first to reach node D , namely the one sent along the fastest path. The fastest path will also be the unique path, since node D will henceforth disregard all PLBD packets it receives containing that particular identifier.

Thus, to recapitulate, PLBD:s can effectively be studied as propagating unicasts. We illustrate this for LUNAR in Figure 3, describing the protocol with the help of a message sequence chart. We can see that, for the case of LUNAR,

it is completely packet driven and in essence only reacts to incoming packets by updating internal tables and generating a new unicast or PLBD packet. If we always study the fastest path for every PLBD we cannot get any interference from other copies of the same broadcast packet since these will be dropped everywhere but along the fastest path according to our assumptions. Nodes that are not on the fastest path will therefore not be part of a chain forwarding the packet all the way to the intended destination node. Thus, we have fully motivated our packet transformation model and can go on to describing the model itself. What we essentially do is to reduce it from a parallel to a sequential one whereby complexity is significantly reduced.

4 Modeling Approach

4.1 The UPPAAL LUNAR Model and Verification Properties

UPPAAL [3] is a tool for simulation and verification of systems modeled as timed automata. Verification properties are passed to the system as Linear Temporal Logic (LTL) formulae. We have chosen to use UPPAAL in our work because of its powerful model checking capabilities and because we can use time in our models in a straightforward way. This further enables us to extract time bounds for route formation and initial IP packet delivery.

The LUNAR timed automata model includes a template (`lunar_message`) which models a packet in transit between two nodes. We use a system definition with three processes, representing the initial route discovery and two retries. Thus, a delay is passed as a parameter in order to model the timeout triggered resend. As in previous work we do not model any expanding ring search, but use a timeout value of 75 ms corresponding to three times the ring timeout in current LUNAR implementations and settle for two resends. Time only passes when messages are transmitted between network nodes, and we have used a range of [2,4] ms to model this delay. This represents four to eight times the theoretical delay lower limit (DLL) for the basic access mechanism of IEEE 802.11b [7]. Note that intermittent transmission failures on lower layers (e.g. due to packet collisions) are treated as link breakages in our model. In addition to the general assumptions in Section 2.4 we assume route caches to be initially empty.

Our model is to some extent less abstract than in previous work since we now model the selector tables explicitly. This is done through arrays (since there are no more complex data structures available), but it is still feasible since we gain state space usage from the PLBD abstraction. When a packet arrives at a node it needs to be switched so as to use new selectors. These are modeled using global arrays that for each node (MAC address) map selector value to a (MAC address, selector) pair.

Along the path of a PLBD, symbolic addresses of the intermediate nodes are generated as we go. These can be seen as pointers to the real addresses. Therefore, we select them from a limited range of numbers, e.g. [0,8] if we admit a maximum of 9 intermediate nodes along the fastest path. For each new route request the symbolic addresses are selected from different ranges, even though

they may in reality point to the same node. Errors due to subtle faults in the algorithm that allocates selectors might elude our analysis as a result of this assumption.

We choose to verify deadlock freedom as well as route formation and initial IP packet delivery. These are verified by checking that we can eventually get to the `snd_node_rec_lunar_rrep` (sender node received LUNAR RREP) and `message_del` (IP packet delivered) states along all execution paths. To extract the time bounds, a global timeout is used and experimentally tuned for the upper range. For the lower range we instead use LTL formulae to check possibility for route formation and initial IP packet delivery along at least one execution path.

4.2 Correspondence Between Scenarios

Instead of specifying each individual scenario exactly, we are now able to parameterize on the following:

- Maximum network diameter (number of hops), d_{max} . The maximum number of possible intermediate nodes on the unique PLBD path between source and destination, plus one.
- Number of possible link failures, f , during playout of the scenario. Note that these represent critical link failures in the sense that we model them by dropping a packet nondeterministically along the fastest path.
- (Minimum network diameter, $d_{min} \leq d_{max}$; but this value should be set to 1 to allow for all possibilities of communicating nodes' positions. The only time we use a different value is when checking correspondence to previous scenarios where positions of source and destination nodes were specified.)

The scenarios we can study using the new model encompass all the ones in our previous work (shown in Figure 1). Our definition does not require the routing protocol to find a route (or send an initial IP packet along the route) if all paths are broken. We can include link breaks if we make sure that the protocol is given the chance to find a path along some other route, in accordance with the requirement of Definition 2. The inclusion of link breaks is important in order to verify that the protocol copes with those situations as well, in the case of LUNAR by initiating another route discovery after a timeout.

Scenario (g) in previous work corresponds to setting up our new model with minimum and maximum path lengths of three and with one possible link break. This is because in scenario (g) there is one link break that can occur at any time. The minimum and maximum path lengths are three, both before and after the link break. As an illustration see Figures 4 and 5 which show two possibilities for the packet traversal. Here, solid lines denote packets that are delivered, and dotted lines denote packets that are dropped because the receiver has already seen that particular PLBD identifier. Other traversals are possible and node E may go down at other times, but because of the dampening, it should be quite clear that the maximum path length will be three regardless of the order in which packets are delivered.

We validate that this is correct with our new model by extracting a bound on initial message delivery time, which is [18,111] just as in our previous work.

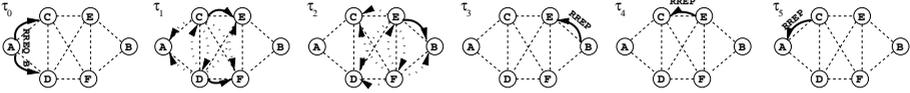


Fig. 4. Stepwise traversal of scenario (g) - Route setup initiated before link break

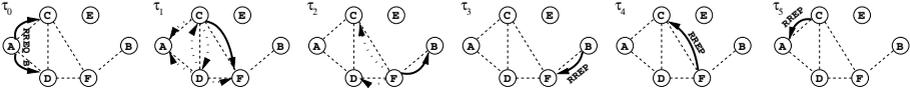


Fig. 5. Stepwise traversal of scenario (g) - Route setup initiated after link break

Using the same reasoning, we can easily translate all the previous scenarios to parameters in our new model.

5 Verification Results and Analysis

We have performed verification of LUNAR networks for the properties of interest (see Section 4.1). For general networks, i.e. where $d_{min} = 1$, we are able to verify route setup up to a diameter (d_{max}) of eleven hops, when using $f = 1$. For the same value of f we can verify initial IP packet delivery using $d_{max} = 8$ before running out of memory. This greatly surpasses the network size for which LUNAR is meant to operate (3 hops), this limit being due to the so called ad hoc network horizon [1]. Each verification takes less than a few minutes on a Macintosh PowerBook G4 laptop computer with a 1.33 GHz processor and 1.25 GB of memory. We also include some measurements from using the same processing power and verification software configuration as in our previous work. These data are presented in Table 1 together with our previous results to illustrate how substantial the performance increase is.

Due to space constraints, we are here only able to include one of our result plots. Figure 6 shows bounds on route formation and initial IP packet delivery times for the case when $d_{min} = d_{max}$ and $f = 1$. The same results as in corresponding scenarios of our previous work are obtained. As mentioned, we can also

Table 1. Comparison of UPPAAL verification results

Scenario	Explicit broadcast modeling			Using broadcast abstraction		
	States searched	Time used	Search completed	States searched	Time used	Search completed
(a)	15072	3.89 s	Yes	487	< 1 s	Yes
(e)	123196	57.91 s	Yes	487	< 1 s	Yes
(g)	2.01e+06	11:43 min:s	Yes	910	< 1 s	Yes
(h)	2.97e+07	1:59 h:min	No	910	< 1 s	Yes

verify the more general case where $d_{min} = 1$, and the difference in time bounds is that they then reach down to 4 ms for the route setup time and to 6 ms for the initial message delivery. The reason is that the most extreme case then is when source and destination are in direct contact, whereby the route setup can be completed in two transmissions. With this setting of d_{min} the verification also includes all intermediate situations, which increases the complexity.

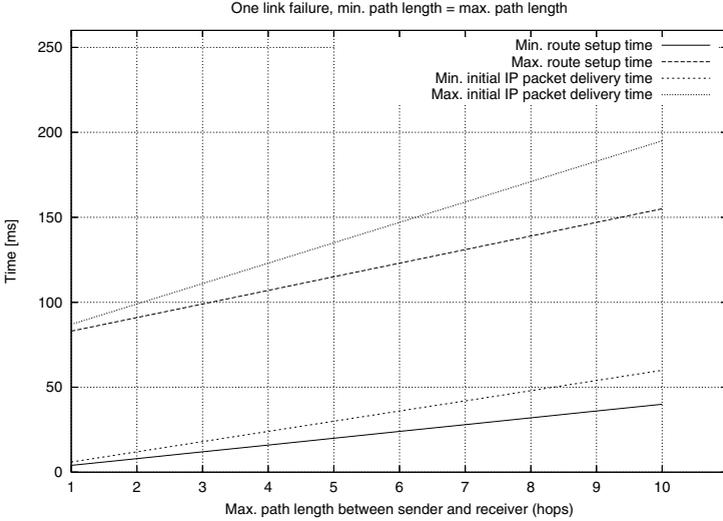


Fig. 6. Example plot showing time ranges extracted

Thus, since we are able to use a rather high number for the network diameter we are in fact able to study all networks of practical significance even though we have not produced a general proof. Our results are general in the sense that any mobility model can be accommodated. We only require that it, at some point, yields a network configuration with a unidirectional path between a given pair of nodes. Link breakages affect the initial state of route caches in the network. It is therefore important to study if different settings for f cause the protocol to behave differently. When increasing f from 1 to 2, and using $d_{min} = 1$, we can verify route setup up to $d_{max} = 8$ and initial IP packet delivery up to $d_{max} = 6$. Introducing more link breaks thus reduces the maximum network diameter that can be used. Due to the routing protocol structure of LUNAR its worst case behavior is captured by admitting the same number of link breaks as resends. The protocol cannot be expected to guarantee successful route discovery if more link breaks occur, since all retries may then be lost. Given that we only model two LUNAR resends (see Section 4.1) we therefore choose to set the limit at $f \leq 2$ in this study.

We can perform verification of all the scenarios studied in previous work, but considerably faster. Furthermore, we are able to study more general network configurations of much greater size. In a related project we are currently performing

real world experimental evaluation of a number of ad hoc routing protocols. There, scenarios with a maximum of four nodes are used and even then, we note trouble in forming multi-hop paths which causes severe performance penalties for TCP [8].

6 Comparison of Route Discovery in DSR and LUNAR

DSR [2] and LUNAR can both be considered as on-demand protocols because neither of them relies on any kind of periodic packets to be exchanged between nodes. In the basic route discovery phase, the two protocols operate in a similar way. However, there are some important differences:

- DSR is a source route protocol which means that the RREQ packet includes addresses of all the nodes passed along the path from source to destination. This list is then returned to the source node and used as header for each IP packet that is subsequently to be routed. At each step, nodes use the next address in the header as new destination. In the case of LUNAR, label switching is instead used in nodes for the rewriting of addresses.
- LUNAR only stores the first response received from a route discovery. In DSR, on the other hand, a node may learn and cache multiple routes to any destination. This is also possible through overhearing routing information from packets sent by others as opposed to in LUNAR where nodes only use information they have themselves requested.
- In DSR, nodes which are not themselves the sought destination may answer with one of their routes. The answer will contain the list of addresses traversed thus far concatenated with the cached route. Loop segments are identified and removed, and a node cannot return a route in which it is not itself included.

In a DSR model we need to account for these differences properly. Instead of one destination for the PLBD, we need to study a set of answering nodes, \mathcal{D} . As an upper bound for this set we have the number of disjoint paths originating from network node S . We can, however, settle for all those that reach a neighbor of D , since the others will not be valid at the time of the RREQ. In the DSR draft it is said that the number of hops will often be small (e.g. perhaps 5 or 10 hops). It is also stated that the DSR protocol is designed mainly for mobile ad hoc networks of up to about 200 nodes. In a finite network, the set of answering nodes is also finite. The maximum value will be the total number of nodes in the network minus one (the sending node). This case appears if S and D are directly connected and all other nodes are also connected to both the source and destination. The implication for verification is, however, not as severe as it may first seem since no path can then be longer than two hops.

7 Related Work

Chiyangwa and Kwiatkowska [9] have studied timing properties of AODV [10] using UPPAAL. Their model uses a linear topology with specialized sender, re-

ceiver and intermediate nodes. The authors investigate how network diameter affects the protocol. They report that at 12 intermediate nodes, the recommended setting for route lifetime starts to prevent long routes from being discovered. They propose adaptive selection of this parameter to compensate for the behavior in large networks. This work is related to ours, but the linear scenario type contains a static number of nodes and its motivation is to discover a maximum diameter. Apart from providing a formal motivation to a single network path, our methodology encompasses a variety of topologies. Their method involves constructing a specialized model where we use the same protocol instance at each node which simplifies the modeling process.

Obradovic et al [11] have used the SPIN [12] model checker and the HOL [13] theorem prover to verify route validity and freedom from routing loops in AODV. They used conditions on next node pointers, sequence numbers and hop counters to form a path invariant on pairs of nodes (on the path from source to destination). Three lemmas were then verified using SPIN after which HOL was used to prove that the three lemmas imply the path invariant theorem (using standard deductive reasoning). The approach requires a significant amount of user interaction and is not directly applicable to other protocols.

Das and Dill [14] also prove absence of routing loops in a simplified version of AODV. The strategy is similar to that of Obradovic et al, but more automated. They use predicate abstraction and can discover most of the quantified predicates automatically by analyzing spurious abstract counter-example traces, albeit with some mechanical human involvement. The initial predicate set is formulated in a manual step where conditions on next node pointers, hop counters, and existence of routes are constructed. The method successfully discovers all required predicates for the version of AODV considered. Proficiency in formal verification is required in order to make general use of their method.

de Renesse and Aghvami [15] have used SPIN to model check the ad hoc routing protocol WARP. They use a general 5-node topology, and provide a non-exhaustive verification (using the approximating bitstate hashing mode [12]), covering 98% of the state space.

Xiong et al [16,17] have modeled AODV using Petri nets. A topology approximation mechanism describes dynamic topology changes. They report on a looping situation found during a state space search of a general ten node topology. Their broadcast model uses an average number of messages based on the average degree and the total number of nodes in the graph. The resulting PLBD implementation is less abstract than ours and models redundant packet transfers between nodes not on the fastest path between the sender and receiver. In contrast to our approach link failure effects are also not included in their model as they assume unicast transmissions to be globally receivable regardless of topology.

With our method we can use the same protocol instance for each symbolic node and easily verify high level properties of ad hoc routing protocols, such as “initial IP packet delivered”. We do not put strict requirements on the topologies, but admit for general networks of a certain diameter and a given number of link

breakages. The modeling and verification processes are thus quite simple and applicable to a range of different protocols.

8 Conclusions and Future Work

We have developed a new efficient method for modeling PLBD, one of the operations in ad hoc routing protocols most responsible for state space explosion in previous approaches. We applied the technique, verifying the operation of route establishment in the ad hoc protocol LUNAR, and derived upper and lower time bounds for both route establishment and first packet delivery over the resulting route. We verified route setup in networks of up to eleven hops in diameter, well over the envisioned upper limit for practical application of ad hoc routing in realistic scenarios.

For our future work we intend to perform the same verification for the DSR protocol, which we have only sketched here. We want to see if there are other protocols that utilize primitive operations, responsible for much of the complexity, which can be abstracted away from in order to enable for verification in realistic networks. These analyses can further be used to compare the quality of competing protocols.

References

1. Tschudin, C., Gold, R., Rensfelt, O., Wibling, O.: LUNAR: a lightweight underlay network ad-hoc routing protocol and implementation. In: Proc. Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN). (2004)
2. Johnson, D.B., Maltz, D.A., Hu, Y.C.: Internet draft: The dynamic source routing protocol for mobile ad hoc networks (DSR). <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt> (2004)
3. Larsen, K.G., Petterson, P., Yi, W.: UPPAAL in a Nutshell. *Int. Journal on Software Tools for Technology Transfer* **1** (1997) 134–152
4. Wibling, O., Parrow, J., Pears, A.: Automatized verification of ad hoc routing protocols. In: Proc. 24th IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FORTE). (2004)
5. Ni, S.Y., Tseng, Y.C., Chen, Y.S., Sheu, J.P.: The broadcast storm problem in a mobile ad hoc network. In: Proc. ACM MobiCom. (1999) 151–162
6. Wibling, O.: LUNAR pseudo code description. http://user.it.uu.se/~oskarw/lunar_pseudo_code/ (2005)
7. Xiao, Y., Rosdahl, J.: Throughput and delay limits of IEEE 802.11. *IEEE Communications Letters* **6** (2002) 355–357
8. Lundgren, H.: Implementation and Experimental Evaluation of Wireless Ad Hoc Routing Protocols. Phd thesis, Uppsala University (2005)
9. Chiyangwa, S., Kwiatkowska, M.: A timing analysis of AODV. In: Proc. Formal Methods for Open Object-Based Distributed Systems: 7th IFIP WG 6.1 International Conference (FMOODS). (2005)
10. Perkins, C.E., Royer, E.M.: Ad hoc on-demand distance vector routing. In: Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications. (1999) 90–100

11. Obradovic, D.: Formal Analysis of Routing Protocols. Phd thesis, University of Pennsylvania (2002)
12. Holzmann, G.: The Spin Model Checker, Primer and Reference Manual. Addison-Wesley, Reading, Massachusetts (2003)
13. University of Cambridge Computer Laboratory: Automated reasoning group HOL page. <http://www.cl.cam.ac.uk/Research/HVG/HOL/> (2005)
14. Das, S., Dill, D.L.: Counter-example based predicate discovery in predicate abstraction. In: Formal Methods in Computer-Aided Design, Springer-Verlag (2002)
15. de Renesse, R., Aghvami, A.: Formal verification of ad-hoc routing protocols using SPIN model checker. In: 12th Mediterranean Electrotechnical Conference (IEEE MELECON). (2004)
16. Xiong, C., Murata, T., Tsai, J.: Modeling and simulation of routing protocol for mobile ad hoc networks using colored Petri nets. In: Proc. Workshop on Formal Methods Applied to Defence Systems in Formal Methods in Software Engineering and Defence Systems. (2002)
17. Xiong, C., Murata, T., Leigh, J.: An approach to verifying routing protocols in mobile ad hoc networks using Petri nets. In: Proc. IEEE 6th CAS Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication. (2004)