Worst Case Performance Analysis of Wireless Ad-hoc Routing Protocols: Case Studies

Yan Yu, Deborah Estrin, Sandeep Gupta

Computer Science Department, ISI, Electrical Engineering Department USC

Email: yanyu@usc.edu, estrin@isi.edu, sandeep@poisson.usc.edu

04/15/2000

Abstract

Wireless communication poses challenges for network protocol analysis, such as high channel error rate and topology dynamics due to mobility. Motivated by mission critical applications, we present an approach to systematically analyze and test the performance of wireless ad-hoc routing protocols under worst case scenarios. Because of some important energy-stringent applications, we targeted the network partition problem that results from some nodes being depleted of power. The corresponding worst case scenario is the one that causes network partition with the least host traffic. We obtained the worst case scenario using heuristic search techniques combined with an extended finite state machine (FSM) model to represent the wireless routing protocol. Simulation showed that much less traffic is required to bisect the network for our "worst" case scenario than for any one of a large number of random scenarios. Furthermore, the simulation result under the "worst" case scenario was similar to that obtained in our theoretical analysis. Finally, two case studies for AODV and DSR (namely, STRESS simulation of AODV vs. DSR, and a modified version of DSR called Energy aware DSR) are presented to illustrate how worst case scenario analysis may provide new insight into protocol evaluation and design tradeoffs.

1 Introduction

When there is no fixed infrastructure available for communications, wireless ad-hoc networks can be quickly formed to facilitate communication. Despite their convenience, wireless networks are more likely to fail than wired networks, and their performance is not as effective as wired networks. For many generic IP based applications, ad-hoc networks provide sufficient and convenient solutions. However, some mission critical applications (e.g., disaster recovery) enabled by wireless network have more stringent requirements than that provided by "best-effort" networks. Motivated by these mission critical applications, we studied the performance of ad-hoc routing protocols under worst case scenarios. In particular, we studied the conditions causing the most rapid network partition.

There has been some performance comparison of wireless ad-hoc routing protocols based on simulation results [5, 10, 6]. However, previous results were all based on the simulation of some random scenarios, except [6], which also considered three hopefully more realistic scenarios: conference scenarios, event coverage scenarios, and disaster area scenarios, where the simulation parameters were selected to represent the specific type of application. While the performance of a particular protocol may change dramatically under different operating conditions, it is prohibitively complex to explore the space of all possible operating conditions. A simulation study may consistently execute under the conditions where the protocol functions effectively, but may ignore (or be unaware of) other conditions under which the protocol functions poorly. It is unlikely to generate the worst case scenario by simulating randomly generated scenarios. Therefore, a systematic approach would be beneficial to test or verify wireless ad-hoc routing protocols. We are unaware of any previous systematic approach or analytical study of wireless ad-hoc routing protocols.

Although not all wireless networks are energy constrained, there are some important classes of problems that have stringent energy requirements (e.g., environmental monitoring, disaster recovery, or battle-field surveillance), where an unattended untethered sensor net is often deployed. In such applications, energy becomes the bottleneck resource. Such applications will also benefit from the type of methodologies presented in this paper: examining conditions under which the network partitions the sensory data sources due to energy depletion. In our ad hoc routing context, the corresponding worst case scenario is defined as the least host traffic that causes network partition.

Systematic Testing of Robustness by Evaluation of Synthesized Scenarios (STRESS [1]) provided a systematic framework to verify the correctness or performance claims of the network protocols. The previous research on STRESS [1, 4, 13] was focused on correctness or performance analysis of multicast routing protocols over LANs (e.g., Protocol Independent Multicast (PIM) and Scalable and Reliable Multicast (SRM) protocol).

Our current study applies the STRESS techniques to wireless ad-hoc routing protocols. Being more susceptible to the effects of interference and environmental factors, wireless transmission provides a less reliable platform for network protocols than that of wired transmission. In addition, the movement of mobile nodes not only makes network topology more dynamic than in a wired network, but also exacerbates the above problems. Furthermore, a tighter coupling between lower level properties and higher level protocols exists in wireless network, such as the relation between power consumption and the number of messages transmitted. The power level and the amount of the traffic transmitted also affect the channel error rate. The above challenges complicate the analysis of wireless and mobile network protocols, and make analyzing and testing wireless routing protocols systematically an even more complex task. This paper presents a preliminary study that has been explored to achieve this goal.

Through automatically synthesizing the worst case scenarios that target extreme conditions, we intend to provide insight into some protocol design and deployment issues, such as topology (e.g., density) deployment, pathological traffic patterns to be avoided, and protocol's tolerance to node mobility. The research presented in this paper sheds light on some of these problems in wireless ad-hoc routing protocols.

To generate the worst case scenarios for targeting extreme conditions, first, the network bisection (due to nodes being depleted of energy) problem is identified as the target extreme condition. Then the worst case scenario for this extreme condition is automatically generated, which can be used as input to a network simulator. Both the worst case scenarios and some random scenarios are simulated to confirm our analytical analysis. The simulation result from the above "worst" case scenario is similar to the worst case in the theoretical analysis, and the generated worst case scenario is much worse than randomly generated scenarios, which are often used in conventional protocol evaluation.

To give an example of how the STRESS worst case analysis of the wireless ad-hoc routing protocols can provide new insight into the protocol design and deployment, we present two case studies. First, simulation data from Ad-hoc On-Demand Distance Vector protocol (AODV [9]) under the worst case scenarios again confirmed the STRESS worst case analysis in addition to the above simulations for DSR. Furthermore, based on the simulation of the worst case scenarios, AODV has lower or similar control overhead than Dynamic Source Routing Protocol (DSR) [7] in most scenarios. This is in contrast with previous studies comparing these two protocols [5, 10], in which DSR has lower control overhead than AODV. Second, in order to study the impact of the STRESS worst case analysis on the energy aware ad-hoc routing protocols, DSR was modified to make route selection decisions based on energy information intead of hop count. Our simulation studies indicated that for the worst case scenario, the modified version of DSR, that we call Energy aware DSR (EDSR), delivered 15% more host packets before network partition. However, for random scenarios, the same version of EDSR only delivered 2 - 3% more host packets on average. Therefore, STRESS analysis provides a different perspective on protocol efficiency evaluation from random scenario simulation.

The remainder of this paper is organized as follows. In Section 2, the algorithm (i.e., STRESS algorithm) to automatically generate the worst case scenarios for the target network bisection problem is presented. For convenience, the worst case scenario generated by the STRESS algorithm is referred to as STRESS scenario. In Section 3, the simulation results of the above-mentioned worst case scenarios for DSR is presented to confirm the above worst case analysis. In Section 4, two case studies were presented to show how STRESS worst case analysis provides a different point of view into the protocol performance eveluation from random-scenario simulation. Some discussion and future work was presented in Section 4.3. Finally we draw conclusions in Section 5.

2 The STRESS algorithm for generating the worst case scenario causing network bisection

As mentioned previously, justified by the energy stringent applications, the network bisection (due to nodes being depleted of energy) problem is identified as the target extreme condition. The corresponding worst case scenario is identified as that requiring the least host traffic to cause network partition. An algorithm to automatically synthesize worst case scenarios for the target extreme conditions is presented in this section.

The key heuristic of the algorithm is to force the traffic to focus on the nodes along the bisection line so as to cause network bisection as rapidly as possible.

In addition, we are trying to minimize energy consumption by other nodes in order to deplete the power of the nodes along the bisection line, such that without the network bisection, the network can still function as normal to make the worst case scenario more pathological.

In this section, the underlying model and the class of protocols being addressed by the algorithm are specified, then the details of the algorithm are presented.

2.1 Underlying model

Our initial model assumes a random topology with symmetric links and no node mobility. The network partition occurs if all the nodes on a bisection line are depleted of power. A **bisection line** is a sequence of nodes such that when all of them are depleted of power, the network partitions. The bisection line has a local minimum property, i.e., any true subset of it will not cause network partition.

The underlying energy model of each node is the following: Each node has an initial power level, sending each packet consumes some power P_s, and receiving each packet consumes some power P_r. Therefore, after sending and receiving some packets, when the power level decreases below some threshold, the node will lose communication capability (i.e., be declared dead). Therefore, each node can only send or receive a maximum number of packets in its lifetime, P_l.

The underlying wireless ad-hoc routing protocol is assumed to use cached routing information for route reply in its route discovery mechanism. It is a common mechanism used by several wireless ad-hoc routing protocols, such as DSR and AODV.

Our initial model ignores energy consumed by the control messages, which include MAC/ARP/Routing layer control packets. Based on the simulations, MAC layer RTS/CTS hand shaking packets dominate the total control packets, and the number of MAC layer control packets is approximately proportional to the number of unicast data packets. Note that many routing packets do not need RTS/CTS. For example, in DSR, no RTS/CTS is necessary for broadcasted route request, only unicast route reply/error need RTS/CTS among all the routing control message. Therefore, we can estimate the amount of control packets by considering data packets being transmitted at each node especially when we initially work on a static network, and there is little network dynamics except that some nodes may be drained of energy in the end. However, control packets are an important secondary factor that we will include in our future model, and it is especially necessary when we consider higher network dynamics, such as node mobility.

2.1.1 Theoretical worst case analysis

In addition to the above-mentioned energy model, we assume that the number of packets sent is the same as the number of packets received for each intermediate forwarding node, and each data packet is of fixed length. If we ignore the energy consumed by control packets, each node can only forward (receive and transmit) P_l packets in its lifetime, where $P_l = \frac{E_i}{(P_s+P_r)*L_p/bw}$. (Initial node energy level is E_i, sending and receiving packets consumes P_s/sec, P_r/sec respectively. Packet length is L_p, bandwidth is **bw**.)

2.2 Algorithm

The input of this algorithm is the network topology and the given bisection line. (Some node separator algorithm can be used to identify the bisection line for a given network topology [11, 3].) The output of the algorithm is the traffic pattern in the form of a list of source and destination pairs. The position of the element in this list reflects the order in which the source issues the traffic request. The objective of this algorithm is to identify the worst case traffic pattern (with minimum host traffic) that will cause network partition. Therefore, the key heuristic used in the algorithm is to **force the traffic to focus on the nodes along the bisection line**.

The whole procedure is composed of two phases: the first phase discovers and builds up the route through a route discovery phase; next, in the steady traffic phase, the communication between the two far ends of the bisection line is used to drain the power of the nodes along the bisection line.

2.2.1 Algorithm Details

The algorithm takes the bisection line sequence as an input, and outputs the traffic pattern that will cause network bisection (in the form of source nodes list).

The algorithm operates as follows:

1. In the route discovery phase, the algorithm:
   (a) Picks the destination node, **dst**, such that it is directly reachable from the first node of the bisection line, but is not on the bisection line.
   (b) Builds up the route between the consecutive nodes on the bisection line:
      - Begins from the first node, checks each node on the bisection line in order.
      - If the next node y on the bisection line is directly reachable from the current node x, includes y to the source nodes list. Otherwise, discovers the shortest path between x and y (using breadth-first search). Includes the nodes on this path into the source nodes list.

   (c) After reaching the end of the bisection line, expands one hop away, includes the neighbor of the last node on the bisection line into the source nodes list.
2. In route construction phase, each node in the source nodes list issues one communication request in the order that it appears on the list. The interval between consecutive communication requests should be long enough to build up the route cache in each intermediate node.
3. In steady state, the traffic only involves the destination and the last node of the source nodes list, as

shown in Figure 1. Such communication continues until the network partition occurs or the source node is depleted of power.

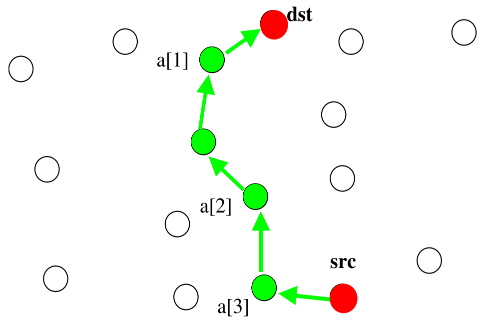

Figure 1: steady state traffic generated by STRESS scenario

The heuristic in the above algorithm (i.e., force the traffic to focus on the nodes along the bisection line) successfully reduced the search space. The complexity of the STRESS algorithm with the help of heuristic is O(m), where m is the number of nodes on the bisection line if each bisection line node is reachable from its successor. Otherwise, the complexity is a bit higher, where a little extra time is spent on discovering the shortest path between the current node and the next one on the bisection line. In contrast, the complexity of unconstrained search is much higher, $C(n, 2)^k$, where n is the number of nodes in the network, k is the number of node pairs that participate in the communication. Each communication is initiated sequentially, and the communication continues until network partitions.

3 Simulation of the STRESS scenarios for DSR

3.1 Objective of the simulation

The main objective of the simulation is to test our algorithm, showing that although the above STRESS algorithm is based on some simplifying assumptions (e.g., the initial model does not consider control packets), the simplified model still provides useful insight.

Through simulation, we intend to test the following hypothesis:

1. The simulation result from the STRESS worst case scenario is similar to the worst case in the theoretical analysis.

2. The STRESS worst case scenario is significantly more severe than randomly generated scenarios.

3.2 Simulation setting

We simulated the STRESS scenario in a network simulator ns-2 [14, 2]. The simulation data presented in Figures 2-4 is for a 5 x 5 grid network with uniformly distributed nodes. The transmission range of each node is the distance between a node and its horizontal or vertical neighbor. The initial energy level at each node is $E_i = 0.05$ Joules. The energy consumed by sending/receiving a packet, $P_s = P_r = 281$mw. CBR packet length is about 600 bytes, including 512B data and packet header. The bandwidth is 2*1e6 b/s. Therefore, based on the analysis in Section 2.1.1, a node can only forward (receive & send) about **36** data packets in its lifetime if the energy consumed by control packets is ignored. Thus, the real lower bound of the number of host packets required to cause network partition is less than **36**, but should be close to **36** if the energy consumed by control packets is not a significant portion of the total consumed energy. Otherwise, if there are lots of control packets, the lower bound of the number of host packets required to cause network bisection is much less than **36**.

We simulated the following types of scenarios:

1. STRESS scenario – the worst case scenario generated by the STRESS algorithm described in section 2.2.

2. Randomly generated scenarios – the topology is the same as in (1). There are two subtypes of random scenarios:

   (a) Random scenarios with the fixed destination and randomly selected sources. The fixed destination may represent some information sink in the network. The fixed destination and the number of source nodes are the same as in scenario (1). The randomly selected source nodes initiated the communication with the fixed destination in a randomly selected order. The random scenarios with the fixed destination were included for fairness of comparison since the fixed destination may concentrate the traffic, and in turn will lead to early network partition.

(b) Random scenarios where the source and the destination are both randomly selected. The number of (source, destination) pairs is the same as in scenario (1).

3.3 Evaluation metrics

The STRESS algorithm attempts to find the worst traffic pattern with minimum host traffic that will cause the target network bisection. Besides host traffic, there are other factors and metrics with regard to the network partition problem. For example, energy is often a major constraint in some wireless and mobile network applications. Therefore, in addition to the primary metric (i.e., the amount of host traffic required to cause network partition), other energy-related metrics are proposed as follows:

The simulations are evaluated against the following three metrics:

1. Total number of host packets required to partition the network. (This was the primary objective function for the above STRESS algorithm.)

2. Total energy consumed from the network to cause network partition. (Determining the worst traffic pattern that will cause network partition with least amount of energy consumed from the network is a secondary objective.)

3. Variance in the remained energy of the nodes. More balanced energy consumption may prevent some nodes from being depleted more rapidly than other nodes, thus prevent early network bisection.

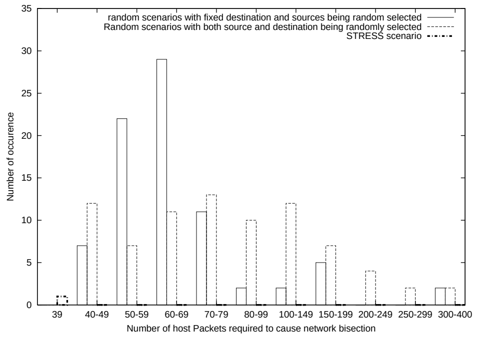

Figure 2: Number of host packets required to cause network partition

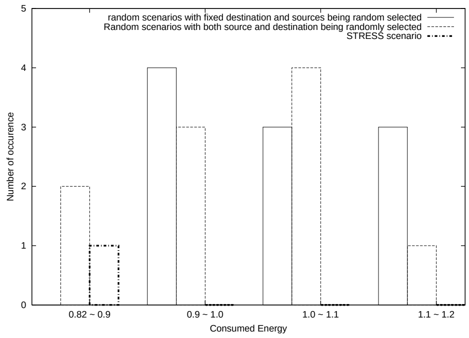

Figure 3: Total energy consumed from the network to cause network partition

3.4 Simulation result for DSR

3.4.1 Number of host packets required to cause network bisection

For a 5 x 5 grid network, as shown in Figure 2 (each data point for random scenarios in Figure 2 is averaged over 10 simulation runs), the number of host packets sent out in the STRESS worst case scenario is only 39. For the same topology, under random scenarios, where the source is randomly selected or both the source and the destination are randomly selected, it requires 12.8% - 869% more host packets to cause network bisection.

3.4.2 Total energy consumed from the network to cause network partition

As shown in Figure 3, the STRESS scenario requires up to 39.8% less energy from the network to cause network partition. Note: the STRESS algorithm tries to minimize the number of host packets to cause network partition, not necessarily the least amount of energy. However, as a side effect, the STRESS scenario requires less energy than most random scenarios, which made the STRESS scenario more pathological, since without network partition, the network would have functioned as normal for a longer duration.

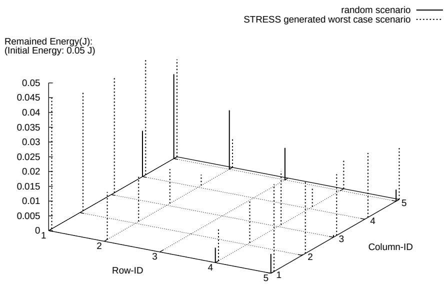

Figure 4: Energy remained at each node after network partition

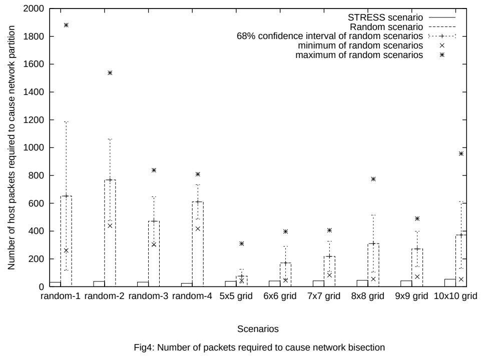

Figure 5: Much less host packets required to cause network partition for STRESS scenario vs. random scenarios

3.4.3 Variance in the remained energy of the nodes

Figure 4 informally illustrated the relation between the variance of the remained energy at each node after network bisection and the early network partition problem.

As shown in Figure 4, in STRESS scenario, the energy consumption is concentrated on the bisection line. After the bisection line nodes are depleted, there is still plenty of energy left in the rest of the network. However, the communication among the other nodes can no longer pass through the bisection line because of network partition. On the other hand, for most random scenarios, when the non-trivial network partition occurs, the remaining functioning nodes are also almost depleted. Therefore, it does not really matter whether the network is partitioned or not.

3.5 More simulation result for larger grid network and random topology

To test the general applicability of the above analysis, we also simulated larger systems with more nodes, and some networks with randomly generated topologies.

Figure 5 and Figure 6 show the simulation results for 10 network topologies. The first four scenarios, *random-1 - random-4* are randomly generated topologies with 50 nodes in a 1500m x 500m rectangle network. The transmission range is 250m. (These numbers are typically used parameters in previous protocol performance evaluation or comparison work [5, 10, 6]). The remaining scenarios are 6 grid networks with

6x6, 7x7, 8x8, 9x9, 10x10 uniformly distributed nodes respectively. The transmission range of each node is the distance between a node and its horizontal or vertical neighbor. The STRESS scenario and random scenarios use the same topology.

Each data point for the random scenarios is the average of the results from 10 simulation runs with 10 randomly generated traffic for the same topology. (For random scenarios, the topology and the CBR traffic is generated by a random scenario generator and a traffic generator included in ns-2.)

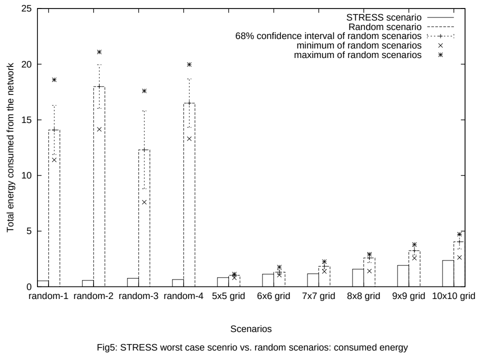

Figure 6: Less energy consumed to cause network partition for STRESS scenario vs. random scenarios

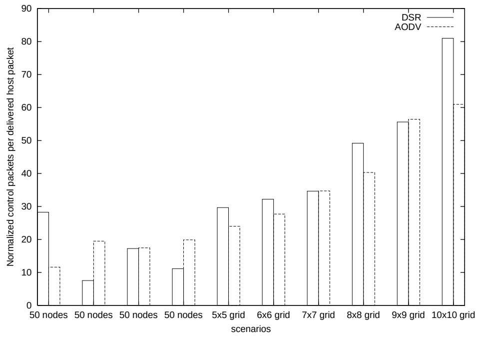

Figure 7: Protocol overhead: Normalized control packet for STRESS scenarios for DSR and AODV

Figure 5 and Figure 6 confirmed again that STRESS scenarios require 28% - 99% less host traffic than randomly generated scenarios to cause network partition. At the same time, the STRESS scenario requires up to 96% less energy from the network than most randomly generated scenarios to cause network partition.

As you may notice, there is dramatic difference between the STRESS scenario and random scenarios in a random topology. In a random topology, usually there exists a short bisection line, which is shorter than the network's diameters in other directions. (In the grid network, the length of the bisection line is the diameter of the network.) Therefore, this will lead to network bisection very early in the STRESS scenario in some random-topology networks.

3.6 Summary of the simulation results of the STRESS scenarios

1. 28% - 99% less host traffic was required to bisect the network in the STRESS generated scenario than that generated by a random generator for the same network topology and the same energy model. The STRESS algorithm was successfully used to identify much worse case scenario in terms of the number of host packets sent to cause the network partition.

2. Up to 96% less energy was consumed to bisect the network for the STRESS generated scenarios than that generated by a random generator.

3. The number of host packets required to cause network partition in the STRESS worst case scenario is very close to the result in the theoretical analysis.

   As explained in Section 3.2, for the given energy model, and the given initial energy level used in the simulation, at least 36 data packets must be received and sent to deplete a node, assuming the energy consumed by the control packets can be ignored. However, if control packets consume significant amount of energy, the number of packets that a node can receive and forward before it is depleted can be much less than 36. In simulation results, as shown in Figure 5, from 24 to 45 is very close to this number.

4 Case studies

STRESS analysis can help to identify the worst case scenario for the target extreme condition (e.g., the network partition due to some nodes being depleted of energy). This may provide some insight into the worst case scenario to be avoided, or what might be the worst case performance for a given protocol. To give some concrete examples of how the above analysis of the wireless ad-hoc routing protocols may provide insight into the protocol design and deployment, we presented two case studies on AODV and DSR: (1) DSR vs. AODV, and (2) DSR vs. Energy aware DSR.

4.1 DSR vs. AODV

The simulation data of the STRESS scenarios for AODV is similar to that of DSR in terms of the total number of host packets sent to cause network partition (as shown in Table 1). This again confirms that STRESS analysis helps to identify the worst case scenario for the network partition problem, and the algorithm discussed in Section 2 should be applicable to the class of wireless ad-hoc routing protocols utilizing route cache information and in which the route selection is based on the shortest hop count. Furthermore, as shown in Table 1, although AODV requires almost the same amount of host traffic to bisect the network as DSR (36 vs. 39), it requires less control overhead than DSR, both in routing packets and in total control packets, which also include MAC layer control packets. Therefore, AODV has more energy left after network bisection.

Table 1: Simulation results for STRESS scenario for 5x5 grid network for DSR and AODV

	STRESS scenario for DSR	STRESS scenario for AODV
Total number of host packets sent	39	36
Total energy consumed to cause network bisection (J)	0.820605	0.698654
Normalized energy consumed per delivered packet (J)	0.023446	0.023247
Normalized control packets generated per delivered host packet	29.65	24

For AODV, again more scenarios on larger systems and random topology network were simulated. The simulation result is shown in Figure 7, where the same scenarios were used as those in Figure 5 (see section 3.5).

As shown in Figure 7, AODV requires similar or lower normalized control overhead than DSR in most STRESS scenarios except two: scenario 2 and 4. Therefore it has more energy left after network bisection, and in turn has more chance to communicate among the subset even after network partition. However, in previous performance comparisons of DSR and AODV [5, 10, 6], AODV typically requires more control overhead. That observation was attributed to the fact that DSR aggressively utilizes route cache and promiscuous listening such that it suppresses propagating route request in many circumstances. Therefore, the analysis of the worst case performance may expose some important scenarios that may be missed by random-scenario simulation. It provides a different point of view on protocol efficiency in terms of control overhead in this case study. Hence, although we can not draw any conclusion from the worst case study, worst case study can provide useful complementary information and may provide network operators some insight when selecting a wireless ad-hoc routing protocol.

4.2 Energy aware DSR

In order to study the impact of the STRESS worst case analysis on ad-hoc routing protocol evaluation, DSR was modified to make route selection decisions based on energy information instead of hop counts.

4.2.1 Energy aware DSR

There has been a fair amount of work on alternative wireless ad-hoc routing protocols to address energy efficiency problem. [8] proposed that the routing decision be based on shortest cost in terms of different power aware metrics. [12] provides energy conserving routing for static, power-controlled wireless ad-hoc networks. Their algorithm is based on a prior knowledge of the traffic, i.e., the sets of the origin and destination nodes and information generation rates at the origin nodes.

STRESS worst case performance analysis may provide new insight into the design trade-offs of energy sensitive ad-hoc routing protocols. In order to study the impact of STRESS worst case analysis on energy aware ad-hoc routing protocols, we made minimal changes to DSR to make route selection decisions based on energy information rather than shortest hop count information. The energy aware metric selects the route that is composed of nodes with more remaining energy. In the route discovery phase, more route discovery is encouraged if few alternative routes are available for route selection. We call the new version of this protocol Energy-aware DSR (EDSR).

4.2.2 Simulation result for Energy Aware DSR (EDSR) for STRESS scenarios

Again ns-2 was used as the network simulator. The DSR protocol in ns-2 was modified to perform energy aware routing as mentioned above. The protocol performance for the original DSR and EDSR for the

STRESS scenarios are shown in Table 2.

As shown in Table 2, EDSR prolonged network lifetime in the STRESS scenario, i.e., delivers 15% (from 35 to 40) more host packets before network partition. It only slightly increased normalized energy consumption and control overhead.

Table 2: Simulation result for STRESS scenario for original DSR and EDSR for 5x5 grid network for DSR and AODV

	STRESS scenario for original DSR	STRESS scenario for EDSR
Number of delivered packets	35	40
Normalized energy consumption per delivered packet	0.02344	0.02588
Normalized control packets per delivered host packet	29.65	31.325

4.2.3 Simulation result for EDSR in some random scenarios

4 .2.3.1 Simulation setting:

Here, EDSR was simulated on some random scenarios. The data presented here is for a 1500m x 500m network with 50 nodes randomly distributed in it, with low mobility and moderate traffic load. The maximum speed is 1, 3, 5 m/s. The pausing time is 50, 100, 150, 200s. The number of CBR-traffic connections is 10, 15, and 20.

The simulation lasted until 50% of the nodes were depleted. The number 50% is a somewhat arbitrary threshold used to indicate that the communication is compromised because some nodes are depleted of power and thus lost the communication capability. We would like to point out that network partition was not used as a criteria to stop the simulation. With the presence of node mobility, at one moment, the network may become segregated because of some node moving out of range, but at the next moment, the network maybe no longer be bisected because of node mobility. This criteria is referred as “network ineffectiveness”. The radio model uses the characteristics similar to that of the Lucent WaveLAN card, such as with 250m transmission range.

4.2.3.2 Simulation result

The simulation result of 96 simulation runs for the above parameter setting is summarized in Table 3. The 96 simulations are the combination of 16 mobility scenarios and 6 traffic patterns.

Table 3: Original DSR vs. Energy aware DSR

	Original DSR	EDSR
Number of delivered host packets before network bisection	595.91	606.77
Increase ratio in the number of delivered packets in EDSR vs. original DSR		2.78%
Normalized energy consumption per delivered packet	0.0373	0.0365
Normalized control packets per delivered host packet	23.4669	22.5594

As shown in Table 3, for random scenarios, on average EDSR can deliver 2 - 3% more host packets before network becomes ineffective. As shown in Table 3, EDSR also slightly decreases (2.15% less) average energy consumed per delivered host packet, and average control overhead (3.86% less). This may be explained by the following fact: In shortest hop routing, some nodes may be depleted rapidly because traffic concentrates on the shortest path. Then, the packet will get dropped after reaching the early depleted node. This will trigger more control messages for route maintenance and new route discovery process, which in turn will consume more energy than under normal conditions.

4.2.3.3 Implication of the EDSR simulation results of the STRESS worst case scenario and random scenarios

The simulation result indicated that for the STRESS worst case scenario, this particular version of Energy aware DSR can deliver 15% more host traffic before network partition. However, for random scenarios, the same version of Energy aware DSR only delivers 2% - 3% more host packets on average. Thus, in the context of EDSR, the STRESS scenario again provided a different perspective on evaluating protocol performance from random scenario simulation.

4.3 Discussion and future work

The key procedure in the STRESS algorithm is to use the route discovery phase build up the particular route so that the traffic passes through all the nodes on the bisection line. This heuristic should be applicable to other wireless ad-hoc routing protocols utilizing route cache information and in which the route selection is based on the shortest hop count. However, future work should explore systematic ways of automatically generating the heuristic to identify the worst case scenario for any given routing protocol.

Our initial model ignored the energy consumed by the control overhead. As mentioned previously, this simplification is acceptable when there is little network dynamics (e.g., node mobility). Our simulation results confirmed that the simplified model can provide some useful insight. However, in future work, we would like to include node mobility and control overhead into our model to characterize the protocol performance under the worst case scenario more precisely.

In wireless networks, the interdependency between the higher layer protocol and lower layer properties complicates the protocol design and analysis. For example, the channel loss will trigger retransmission, and retransmission will consume more power. The reduced power level and and the traffic triggered by retransmission will again increase the channel's loss rate. We would like to extend our model to include this interdependency, and see if there is change in the worst case scenario and the amount of traffic required to bisect the network.

We also plan to apply STRESS techniques to other wireless network protocols (e.g., sensor net protocols).

5 Conclusion

Motivated by mission critical applications, we developed an approach to systematically test wireless ad-hoc routing protocol performance under worst case scenarios that target extreme operating conditions. Because of some important energy stringent applications, we targeted the network partition problem that results from some nodes being depleted of power rapidly. The corresponding worst case scenario is the one that causes network partition with the least host traffic. Using extended FSM to model the wireless routing protocol and heuristic search, we were able to identify the worst case scenario (i.e., the least host traffic was required to deplete the energy of the bisection line nodes, and in turn caused network partition). Both the STRESS worst case scenarios and some random scenarios were simulated. Significantly lower host traffic was required to bisect the network for the STRESS scenario than for random scenarios for the same network topology. Furthermore, the simulation result from the above STRESS scenario is similar to the worst case in the theoretical analysis.

Two additional case studies were presented to give a concrete example of how the analysis of the wireless ad-hoc routing protocols under the worst case scenarios provides useful insight into protocol design and deployment. First, based on the simulation result for the worst case scenarios, AODV has less or similar control packet overhead than DSR in most cases, contrary to the results from previous simulation studies on random scenarios. Second, in order to study the impact of the STRESS analysis on energy sensitive ad-hoc routing protocols, DSR was modified to EDSR (energy aware DSR), where routing decisions were based on energy information instead of hop count. Simulation studies indicated that EDSR moderately prolonged the network lifetime (deliver 15% more host packets before network partition) for the STRESS worst case scenario. However, it only slightly prolonged the network lifetime (on average, deliver 2% - 3% more host packets) for random scenarios. In this way, the STRESS analysis of the worst case scenario provided a different perspective on protocol performance evaluation.

6 Acknowledgements

The authors are grateful to Ahmed Helmy, Lewis Girod and all members at Computer Networks and Distributed Systems Lab at USC for many invaluable discussions, feedback and help.

References

[1] Ahmed Helmy; Deborah Estrin. Simulation-based 'stress' testing case study: A multicast routing protocol. *Sixth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, July 1998.

[2] Kevin Fall and Kannan Varadhan (Eds.). ns notes and documentation, 1999. pages http://www–mash.CS.Berkeley.EDU/ns/.

[3] F. Cao; J. Gilbert; and S-H. Teng. Partitioning meshes with lines and planes. page ftp://ftp.parc.xerox.com/pub/gilbert/csl9601.ps.Z.

[4] Ahmed Helmy; Deborah Estrin; Sandep Gupta. Fault-oriented test generation for multicast routing protocol. *Formal Description Techniques & Protocol Specification, Testing, and Verification (FORTE/PSTV) IFIP*, Nov 1998.

[5] Josh Broch; David A. Maltz; David B. Johnson; Yih-Chun Hu; Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. *the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking; ACM (MOBICOM'98) Dallas TX*, October 1998.

[6] Per Johansson, Tony Larsson, Nicklas Hedman, and Bartosz Mielczarek. Routing protocols for mobile adhoc networks a comparative performance analysis. *the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, ACM, (MOBICOM'99)*, August 1999.

[7] David B. Johnson and David A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks. In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5.* Kluwer Academic Publishers, 1996.

[8] Mike Woo; Suresh Singh; C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. *SIGCOMM'98.*

[9] Charles Perkins; Elizabeth Royer. Ad hoc ondemand distance vector routing. *2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb 1999.

[10] Samir Das; Charles Perkins; Elizabeth Royer. The dynamic source routing protocol for mobile ad hoc networks. http://www.ietf.org/internetdrafts/draftietfmanetdsr01.txt. *INFOCOM'2000*, March 2000.

[11] Lipton Tarjan. A separator theorem for planar graphs. *Partitioning Meshes with Lines and Planes*, pages 36:177–189, April 1979.

[12] Jae-Hwan Chang; Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. *INFOCOM'2000*, March 2000.

[13] Ahmed Helmy; Sandep Gupta; Deborah Estrin; Alberto Cerpa; Yan Yu. Performance evaluation of multipoint protocols using systematic scenario synthesis: A case study for timer-suppression mechanisms. *USC Tech Report 00-727*, 2000.

[14] Sandeep Bajaj; Lee Breslau; Deborah Estrin; Kevin Fall; Sally Floyd; Padma Haldar; Mark Handley; Ahmed Helmy; John Heidemann; Polly Huang; Satish Kumar; Steven McCanne; Reza Rejaie; Puneet Sharma; Kannan Varadhan; Ya Xu; Haobo Yu; Daniel Zappala. Improving simulation for network research. *Technical Report 99-702, University of Southern California*, March 1999.