

Simulation and Evaluation of Unsynchronized Power Saving Mechanisms in Wireless Ad hoc Networks

Philipp Hurni¹, Torsten Braun¹, and Laura Marie Feeney²

¹ Institute of Computer Science and Applied Mathematics, University of Bern
hurni, braun@iam.unibe.ch

² Swedish Institute of Computer Science lmf@ics.se

Abstract. Power saving mechanisms in wireless ad hoc network nodes mainly switch off the transmission and reception hardware for a maximal amount of time and turn it on again within a given interval. Many approaches aim to synchronize the state changes of the nodes in the network through distributed beacon generation and introduce mechanisms where nodes synchronously wake up at designated points of time to exchange announcements about pending traffic. Synchronization however is difficult to achieve, in particular in ad hoc networks.

This paper describes the simulation, evaluation and refinement of a recently proposed power saving approach based on asynchronous wake-up patterns and wake-up announcements integrated with AODV. We show that significant improvements of the connectivity under low wake ratios can be achieved by carefully designed forwarding strategies of AODV route request messages.

1 Introduction

In wireless mobile ad-hoc and sensor networks, efficient power saving mechanisms can drastically increase network lifetime. However, reasonable connectivity properties are nevertheless necessary for wireless networks to operate properly. The major part of many power saving mechanisms consists in introducing central or distributed synchronization and periodic switching between a sleep state and a wake state. Such synchronization measures however always cause new overhead. Recent publications therefore proposed variants of unsynchronized power saving mechanisms.

One of the most important related work is the IEEE 802.11 [2] power saving mode, where all nodes ideally wake up at the same time, at the beginning of a beacon interval, and remain awake during the ATIM window (Asynchronous Traffic Indication Map) to exchange traffic announcements in case of pending traffic, and fall asleep again if there is none. Quorum based systems [3] divide a single hop network into intersecting groups of nodes which wake up in different wake intervals, thereby permitting to reach any node by relaying traffic announcements on intermediate nodes.

The power saving mechanism specified in [1] avoids the overhead of time synchronization. It proposes to take advantage of intersections in asynchronous wake patterns by relaying messages on intermediate nodes.

Another approach aiming to bypass the overhead of permanent synchronization proposes an asynchronous sleep-wake cycle with two ATIM-windows to disseminate phase announcements [4]. A transfer window is in between the two ATIM windows during which the node has to stay awake for at least 50% of the total wake cycle. This ensures that neighboring nodes are always able to forward pending traffic and phase announcements within one cycle.

This paper describes the integration of the basic concept of the power saving mechanism proposed in [1] into a multi-hop wireless ad hoc environment using AODV and the IEEE 802.11 medium access procedure. Section 2 describes the design of the routing and medium access layer and proposes modifications to these. In section 3, we outline and examine the results of different simulation scenarios and propose refinements of the mechanism [1]. The key results concern the optimization of the connectivity properties when applying carefully designed forwarding strategies to rebroadcast AODV route requests. For low wake ratios, we can significantly improve the connectivity by increasing the number of message rebroadcasts. Section 4 concludes the paper.

2 Power Saving in Multi-Hop Wireless Ad hoc Networks

2.1 Basic Power Saving Mechanism

The mechanism proposed in [1] defines two wake and two sleep periods during one basic cycle duration T , as depicted in Figure 1. Each of the wake periods shall have the same duration t . The nodes strictly alternate between a fixed wake period (A) and a random wake period (B). The fixed wake period (A) always starts at the same time, exactly at the beginning of the basic cycle. The start of the random wake period (B) is uniformly distributed between the end of the fixed wake period (A) and the start of the next one. All nodes are assumed to operate with the same basic cycle duration T , although remaining unsynchronized. All nodes shall switch between the wake and sleep states in their individual wake-up pattern. In addition, we assume all nodes to operate with the same wake ratio $W = 2t/T$. Due to low drifts in clocks, this should be achievable in practice. Small differences as they might occur due to clock drifts do not matter. The fixed wake period (A) enables a node aiming to contact any neighboring node, if its periodically occurring fixed wake period pattern is known. If however there is no intersection between the fixed wake periods of the sender and the neighbor, it may never learn about its presence. This motivates the choice for the random secondary wake period (B). It ensures that two nodes with disjoint wake-up pattern will sooner or later be awake at the same time and therefore be able to exchange announcements about their own wake period. By receiving these,

the nodes will learn about the wake-up patterns of their neighbors, and thus be capable to reach any neighboring node during their fixed wake period (A).

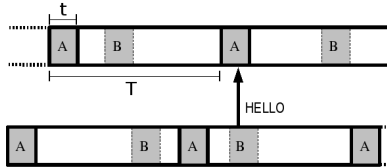


Fig. 1. Neighboring nodes with disjoint wake-up pattern announcing their wake state

2.2 Integration of the Ad Hoc On Demand Distance Vector Protocol (AODV)

AODV [7] is the most appropriate routing protocol considering the use with the power saving mechanism. The mechanism proposed in [1] is well integrable into on-demand routing. We modified the HELLO messages specified in [7] to contain a time entry indicating the time left until the beginning of the next fixed wake period (A). HELLO messages are broadcast within a HELLO interval during the next fixed or random wake period. If any neighboring node is awake, it will receive the wake-up announcement. With the time information contained in the HELLO message, it is able to calculate the relative time difference to its own fixed wake period (A). The node shall therefore update its routing table entry for the corresponding node with this time difference anytime it receives a HELLO message. Small clock drifts may only have little influence, because with every wake-up announcement received in a HELLO message, the table entries of the nodes can be updated and possible drifts can be corrected. Choosing a random interval duration for the HELLO interval guarantees that within a certain amount of time, any neighboring nodes will be detected. Even if their fixed wake periods (A) are disjoint, the mechanism specified in section 2.1 will ensure intersections with the random wake period (B), through which the nodes can learn about each other's presence by exchange of HELLO messages. Figure 1 shows such two neighboring nodes with non intersecting fixed wake patterns.

If a HELLO timer runs out during the sleep state of a node, the HELLO message is scheduled to be sent during the next wake period. The HELLO timer is furthermore restarted immediately. This ensures that the expected value of the count of HELLO packets sent remains independent of the wake ratio and is controllable solely through the parameter HELLOINTERVAL, which specifies the interval to choose a random time delay until scheduling the next HELLO message.

2.3 Medium Access Control (MAC) Layer Modifications

IEEE 802.11 [2] was chosen as the reference case for the integration of the power saving mechanism, as it is the most widely used standard for wireless communication. IEEE 802.11 proposes to access the medium immediately after the medium has been sensed idle for at least one DIFS, and to apply a random backoff procedure only if the medium had been sensed busy before. This immediate medium access guarantees minimal delays and better throughput under low medium utilization and low node densities. We adopted this MAC procedure to apply a random backoff in any case of medium access, as it led to fewer collisions and therefore proved to be more effective for the unacknowledged broadcast of route requests that is elementary for the success of the route establishment of AODV. The integration of the power saving mechanism proposed in [1] into a 802.11 environment leaves some design questions open, in particular concerning the behaviour of nodes in different states of the MAC layer protocol and the state changes of the power saving mechanism. The following assumptions were chosen:

- As unicast traffic is acknowledged, we decided to use DFWMAC (Distributed Foundation Wireless MAC) without RTS/CTS.
- If the wake state timer runs out while a node is receiving a frame, the node does not turn off immediately, but remains awake until the end of the transmission. In case of unicast traffic, the receiver stays awake until having sent the following ACK.
- If the wake state timer runs out while a node is waiting during the DIFS, the contention is aborted and the node turns to the sleep state.
- If the wake state timer runs out while a node is contending for medium access and is already counting down a random backoff interval, but has not yet accessed the medium, the node instantly switches to the sleep state. The node will restart contention by choosing a new random backoff value during the next wake state. The contention window will remain unchanged.
- If the wake state timer runs out while a node is transmitting, the node switches to the sleep state after having completed the transmission.

We decided to give unicast traffic a higher priority than broadcast traffic. If a node wakes up, and there are different types of messages in the queue to be sent, it sends the unicast (RREP or DATA) messages first. This should ensure predictable behaviour of unicast data traffic delays. Moreover, it is not necessary to handle HELLO broadcasts very quickly, and RREQ broadcasts yet have the advantage of redundancy through neighboring nodes.

3 Evaluation of Power Saving Mechanisms

3.1 Simulation Environment and Parameters

In the following sections, we perform several experiments to measure the ad hoc networks' properties. First, we examine the neighbourhood discovery process when no traffic is generated rather than the HELLO message transfers of

the mechanism specified in section 2.2. We determine how long it takes for each node to learn about the presence and the wake-up patterns of their respective neighbors. Secondly, we study the performance of four algorithms concerning the forwarding of RREQ messages according to AODV route discovery. We discuss advantages and drawbacks of each approach in respect to connectivity optimization and power consumption.

For performance evaluation, we used the OMNeT++ Network Simulator [8, 9]. We made use of the Mobility Framework from TU Berlin [10], a framework to support simulations of wireless and mobile networks within OMNeT++. This framework incorporates a sophisticated transmission model which is based on calculation of SNR (Signal-to-Noise Ratio) and SNIR (Signal-to-Noise-and-Interference Ratio) values according to a restricted free space propagation model. This model takes transmitter power, distance, wavelength and path loss coefficient of signal dispersion into account. The following simulation parameters were kept fix throughout all simulation scenarios:

Nodes	200 (uniform distribution)
Area	1000m × 1000m
Bitrate	2 Mbps
Carrier Frequency	2 GHz
Transmitter Power	2 mW
SNR Threshold	3 dB
Path Loss Coefficient	2
Thermal Noise	-110 dBm
MAC Header	32 Byte
AODV Header	34 Byte
RREQ_RETRY_TIMER	800ms
HELLOINTERVAL	2s

Nodes are assumed to be stationary throughout all cases of the following sections. We adjusted the parameters such that the transmission range in case of one transmitter and no interfering other stations reaches 100m. In this case, the same station was considered not to interfere with a receiver farther than 200m (interference range). Furthermore, any station within the range of 200m (carrier sensing range) also considered the channel to be busy. In the following experiments, we basically varied the basic cycle duration T and the wake ratio W of the proposed power saving mechanism.

3.2 Neighborhood Detection

Basic Cycle Duration T	200ms
--------------------------	-------

At the start of the simulation, we assume all nodes to be in one of the respective states (wake states A, B; sleep state). Furthermore, every node starts its HELLO timer in respect to the given interval length HELLOINTERVAL. The first HELLO message is sent after an offset t_H , which is a uniformly distributed

random value in the interval $[0, \text{HELLOINTERVAL}]$. The random wake period (B) and the mechanism of choosing random time intervals between two HELLO messages ensures that every node should detect its neighbors within a certain amount of time. The following experiment deals with measuring how long it takes for the nodes to discover each other under the given parameters.

To make sure that neighborhood detection succeeds quickly, we optimized the neighborhood detection process: if a node receives a HELLO message from a neighbor, for which it had no entry in its table before, it schedules a HELLO message at the start of the next fixed wake period of the sender node, waking up regardless of its actual state. In this case, there is a high probability that the receiving node is yet unknown to the sending node. With the wake-up time information in the HELLO message, the receiving node can announce its own wake-up pattern to the sending node and significantly speed up the neighborhood detection process.

With the given basic cycle duration T and $E(t_H) = 1s$, every node would broadcast a HELLO message every 5 cycles in average. Figure 2 depicts the results of the simulation when traffic only consists in periodic HELLO message transfer. The simulation results yield that approximately 80% of the neighbors are already detected within 2s for a wake ratio of 20%. For the very low wake ratio of 1%, 95% of the neighbors are discovered within 2 minutes. The evaluation of the neighborhood detection process yields, that nodes can discover most of their immediate neighbors within a few minutes.

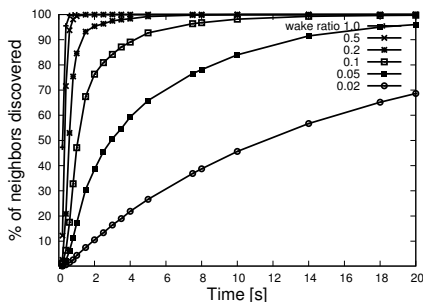


Fig. 2. Neighborhood detection

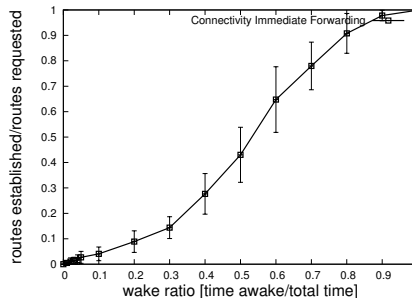


Fig. 3. Connectivity when forwarding immediately

3.3 AODV Route Establishment

Route Requests (RREQ)	60
Neighborhood Discovery Interval	2min
Basic Cycle Duration T	200ms

To evaluate the route establishment performance of the power saving mecha-

nism proposed in [1], we triggered 60 AODV Route Requests (RREQ) after a neighborhood discovery interval of 2 minutes. Every second during one minute, a pair of nodes, sender and destination, was chosen randomly among the distributed nodes to trigger an AODV RREQ. This simulation scenario remained unchanged throughout this section. The reason not to choose a higher number of route requests lies in the route establishment mechanism of AODV, which allows to shorten route requests by letting intermediate nodes respond instead of destination nodes. The connectivity properties will therefore become slightly better with every route request triggered. We wanted this influence to remain small to measure the performance of the power saving mechanism, rather than pure AODV properties.

If after a given time interval there is no response to an outgoing RREQ, [7] proposes to retry the request again with a higher sequence number. The retry limit was set to 2 in all cases of this scenario, permitting 3 RREQ broadcasts in total for each route discovery process.

The selected node density is not necessarily sufficient to ensure that every random pair of nodes can reach each other. It is most likely that there are scattered clusters of nodes without any path to each other. Therefore, the route requests where sender nodes tried to reach unreachable destination nodes were not triggered and accounted in the evaluation. The further sections therefore take the number of routes that are actually possible under the given circumstances as a basis for the calculation of the success ratios.

Immediate Forwarding of Route Requests.

If a sender node wants to establish a route to a destination, it sets up a RREQ packet, contends for medium access and broadcasts the packet. Every RREQ received is forwarded only once. Neither the sender node, nor any of the intermediate nodes make use of the time information contained in the RREQ message when rebroadcasting it. In contrast, the RREP packets, which are sent unicast, always make use of the announcement about the next fixed wake period. Their transmission is therefore scheduled for the beginning of the fixed wake period (A) of the receiving node.

We discovered that many RREQ messages broadcasted across the whole network were lost due to collisions with their copies. Since broadcasts are unacknowledged, every packet loss remains undetected for the originating node, and this proved to have a strong impact on the success of the RREQ-RREP query cycle. The problem of coherent RREQ discovery waves is studied in [11]. Imposing a jittering scheme on the retransmission of RREQ broadcasts significantly reduces the number of collisions. We chose every node to defer for a small random period between 0 and $3 \times (\text{RREQ_length}/\text{bandwidth})$, which proved to deliver substantially lower collision rates.

Figure 3 depicts that this strategy works well with high wake ratios, but remains ineffective for low wake ratios. In theory, the probability for a RREQ to reach its destination is expected to decrease with increasing distance. The farther the

destination is away, the more collisions may occur.

As we are dealing with a network of random topology, the length of the optimal route between two nodes was taken as a metric to measure the distance between sender and destination. Figure 4 displays the rates of successful RREQs. For each distance x , the value of y depicts the ratio between the number of successful RREQs to destinations with this distance and the totally sent RREQs. The ratio between the routes requested and the routes established decreases with increasing distance and decreasing wake ratio. It seems apparent, that the influence of even small reductions of the wake ratio is vast using the immediate forwarding strategy.

Figure 5 illustrates the ratio between the length of successfully established routes and the actual distance measured as optimal path length from sender to destination node. The lower the wake ratio, the longer are the paths that are actually found, relative to the optimal distance. When operating with lower wake ratios, the paths to the destinations may still be found, but with significantly more hops than in the optimal case. The average path length for a wake ratio of 100% averages out to 9.75 hops, whereas the average optimal path length averages out to 9.05 hops. It is obvious that even with wake ratios of 100% there is still a difference between the optimal path length and the path length that was actually found. Because we are dealing with nonreliable broadcasts, the routing protocol can not guarantee optimality of the routes.

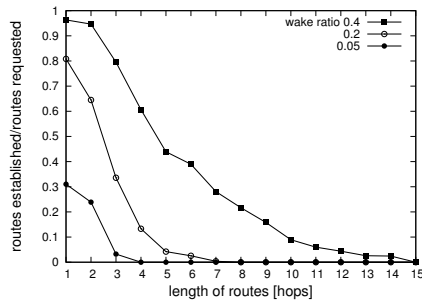


Fig. 4. Success ratio as function of the distance

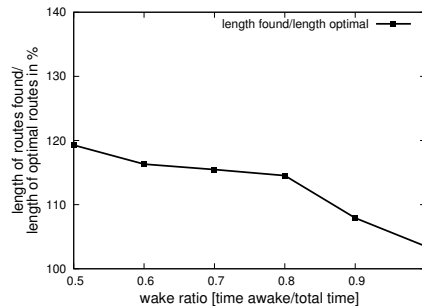


Fig. 5. Path optimality as function of the wake ratio

Forwarding at the Instant of Best Intersection.

With the wake-up announcements contained in the HELLO messages, the nodes know when to reach their immediate neighbors. Given a fixed wake ratio W and a fixed basic cycle duration T , a node intending to broadcast a message can figure out the best instant to forward the message using previously received wake-up announcements. The best instant shall be found as follows:

Let A be the current node, S the source node, D the destination node, N_1, N_2, \dots, N_i the neighbors of A .

- i) if A is S or an intermediate node and $D \notin \bigcup N_i \setminus N_j$, then the best instant is any instant when the largest subset of $\bigcup N_i \setminus N_j$ is awake, where N_j is the node from which A received the broadcast. In case $A = S$, N_j equals \emptyset .
- ii) if A is S or any intermediate node and $D \in \bigcup N_i \setminus N_j$ then the best instant shall be any moment when D is awake.

Figure 7 depicts the concept to search the best instant to forward a RREQ. If there are more than one possible instants, one intersecting moment shall be chosen at random. This shall prevent each retry attempt to fail because of identical choices of forwarding instants. In Figure 7, the node selects any instant in-between Δx , provided that neither N_1 nor N_2 nor N_6 happens to be the sender of the incoming RREQ. In such a case, the intersection between N_3 and N_4 would also be appropriate.

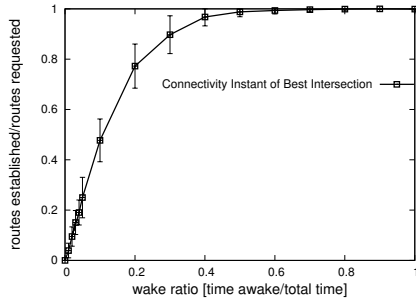


Fig. 6. Connectivity forwarding in the Instant of Best Intersection

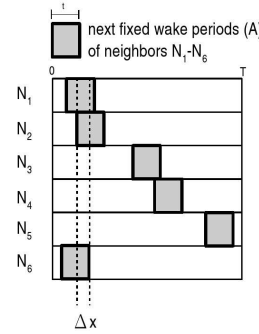


Fig. 7. Announced fixed wake state periods in an intersection table

The node aiming to transmit shall schedule the RREQ for the designated moment, regardless of its own wake pattern. If the node is in a sleep state, it shall wake up at the designated instant, transmit the message, and then continue with the former wake pattern. The additional wake time is added up and accounted for within the evaluation.

Figure 6 depicts that the connectivity is vastly improved when choosing the moment to forward according to the algorithm specified above. Even with a wake ratio of 20%, approximately 75% of the routes could still be established.

Further investigations and evaluations of the message flow led to the conclusion that it was not the wake ratio, which caused some routes not to be established, but the topology and the design of the algorithm used to determine the best moment to forward a broadcast. Especially if there were bottlenecks in the network topology, certain indispensable nodes were never selected to receive a RREQ by

its neighbors, which caused the mechanism to fail in every retry attempt. Figure 8 confirms that the route establishment success ratio decreases with increasing distance. The improvement indeed lies in the utilization of the wake-up information announced in the previous messages.

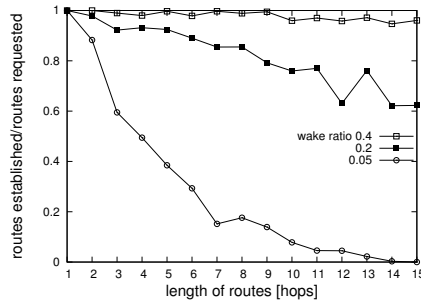


Fig. 8. Success ratio as function of the distance

Forwarding with Enhanced Broadcast.

The insights gained during the evaluation of the approaches above implied the following mechanism to be tested out: If a node aiming to forward a route request would be permitted to forward the RREQ more than once, it could nearly reach full connectivity even for low wake ratios. The following algorithm was used to find the appropriate instants to forward route requests:

Let A be the current node aiming to forward a broadcast packet, S be the source node, D be the destination node, N_1, N_2, \dots, N_i are the neighbors of A , and N_j the node from which A received the packet. Let R be the set of unreached neighbors and T the set of best instants. To figure out the minimal set of instants for A to rebroadcast the packet, proceed as follows:

$$R = \bigcup N_i \setminus N_j$$

WHILE ($R \neq \emptyset$) *do*

$$\quad \text{find the best instant } t_k \text{ to reach the largest subset } M \subseteq R$$

$$\quad T = T \cup \{t_k\}$$

$$\quad R = R \setminus M$$

od

T then contains the minimal set of instants t_k for broadcasts to be sent to reach all neighbors. Yet it is still possible that some neighbors remain unnotified. Collisions can still occur and hinder RREQ broadcasts to be spread across the network. Furthermore, waking up at the minimal set of instants does not guarantee the intended broadcast to be successful, because the medium access is delayed due to the contention back-off mechanism of the MAC layer. If the

node waking up has to wait for other stations to finish their transmissions until accessing the medium, the targeted nodes can already have turned to the sleep state again.

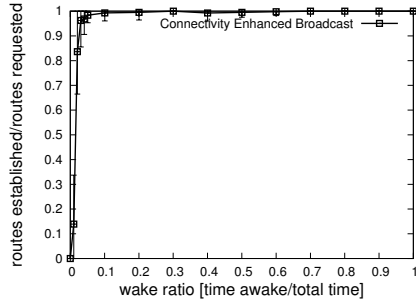


Fig. 9. Connectivity with Enhanced Broadcast

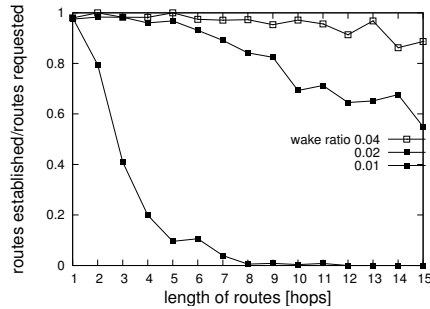


Fig. 10. Success ratios with Enhanced Broadcast

Figure 9 illustrates the effectiveness of this approach. The results however lead to the conclusion, that the more a route request is forwarded, the higher is the probability of reaching the destination. This approach leads to more message exchanges than the prior approaches. Nodes intending to send route requests must wake up several times and abandon their usual wake-up pattern. The additional wake-ups and message transmissions cause higher power consumption and wake time. The lower the wake ratio, the more additional wake-ups became necessary. We measured the additional overhead caused by the extra wake-ups. As the simulation scenario setup first lets the nodes discover their neighbors for 120 seconds and then only proceeds with a very low network load, the total of the additional overhead caused by the extra wake-ups did not carry much weight. We measured that a wake ratio fixed to 4% by setting the duration of the wake periods A and B turned out to be a net wake ratio of 4.087%. However, we discovered that the additional overhead clearly became bigger with decreasing wake ratio. Probably, with higher network load, the additional wake time would increase as well.

One may object that transmission mode is the most expensive operation mode for wireless devices, and that the additional transmissions are costly. This concern is legitimate. Recent investigations [6, 5] on the of power consumption of wireless network devices showed that the power consumption of the transmit mode is indeed highest (182% of idle mode consumption) and that receiving is also expensive (121% of idle mode consumption). However, the sleep mode inherently has the lowest power consumption (6% of idle mode consumption). By maximizing the sleep time, the overall benefit of the approach studied in this section should still be positive.

Figure 10 displays the success ratios of the forwarding with *Enhanced Broadcast* approach in dependence of the distance, analogous to Figure 4 and 8. It shall

be noticed that the connectivity properties reached with a wake ratio of only 10% are comparable to those of a wake ratio with 40% when using the prior forwarding approach. It shall be noticed, that the curve with the best connectivity function in Figure 10 was reached with a wake ratio of 10%, whereas in the approach using one single broadcast in the instant of best intersection a wake ratio of 40% were necessary to reach comparable connectivity. The fact that there is only one percent difference in wake time between the worst curve (1%) and the second worst curve (2%) of Figure 11 is astonishing. Furthermore, the network reaches a far better connectivity with 2% rather than 1%, whereas it does not perform that much better when doubling the wake time from 2% to 4%. It seems that the wake ratio can be decreased without major impact down to a certain limit, but further reduction leads to a sudden breakdown of the connectivity properties. This effect is commonly known as percolation and occurs in different aspects of ad hoc networks with random topology [14]. The critical values for the wake ratio using the simulation setup as described in 3.1 and the forwarding strategy of the *Enhanced Broadcast* are between 1% and 2%.

The comparison of the three approaches studied in this section suggest that the connectivity can be improved significantly by additional wake-ups and multiple rebroadcasts.

The approach of the *Enhanced Broadcast* has the fundamental disadvantage that it is not suitable for higher node densities. The more neighbors a node has, the more additional wake-ups will be scheduled and the more traffic will be generated, even if there is already a sufficient number of nodes awake to reach any node in the network. This may lead to much higher wake time and useless network load. In our simulation of the *Enhanced Broadcast*, the average count of RREQ forwarding operations increased from 2.24 for a wake ratio of 50% to 3.32 for 20% and to 4.38 for 5%. For even lower wake ratios, it approximated to 6 forwarding operations, which is near the average count of neighbors.

Limited Enhanced Broadcast.

It may not be suitable for high node densities that a node has to rebroadcast an incoming RREQ for almost every neighbor, but still it may be better to forward it more than once. We therefore define an upper limit for the number of forwards and thus limit the transmission overhead. Using the same scenario as before, we set the upper limit to 2 rebroadcasts. The results of the connectivity analysis are depicted in Figure 11. They provide evidence that a significant improvement of the approach using the *Instant of Best Intersection* is possible with a slight increase of the number of rebroadcasts.

3.4 Evaluation of Short Wake Periods

Basic Cycle Duration T	20ms
--------------------------	------

The results of the scenarios above may be interesting from the theoretical perspective, but in practice, an average delay of 100ms per hop is not suitable for

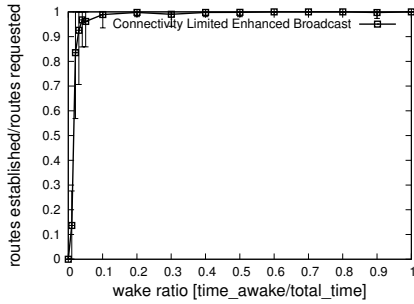


Fig. 11. Connectivity with Limited Enhanced Broadcast

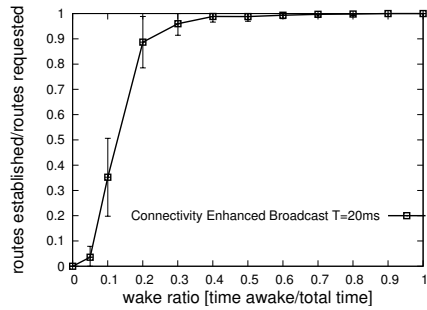


Fig. 12. Connectivity with Enhanced Broadcast and $T=20\text{ms}$

many kinds of application services. Real time services (i.e. Voice-Over-IP) often require maximum end-to-end delays of 200ms. This motivated to analyze the *Enhanced Broadcast* approach of section 3.3 with a much shorter basic cycle duration of $T=20\text{ms}$. One may object that this is too short for today's wireless nodes, as there is a delay for the transition between the wake and sleep state, in particular the wake-up time of the radio hardware. In fact, today's newest ultra-low power wireless sensors such as developed in the WiseNET project [15] have a wake-up delay of only $800\mu\text{s}$. Using 2 wake periods and eventually some additional wake-ups in case of RREQ traffic should therefore still be possible in a total cycle duration of 20ms.

The ad-hoc mode of IEEE 802.11, which is synchronized using beacons from designated stations and ATIM (Ad Hoc Traffic Indication Map) messages to announce traffic, typically uses a beacon interval length of 100ms. The lower this interval duration, the less effective is the power saving mechanism, and collisions occur more likely. Less battery power can be saved then.

The scenario with $T=20\text{ms}$ did not deliver reasonable results for wake ratios lower than 5%. Below this threshold, no traffic was possible anymore, because the wake periods A and B became too short. If the wake ratio is 4%, the wake period duration calculates as $(20\text{ms} * 4\%)/2 = 400\mu\text{s}$. If we assume a IEEE 802.11 slot time of $20\mu\text{s}$, the length of the initial wait period sums up to max. $190\mu\text{s}$. Hence, with very low wake ratios the duration wake periods become close to the time intervals that are necessary for the medium access layer to work properly.

4 Conclusions

In this paper we simulated and evaluated the power saving mechanism for a wireless ad hoc multi-hop scenario proposed in [1], integrated with AODV, and refined its forwarding strategies to make optimal use of the wake-up announcements. The results clearly state that unsynchronized power saving mechanisms can achieve sound connectivity properties even with low wake ratios, provided that a node can arbitrarily change its wake state without further loss of re-

sources.

Potential for optimization lies in the number and the instant of the RREQ rebroadcasts of intermediate nodes participating in a route establishment. Future work should focus on the reliability of the forwarding operations, and mechanisms to limit the overhead and additional wake time of the wake-ups that are necessary to reach neighboring nodes in their announced wake state.

Decreasing the cycle duration furthermore leads to lower end-to-end delays at the cost of better connectivity and reliability at lower wake ratios, and is limited to boundary values that are given by the medium access layer mechanisms and the properties of the radio hardware.

References

1. Torsten Braun, Laura Marie Feeney: *Power Saving in Wireless Ad hoc Networks without Synchronization*, 5th Scandinavian Workshop on Wireless Ad-hoc Networks, Stockholm, 2005
2. ANSI/IEEE Std 802.11 (ISO/IEC 8802-11), 1999 Edition (R2003)
3. Y.C Tseng, C.S. Hsu and T.Y. Hsieh: *Power Saving Protocols for IEEE 802.11-based Multihop Ad hoc Networks*, Proceedings of IEEE INFOCOM, New York, June 2003
4. Laura Marie Feeney: *A QoS aware Power Save Protocol For Wireless Ad Hoc Networks*, In Proceedings of the First Mediterranean Workshop on Ad Hoc Networks (Med-Hoc Net 2002), Sardinia, Italy, September 2002.
5. Laura Marie Feeney and Martin Nilsson: *Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment*, IEEE INFOCOM, 2001
6. Jean-Pierre Ebert, Brian Burns, and Adam Wolisz: *A trace-based approach for determining the energy consumption of a WLAN network interface*, Proceedings of European Wireless, Florence, Italy, February 2002
7. Charles E. Perkins, Elizabeth M. Belding-Royer: *Ad hoc On-Demand Distance Vector (AODV) Routing*, IETF Internet draft RFC 3561, October 2003
8. OMNeT++ Network Simulation Framework (<http://www.omnetpp.org>)
9. András Varga: *The OMNeT++ Discrete Event Simulation System*, Proceedings of the European Simulation Multiconference (ESM'2001) Prague, Czech Republic, June 2001
10. Mobility Framework for OMNeT++ (<http://mobility-fw.sourceforge.net>)
11. E. Osipov and Ch. Tschudin: *Improving the Path Optimality of Reactive Ad Hoc Routing Protocols Through De-Coherent RREQ waves*, Technical Report CS-2004-002, University of Basel, Switzerland
12. Jörg Rech, *Wireless LANs*, Heise Verlag, 2004
13. Jochen Schiller, *Mobile Communications*, Addison Wesley, 2003
14. O. Dousse, P. Thiran, and M. Hasler: *Connectivity in Ad-hoc and Hybrid Networks*, in Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02), New York, June 2002.
15. C. C. Enz, A. El-Hoiydi, J.-D. Decotignie, T. Melly, and V. Peiris: *WiseNET: An Ultralow-Power Wireless Sensor Network Solution*, IEEE Computer Magazine, Volume 37, Issue 8, August 2004