

Energy Efficient Routing in Nomadic Networks

Mads Darø Kristensen
Department of Computer Science
University of Aarhus, Åbogade 34, DK 8200 Århus N, Denmark
Email: madsk@daimi.au.dk

Niels Olof Bouvin
Department of Computer Science
University of Aarhus, Åbogade 34, DK 8200 Århus N, Denmark
Email: n.o.bouvin@daimi.au.dk

Abstract

We present an evaluation of a novel energy-efficient routing protocol for mobile ad-hoc networks. We combine two techniques for optimizing energy levels with a well-known routing protocol. We examine the behavior of this combination in a nomadic network setting, where some nodes are stationary and have a steady power supply. Protocol optimizations for the nomadic scenario are presented and validated through simulations.

1 Introduction

Modern mobile devices are becoming increasingly smaller. When small battery powered devices with limited processing capabilities are participating in MANETs, some interesting new problems occur; problems that are not present when the network consists of more powerful devices. One such problem is that energy efficiency suddenly becomes a major concern when routing network traffic.

Much research has been conducted in recent years to develop energy efficient routing protocols for MANETs. This research has led to a number of proposals for energy efficient routing protocols such as the ones described by Xu *et al.* [8] and Chen *et al.* [2]. Common for these routing protocols are that they assume that all nodes in the network are battery powered. This is not the case in nomadic networks. Here only the mobile devices (nomads) are expected to be battery powered while the stationary devices (oases) may have a constant power supply. As such, we are in this network setting considering the opposite case of Seet *et al.* [7], as our supernodes are stationary rather than moving on buses.

In this paper, we

1. Investigate some existing energy efficient routing protocols and analyze how they can utilize the stronger nodes in the nomadic network setting.
2. Suggest modifications to these protocols so that the nomadic network setting may be taken into consideration.
3. Present simulation results showing the gains of this specialization.

The energy efficient routing protocols that have been proposed all have different ways of achieving energy efficiency but they do fall into a small number of classes. The major divider is between *power-save* techniques that try to utilize the sleep states of the wireless interface, and *power-control* techniques that try to control the transmission power of the wireless interface. There is also a division between techniques that require nodes to have some sort of positioning information through e.g., GPS, and techniques that have no such requirements.

We consider the power-save approach more interesting than the power-control one because, in most MANET scenarios, time spent in the idle state dominates, and a lot of energy can thus be saved by proper utilization of sleep states. This paper therefore focuses on power-save protocols. Furthermore, we only consider protocols that do not rely on positional data.

2 Energy Efficient Routing

Mobile ad-hoc networking (MANET) has fostered much research in the area of efficient routing protocols, be they

proactive or reactive. For a general description of many of these protocols, see [1]. While these protocols perform admirably under many different conditions, most of them focus on network performance, and do not take energy efficiency into consideration as noted by Yu *et al.* [9]. In the context of this paper, we define energy efficient protocols as protocols that maximize the lifetime of the entire network. In recent years, some approaches towards energy efficient MANET routing have been proposed, e.g., [3]. This paper is concerned with two such approaches towards energy efficiency and how they may be combined and modified to fit into the nomadic network setting.

Xu *et al.* [8] describe two power-save approaches; the basic BECA and an extended version called AFECA. These approaches entail dynamically switching the nodes between sleeping, listening, and active states. The nodes switch between these states with fixed intervals, and in order to ensure the successful forwarding of messages, the active nodes may have to retransmit messages a number of times before the receiving node is listening or active. Actively communicating nodes stay awake.

The time intervals used in BECA are the following:

T_l The time that a node spends listening for activity.

T_s The time that a node spends sleeping.

T_a The period of time that a node remains active when no messages are being processed, i.e., when the node is in fact not active anymore.

An inactive node listens for activity for T_l seconds, and if no activity is seen during that period of time it sleeps for T_s seconds before listening again. It is important to note here, that *activity* is not just receiving any message – a node only stays awake if it is needed for forwarding data or routing information. Retransmissions are used to ensure message arrival. T_l and T_s are chosen so that $T_s = k \times T_l$ where k is some small integer. Messages are then retransmitted $k + 1$ times, which guarantees that the next hop node receives at least one of the retransmissions. BECA's ability to preserve energy obviously depends on the ratio k between T_l and T_s ; a large ratio means a larger sleep interval but also that a larger number of retransmissions are needed. Extensive simulation studies done in [4] showed that using a large k yielded a very good energy efficiency; but this came at the expense of high delivery latencies and low delivery ratios. A reasonable choice was therefore found to be $k = 1$. For a more thorough examination of BECA see [4, Chapter 3] and for simulation results where the effects of varying T_l , T_s and k are studied, see [4, Chapter 6].

The difference between BECA and AFECA is that AFECA takes node density into consideration when determining the length of the interval in which a node may sleep. Both approaches are purely power saving algorithms and

not routing protocols. They therefore need to be combined with some existing MANET routing protocol. A reactive routing protocol is well suited for this purpose since the periodic control messages sent in a proactive protocol would keep the nodes awake even in low traffic scenarios. For the simulations described herein, BECA and AFECA are built on top of AODV [6], described below.

Span, as it is described by Chen *et al.* [2], is a power-save approach based on the notion of connected dominating sets (CDS). A CDS is a connected sub-graph S of a graph G such that every vertex u in G is either in S or adjacent to some vertex v in S . In layman terms the CDS is a set of nodes from which all other nodes in the network can be reached. Thus, the nodes (or *coordinators* as they are called in Span) in the CDS are ideally placed to act as routers for the entire network. By load-balancing the coordinator role fairly among the participating nodes, through a coordinator selection process, Span can ensure that the coordinator group will be able to perform as long as possible. Span in itself does in fact not save power; it merely provides an intelligent way of selecting a CDS of coordinator nodes. Once the CDS has been found, a power-save approach must be utilized to do the actual saving of power. The power-save approach used must be tailored to work together with Span so that only non-coordinator nodes are participating in the power-saving scheme, i.e., coordinator nodes must not be put to sleep.

Span selects its coordinators by running a distributed coordinator selection/withdrawal algorithm. Regular nodes check periodically, whether they should become coordinators, and coordinator nodes check to see if they should withdraw from the coordinator role. The coordinator selection scheme takes such factors as the remaining battery capacity of the node and the *utility* of the node into consideration. The utility of a node is a measurement of how much more connected the network would be if that node was chosen as a coordinator, and it is measured in how many more pairs of neighbour nodes that would be connected if the node was chosen as a coordinator. The remaining battery capacity is used to make sure that stronger nodes, i.e., nodes with larger amounts of energy, are selected as coordinators, and the utility is used to minimize the number of coordinator nodes. For a more thorough description of Span see [4, Chapter 3].

In this paper, we present a comparison of the combination of Span with AFECA running on top of AODV compared with the same combination modified for nomadic networks.

AODV [6] is a reactive MANET routing protocol and as such a node only updates its routing table when data is in fact flowing through the network. Routes in AODV are built on demand by broadcasting route request (RREQ) messages and then waiting for the target node, or an intermediate node, to return a route reply (RREP) message.

There is one thing that is maintained pro-actively in AODV and that is the list of neighbors. Nodes emit periodically a HELLO message to show their neighbors that they still exist. These HELLO messages are used for a number of things, e.g., to check whether links are bi-directional and to invalidate active routes.

Placing Span on top of AODV is quite trivial. Span uses some periodic exchange of neighborhood information between neighboring nodes to build the CDS of coordinator nodes. This neighborhood information can easily be piggy-backed on the HELLO messages that are already sent in the AODV protocol. To enforce that only coordinators should be used as routers, we have modified the AODV protocol, so that only coordinators can forward route request (RREQ) messages. Seeing as routes in AODV are built by following the reverse path of RREQ messages, only coordinators will be used for routing.

When BECA (or AFECA) is combined with Span, some optimizations become possible. For one, coordinators are by definition always available, which means that there is no need for the BECA imposed retransmissions between these nodes. This saves a *lot* of the retransmissions normally done in BECA because coordinator nodes are the only routers of data. Retransmissions are therefore only needed at the last hop of a route; when the message is delivered to a non-coordinator node. Another important optimization is that k , the ratio between T_l and T_s , can be larger when Span is used. This is related to the low usage of retransmissions that were just mentioned; if the only places where retransmissions are done is at the last hop of a route, more retransmissions are not significant. A larger ratio would usually mean larger packet delivery latencies, but this is not the case when Span is used since the coordinator backbone is always on. A larger ratio between T_l and T_s will in the end mean lower energy consumption. The effects of these optimizations have been thoroughly simulated and parameter tuned in [4, Chapter 6].

These modifications to the above described protocols yield a system, which gracefully combines the two power-saving schemes with a well-established MANET routing protocol. We are, however, also interested in a nomadic network scenario, wherein some peers are not battery constrained. Such peers would not need to sleep and would therefore be excellent candidates for permanent membership of the coordinator set. Ideally, such “oases” or super-nodes would be within reach of each other, forming a permanent routing backbone—this is however an unrealistic requirement, and our design does not make such assumptions.

The only part of the Span/BECA/AODV chain that needs to be modified to take supernodes into consideration is Span’s coordinator selection mechanism. By forcing supernodes to become coordinators, regardless of how well they fare in the coordinator selection algorithms, these

nodes will do as much of the routing as possible. Coordinators are in Span chosen based on a number of criteria, among which remaining energy is one. It could therefore be expected that Span automatically would favor supernodes as coordinators, but as the simulations show (see Section 4), this predisposition is not enough on its own.

3 Simulation

In this paper we will examine two sets of protocol combinations, the original and the modified Span/AFECA/AODV under nomadic conditions. As a baseline, we have provided results with an unmodified and unaided AODV implementation. Our tests have taken place in our own simulator, purpose-built to accurately model and monitor energy consumption. The simulator has been designed with due consideration of the issues raised by Kurkowski [5]. The simulator is open source and freely available¹.

Routing can be measured using many parameters. One obvious measure given the context of energy efficiency is energy usage. But, seeing as being energy efficient often entails a trade-off between data delivery and energy usage, we have chosen to also include a performance measurement using the product of the success rate of data delivery and the average remaining energy, both measured in percent.

We are using the following conditions in our simulations

Simulation area 1000 m × 1000 m.

Transmission range 250 m (yielding a maximum number of hops of 12).

Node density Low, which is a total of 33 nodes yielding an average node density of ~5, and high, which is a total of 65 nodes yielding a node density of ~10.

Movement Nodes move by a random way-point model at speeds 0–5 m/s with a randomly chosen pause time of 0–60 seconds.

Traffic At any point in time approximately 10% of the nodes are actively communicating. Such communicating nodes send 16 packets of 1024 bytes per second (16 KiB/s). New sender/recipient pairs are selected every 60 seconds.

Energy All mobile nodes start out with the same amount of energy (1000 Joule).

Simulation runs All simulations are run for 600 seconds and the reported results are averages of 25 runs. The standard deviation of these runs were very small and are thus not reported here.

¹<http://www.daimi.au.dk/~madsk/>

Protocol parameters The AFECA and Span combination has been extensively parameter tuned in order to optimize performance as summarized in Table 1 (see [4, Chapter 6] for further discussion).

Listen time (T_l)	Sleep time (T_s)	Active time (T_a)
1 sec	5 sec	30 sec

Table 1. AFECA parameters

4 Results

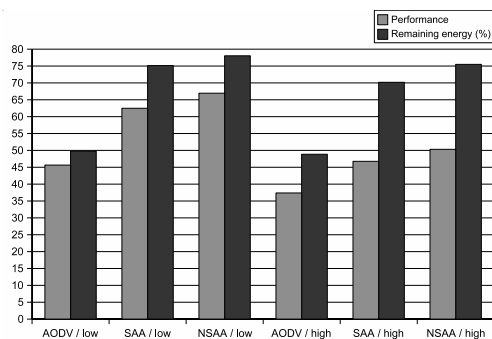


Figure 1. Simulation results for low and high density tests. For each simulation the remaining energy and the performance measure (remaining energy \times delivery ratio) is plotted.

The simulation results can be seen in Figure 1, where SAA designates the Span/AFECA/AODV combination and NSAA the nomadic version. An examination of the plot in Figure 1 yields a number of conclusions. First off, it is evident that a lot of energy is saved by using an energy efficient routing protocol; which also shows that the Span/AFECA combination does a good job of preserving energy. The energy efficient variants of AODV use about half of the energy that pure AODV does. Performance-wise, the difference between the protocols is less pronounced. The energy efficient protocols have the best performance score, but the higher delivery ratio of AODV makes it look a little better in this measure.

Another thing that becomes apparent when looking at the plot is that the nomadic version of Span/AFECA performs better than the non-nomadic one. This tells us two things: one, that it is worthwhile to force supernodes into the coordinator role, and two, that Span's own coordinator selection mechanism is not sufficiently geared towards utilizing the stronger nodes in a nomadic network.

A more thorough examination of Span's coordinator selection algorithm is needed to establish why Span by itself is not sufficiently proficient at choosing the right coordinator nodes. Span's coordinator selection algorithm, which is a calculation of a back-off delay, is shown in Equation (1).

$$delay = \left(\left(1 - \frac{E_r}{E_m} \right) + \left(1 - \frac{C_i}{\binom{N_i}{2}} \right) + R \right) \times N_i \times T \quad (1)$$

The back-off delay calculation in Equation (1) randomly chooses a delay over an interval proportional to $N_i \times T$, where N_i is the number of neighbors for node i and T is the round trip delay for a small packet. The random part of the expression is built of three important pieces. The energy aspect is reflected in the sub-expression $(1 - E_r/E_m)$, where E_r is the remaining capacity and E_m is the maximum amount of energy that the node can have. The utility of the node, that is the current value of having the node as a coordinator, is reflected in $(1 - C_i/\binom{N_i}{2})$, where C_i is the number of additional pairs of nodes that will be connected if i is selected as a coordinator. Finally, the random part R is used to prevent nodes from announcing their willingness to become a coordinator at the same time².

What is most interesting from our point of view is the energy part of the delay calculation. A supernode is always at 100% capacity, and the energy part is thus zero for these nodes. Unfortunately, this is initially also the case for the mobile nodes when they are at 100% capacity, and the linear decline represented by $(1 - E_r/E_m)$ is not enough to ensure that supernodes are favored enough; the mobile nodes decrease too slowly in this measure. Running the simulations for a longer period of time should even this out a bit, because the mobile nodes use more energy. To test this, we have run some simulations for 1200 seconds instead of 600 seconds, and the results for pure Span/AFECA were only slightly better in this case.

5 Conclusion

We have in this paper investigated how methods for energy-efficient routing may be combined, both in an conventional MANET setting and in a nomadic setting.

Our results show that the novel combination of Span and AFECA performs well in an energy constrained setting. We have shown that simple optimizations for the nomadic setting yield superior performance. Furthermore, it is shown that Span alone is unable to adequately utilize the available supernodes in a nomadic network. We believe that there is

²It is debatable how relevant this is in the asynchronous environment that is provided by Span/AFECA. For more information about this see [4, Chapter 6]

still room for improvement in the tuning of the Span coordinator selection algorithm, especially in cases of heterogeneous nodes.

References

- [1] E. M. Belding-Royer. Routing approaches in mobile ad hoc networks. In S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, editors, *Mobile Ad Hoc Networking*, chapter 10, pages 275–300. Wiley-Interscience, 2004.
- [2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal*, 8(5):481–494, 2002.
- [3] L. M. Feeney. Energy-efficient communication in ad hoc wireless networks. In S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, editors, *Mobile Ad Hoc Networking*, chapter 11, pages 301–327. Wiley-Interscience, 2004.
- [4] M. D. Kristensen. Evaluating energy efficient routing protocols for nomadic networks. Master’s thesis, Department of Computer Science, University of Aarhus, Aug. 2006. <http://www.daimi.au.dk/~mads/>.
- [5] S. Kurkowski, T. Camp, and M. Colagrosso. MANET simulation studies: The incredibles. *ACM’s Mobile Computing and Communications Review*, 9(4):50–61, 2005.
- [6] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99.*, pages 90–100, Feb. 1999.
- [7] B.-C. Seet, C.-T. Lau, W.-J. Hsu, and B.-S. Lee. A mobile system of super-peers using city buses. In *Proceedings of the 2nd International Workshop on Mobile Peer-to-Peer Computing*, pages 80–85, 2005.
- [8] Y. Xu, J. Heidemann, and D. Estrin. Adaptive energy-conserving routing for multihop ad hoc networks. Research report 527, USC/ISI, Oct. 2000.
- [9] C. Yu, S.-M. Park, and H.-N. Kim. Every node is born equal: attacking preferential attachment in peer-to-peer mobile multihop networks. In *Proceedings of the 2nd International Workshop on Mobile Peer-to-Peer Computing*, pages 70–74, 2005.