Optimizations for Route Discovery in Asynchronous Duty-Cycling Wireless Networks

Lei Tang* Yanjun Sun† Omer Gurewitz‡ David B. Johnson*

*Department of Computer Science, Rice University, Houston, TX, USA
†Systems and Applications R&D Center, Texas Instruments, Dallas, TX, USA
‡Department of Communication Systems Engineering, Ben Gurion University, Israel

***Abstract*—The use of *asynchronous duty cycling* at the MAC layer affords substantial energy savings in wireless networks. This technique is widely used in sensor networks and other types of wireless networks such as ad hoc networks. With asynchronous duty cycling, each node switches alternately between sleeping and active states; each node waking up asynchronously reduces network contention and wireless collisions caused by nodes waking up simultaneously, but also can have undesirable effects on higher layer protocols. In this paper, we study the problem of on-demand *route discovery* in asynchronous duty-cycling wireless networks and present four optimizations for such route discovery: *Delayed Selection*, *Duty-Cycled Selection*, *Reply Updating*, and *Adaptive Backoff*. Through detailed *ns-2* simulations, we show that, without these optimizations, the routes discovered in asynchronous duty-cycling networks can be over 50% longer than the theoretical shortest routes and can have an ETX 90% larger than the ETX of the optimal routes. With only simple changes made at the MAC or network layers, our optimizations enabled nodes to substantially improve discovered routes, finding routes that were only 0.2% longer than the theoretical shortest routes or routes with an ETX only 9% larger than the ETX of the theoretical optimal-ETX routes, while also reducing route discovery latency and node energy consumption.**

I. Introduction

Asynchronous duty cycling is a technique widely used for conserving energy in sensor networks (e.g., [1], [4], [5], [13], [14], [15]) as well as in other types of wireless networks such as ad hoc networks (e.g., [16], [17]). With asynchronous duty cycling, each node switches alternately between sleeping and active states and stays in sleeping state most of time, leading to substantial energy savings due to reduced idle listening. Furthermore, by allowing nodes to wake up *asynchronously*, asynchronous duty cycling also reduces network contention and wireless collisions that can be caused by nodes waking up simultaneously in synchronous duty cycling protocols.

On the other hand, asynchronous duty cycling imposes challenges on and can have unintended effects on higher layer protocols. For example, we have found that with standard on-demand *route discovery* in asynchronous duty-cycling wireless networks, the routes discovered can be over 50% longer than the shortest routes and have a route ETX that is 90% larger than the ETX of the optimal routes. This increase in route length and ETX (expected transmission count) [2] enlarges packet delivery latency and energy consumption while also reducing end-to-end packet delivery reliability.

One reason for the suboptimal route discovery in asynchronous duty-cycling wireless networks is due to the nodes' asynchronous wakeup timings. Many on-demand wireless route discovery protocols (e.g., the widely-used DSR [7] and AODV [10] protocols) discover the route between a source node and a destination node through broadcasting route discovery packets, forwarded by nodes in the network to reach the destination node. Asynchronous duty cycling allows nodes to wake up asynchronously, at independently chosen times for each node. Consequently, when a node broadcasts a Route Request packet, the Route Request packet reaches that node's neighbors at different times; as the packet is broadcast forwarded hop-by-hop, the different copies of the packet may travel at *very* different speeds along different paths. Moreover, in existing on-demand route discovery protocols, in order to reduce route discovery traffic, each node forwards only the *first* Route Request packet reaching it as part of each route discovery attempt; later arriving Route Request packets are discarded and not forwarded by the node. As a result, a Route Request packet reaching the route discovery destination often traverses a route largely determined by the nodes' wakeup timings, independent of the optimal route, often leading to the discovered route being significantly worse than optimal.

In this paper, we study possible optimizations for on-demand route discovery in asynchronous duty-cycling wireless networks and present four route discovery optimization techniques: *Delayed Selection*, *Duty-Cycled Selection*, *Reply Updating*, and *Adaptive Backoff*. Unlike existing work on energy-efficient routing (e.g., [11], [3], [8]) that extends the network lifetime through optimizing transmission power and node wakeup times, our route discovery optimization techniques seek to improve the route discovered between the source and the destination of a route discovery. By allowing such improved routes to be discovered, the handling of *data* packets along these routes is significantly improved, reducing energy consumption and packet delivery latency and also improving packet delivery ratio.

Our four route discovery optimizations can be easily integrated with existing wireless on-demand route discovery protocols without introducing additional routing messages among the nodes. The *Delayed Selection* optimization allows a node to filter out Route Request packets with inferior routes and to forward the Route Request containing what is normally the best route; this optimization is very effective in improving the routes discovered, although it adds a small delay that may be unsuitable for time-critical applications. The *Duty-Cycle Selection* optimization avoids the delay introduced in Delayed

978-1-4673-2433-5/12/$31.00 ©2012 IEEE

Selection. Instead, Duty-Cycle Selection sorts the received Route Requests within each duty cycle to select and forward the Request with the best route. The *Reply Updating* optimization, instead of optimizing the Route Requests forwarded, allows a node to optimize the route in a Route Reply packet, exploiting the routing information this node learns during forwarding the route discovery packets. Finally, based on the observation that after one wakeup, a node may receive multiple routing packets from different neighboring nodes at the same time, causing packet collisions, the *Adaptive Backoff* optimization enables different senders to adapt their packet transmission backoff values following any collision, for increasing the likelihood of the Route Request packet with the best route being delivered first.

We have implemented these four route discovery optimizations in the *ns-2* network simulator and conducted extensive simulations to evaluate their performance on two widely-used routing metrics (route hop-length and route ETX) under various network scenarios. We found that these route discovery optimizations substantially improved the routes discovered in asynchronous duty-cycling networks, while also reducing route discovery latency and node duty cycle.

The rest of the paper is organized as follows. Section II provides background and reviews the related routing and MAC protocols in asynchronous duty-cycling wireless networks. Section III describes more fully the suboptimal route discovery problem. In Section IV, we present four our optimizations for improving route discovery in asynchronous duty-cycling wireless networks. Section V presents our evaluation of these optimizations, and Section VI presents conclusions.

II. Background and Related Work

In a wireless network, a route between two nodes can be discovered by two approaches. With *on-demand* (or *reactive*) route discovery, a source initiates a route discovery request for a destination only when needed; the request is forwarded by being broadcasted by each node until reaching the destination. Instead, with *periodic* (or *proactive*) routing, the route between the source and each destination is computed based on the network topology information proactively disseminated among the nodes in the network.

Compared with periodic routing protocols (e.g., DSDV [9]), on-demand routing protocols (e.g., DSR [7] and AODV [10]) generally have lower overhead and are more adaptive to changing network topology such as due to node movement or failure or wireless propagation changes. Furthermore, researchers have recently begun standardizing on-demand route discovery for duty-cycling sensor networks [6]. Therefore, we focus on optimizing the route discovery of the on-demand routing protocols in duty-cycling wireless networks.

Wireless transceiving accounts for a major source of energy consumption for nodes in a wireless network. To conserve energy, duty-cycling techniques are utilized in many wireless networks, enabling each node to remain in sleeping state most of the time and only periodically wake up for transceiving packets. The *duty cycle* measures the percent of time that a node is active. Among existing duty-cycling techniques, *asynchronous* duty-cycling techniques [5], [1], [13], [4], [15], [14], [17], [16] are widely used in sensor networks and ad hoc networks, because they allow nodes to independently decide their wakeup times, achieving high energy efficiency while efficiently handling dynamic traffic without requiring global clock synchronization. Hence, our route discovery optimizations are targeted at route discovery in *asynchronous* duty-cycling wireless networks.

With a on-demand routing protocol, a route discovery is only initiated when needed, on-demand. Take, for example, the route discovery mechanism of the DSR protocol. When a node has a data packet to send to another node but does not have a route to it, the source node initiates a route discovery for that destination node by broadcasting a Route Request packet. Upon receiving a Route Request packet, if a non-destination node has not yet forwarded a Route Request packet from this source for this route discovery attempt, it adds its own address to the route embedded in the Route Request packet and rebroadcasts the packet. When the Route Request packet reaches the destination, the destination unicasts a Route Reply packet to the source node following the reverse of the route contained in the Route Request packet. Once the source receives a Route Reply packet from the destination, it uses the route in the Route Reply packet to send its data packets.

Although on-demand routing protocols have been studied extensively, there has been little study on the performance of the on-demand routing protocols in asynchronous duty-cycling networks. In this paper, we study the suboptimal route discovery problem of on-demand routing protocols in asynchronous duty-cycling wireless networks and present four optimization techniques to solve this problem.

Not until recently did researcher start to improve routing in asynchronous duty-cycling networks. Lai and Ravindran [8] modeled an asynchronous duty-cycling network as a time-dependent network and presented an algorithm to compute the shortest path between nodes in the network. Their algorithm differs from our route discovery optimizations in a number of ways. First, similar to DSDV, their algorithm requires a node to maintain a distance vector to all nodes in the network. Second, their algorithm requires the knowledge of link delay and the wakeup times of every node in the network. Considering factors such as node mobility and clock drift, obtaining such knowledge in a wireless network is not only costly but also infeasible for the asynchronous duty-cycling protocols that adaptively decide the wakeup times and wakeup channels (e.g., EM-MAC [14]). In contrast, our route discovery optimizations are independent of network topology and node wakeup times and can be easily integrated with existing routing protocols without adding extra routing message exchanges.

There are also duty-cycling routing protocols designed to improve energy efficiency. For example, the energy aware routing protocol [11] probabilistically chooses a route for data forwarding to achieve extended network lifetime. The techniques in such prior work differ from our approach since our optimizations are designed to improve the route discovery in duty-cycling networks, optimizing the routing metric (e.g.,

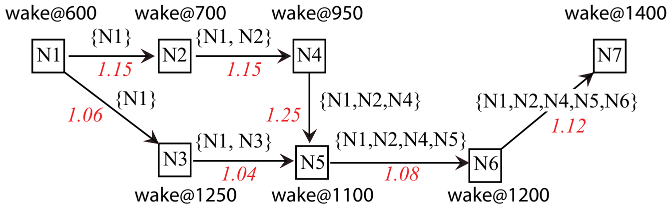

Fig. 1. Example of route discovery discovering a suboptimal route in an asynchronous duty-cycling network. Node 1 is the source and node 7 is the destination. The wakeup time in milliseconds is marked next to each node. The number below each link denotes the link's ETX.

route length or ETX) for the routes discovered.

III. The Suboptimal Route Discovery Problem

In conventional always-on networks, a node can immediately forward the received on-demand route discovery packet as its neighboring nodes are always active. Therefore, the routes discovered between a source and destination node are primarily determined by the shortest route between them and the traffic load and wireless channel condition of the nodes between them.

However, the routes discovered in an asynchronous duty-cycling network are greatly affected by nodes' wakeup timings. With a node waking up at independently chosen times, a broadcasted route discovery packet in asynchronous duty-cycling networks reaches its receivers at different times and a Route Request packet with a suboptimal route can arrive at a node before does a packet with the optimal route. Furthermore, to mitigate route discovery traffic, once a node broadcasts a route discovery packet, it will ignore the route discovery packets received later that are originated from the same source for the same destination. Thus, in asynchronous duty-cycling networks, existing routing protocols often discover suboptimal routes, enlarging the delivery latency of the data packets and the energy spent on forwarding the data packets, while reducing the end-to-end packet delivery reliability.

Figure 1 shows an example of conventional on-demand routing protocols discovering a suboptimal route in an asynchronous duty-cycling network, in which node N1 initiates a route discovery for destination node N7. The shortest route and the best-ETX route between N1 and N7 should be N1→N3→N5→N6→N7 but the route discovered is N1→N2→N4→N5→N6→N7. The suboptimal route discovered in this example is due to asynchronous node wakeup timing: N3 wakes up later than N2, N4 and N5, so a suboptimal route N1→N2→N4 reaches N5 before does the route N1→N3.

With conventional on-demand routing protocols, to discover the optimal route between a source and destination in an asynchronous duty-cycling network, every intermediate node on the optimal route must receive the route discovery packet containing the segment of the optimal route from the source to this node before it forwards the first discovery packet. This becomes increasingly difficult as the distance between the source and the destination increases, since with the simple receive-then-forward paradigm of conventional on-demand routing protocols, the probability of a route discovery packet wandering into a suboptimal route due to asynchronous node wakeup timing increases together with the hop-length of the optimal route between the route discovery source and destination.

For example, using the hop-length of the route discovered, our simulations in *ns-2* network simulator show that when the hop-length of the theoretical shortest route between the route discovery source and destination increases, so does the difference between the hop-length of a route discovered by a conventional routing protocol and the hop-length of the theoretical shortest route.

In our simulations, we use the DSR routing protocol [7] and RI-MAC asynchronous duty-cycling MAC protocol [13]. The basic DSR route discovery mechanism summarized in Section II is used. Since the goal of these simulations is to evaluate the behavior of route discovery, advanced DSR features such as returning a route stored in the route cache and flow-based routing are not used. The collision resolution mechanism of RI-MAC is used to resolve wireless collisions, with which a receiver detecting a wireless collision informs the senders to retransmit the packets using an increased congestion window. To support the broadcast of route discovery packets in asynchronous duty-cycling networks, we use the broadcast packet transmission mechanism of RI-MAC, with which a node sends a broadcast packet to its neighboring nodes one-by-one when a neighbor wakes up. In RI-MAC, each node wakes up asynchronously at randomly chosen times, with wakeup intervals chosen randomly between 0.5 times and 1.5 times the nominal cycle length (e.g., the nominal cycle length in our simulations is 1 second); the node then sends a wakeup beacon to declare it is awake and ready to receive packets. Since a node wakes up at least once during the maximum wakeup interval (i.e., 1.5 times the nominal cycle length), a broadcast packet sender in our implementation stays awake for the maximum node wakeup interval and sends the broadcast packet to every node from which it receives a wakeup beacon. After sending a broadcast packet and receiving the ACK beacon for the broadcast packet from the receiver, the broadcast sender records that the receiver has received this broadcast packet so it will not send duplicate broadcast packets to the receiver.

Figure 2 shows the length of the *first route* and the *minimum-length route* returned to the route discovery source during each route discovery in 1000 m × 1000 m duty-cycling networks, each with 100 randomly placed nodes. Even though a node forwards each route discovery packet only once, the destination of a route discovery may receive multiple route discovery packets, each including a different route from the source of the discovery to the destination. The *first route* is the route in the first Route Reply packet received by the discovery source, whereas the *minimum-length route* is the shortest route discovered during each discovery. The *minimum-length route* differs from the *theoretical shortest route*, as the latter is calculated using Floyd-Warshall algorithm based on the network topology and the radio transmission range (250 m). The first route length is an important metric since the first discovered route is the route used by the source to send the packets currently in its queue to the destination.

The x-axis of the figure shows the hop-length of the theoretical shortest route between the randomly chosen route

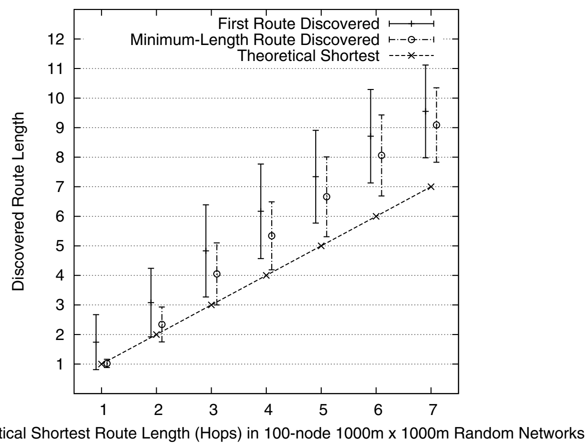

Fig. 2. Average and standard deviation of the hop-length of the first route and the minimum-length route discovered for each route discovery in the random duty-cycling networks using standard DSR route discovery.

discovery source and destination. The length of the theoretical shortest routes in the random networks ranges from 1 to 7. In the simulations, for each theoretical shortest route length l ($1 \leq l \leq 7$), 100 route discovery source and destination pairs are randomly chosen from 100 random networks, provided that the length of the theoretical shortest route between the source and destination of a chosen pair of nodes equals l. The position of a node varies from one simulation run to another. For each chosen pair of nodes, a route discovery is launched by the route discovery source to find the route to the destination.

The error bar shows the average and the standard deviation of the hop-length of the first routes and the minimum routes, which are compared with the hop-length of the corresponding theoretical shortest route. The simulation results indicate that the routes discovered by conventional DSR routing protocol in asynchronous duty-cycling networks are much longer than the theoretical shortest routes: The first routes discovered by conventional DSR routing protocol were on average 53% longer than the theoretical shortest routes. Only 21% of the first routes were of the same length as the theoretical shortest routes while 47% of the first routes were at least 50% longer than the theoretical shortest routes. 20% of the first routes were even at least twice as long as the theoretical shortest routes.

As will be shown in Section V, conventional on-demand route discovery also leads to suboptimal ETX routes in asynchronous duty-cycling networks, which we show can be 90% worse than the optimal ETX routes.

IV. Route Discovery Optimizations

We present four route discovery optimizations for asynchronous duty-cycling wireless networks. These optimizations are fully distributed because they only use the route information in the route discovery packets received or overheard without requiring global routing knowledge. They are also computationally efficient and can be easily integrated with existing wireless routing protocols without introducing extra routing message exchanges. Further, these optimizations are orthogonal and can be applied separately or in any combination. Due to space limitations, we focus on using these optimizations on two widely used routing metrics: route hop-length and route ETX.

A. The Delayed Selection Optimization

One reason for suboptimal route discovery in duty-cycling networks is that a Route Request with suboptimal route may arrive at a node earlier than does the optimal one due to the asynchronous node wakeup timing. The rationale behind our *Delayed Selection* optimization is to filter out the Route Request packets containing inferior routes and only forward the packet containing the route that is normally the optimal route.

With Delayed Selection, a node buffers a received Route Request before selecting which Route Request to broadcast to its neighboring nodes. If while buffering the packet, the node receives a Route Request that belongs to the same route discovery attempt but contains a better route than the route in the currently buffered Route Request, the node discards the currently buffered packet and replaces it with the newly received packet. The route hop-length and ETX information used during the selection are obtained from the route discovery packets received. The ETX of each hop on the route in a Route Request is added to the packet by every forwarder: Upon receiving a Route Request packet from another node S, node R appends the ETX between S and R to the received packet before forwarding the packet. When the Route Request packet reaches the route discovery destination, the destination sends a Route Reply packet to the route discovery source, which includes the route discovered and the route ETX information.

The buffering of a Route Request, however, must be limited in time, yet a small buffering time may not be sufficient for the node to receive better routes from other nodes, while a large buffering time unnecessarily enlarges the route discovery latency. The Delayed Selection optimization adaptively computes the buffering time of a Route Request packet based on the length of the route contained in the packet and the total buffering time of this packet at its previous forwarders.

Let $I_{\max}$ be the maximum node wakeup interval (e.g., 1500 ms, by default, in RI-MAC [13]). Let N_{hops} be the hop-length of the route in a newly received Route Request packet, with T_{buf} being the total buffering time of this packet on previous forwarders. If $T_{buf} \geq I_{max} \times N_{hops}$, the node immediately stops buffering this packet and broadcasts it (if it has not already forwarded a Route Request belonging to the same discovery attempt). Otherwise, the node begins buffering this packet, which can last as long as $I_{max} \times N_{hops} - T_{buf}$. If while buffering this packet, the node receives another Route Request packet with a better route, the node replaces the currently buffered packet with the new one and tests if the new packet needs to be buffered. Here we use the example in Figure 1 to illustrate Delayed Selection filtering out suboptimal routes. In the example, the $I_{\max}$ is 1500 ms and there are two distinct routes between source N1 and forwarder N5: N1→ N2 → N4 → N5 and N1→ N3 → N5. Without Delayed Selection, due to the nodes' wakeup timings, the Route Request packet containing the suboptimal route arrives at N5 first and is forwarded by N5, while the later received Route Request packet containing the optimal route is not forwarded. In contrast, with

Delayed Selection, owing to the length-aware Route Request packet buffering mechanism, N5 filters out the suboptimal route and only forwards the Route Request containing the optimal route, thereby allowing the optimal route to be discovered.

The following proposition uses the route hop-length metric as an example to show the effectiveness of Delayed Selection in optimizing route discovery:

Proposition: Assuming there is no packet loss and packet backlog, Delayed Selection guarantees that the route in a Route Request packet forwarded by a node must be a shortest route from the source of the route discovery to this node.

Proof: By induction. 1) When the hop-length of the shortest route between a Route Request packet forwarder and the source of the route discovery is 1, the statement holds because by I_{max} time units, the forwarder has waken up at least once and received the packet from the source. 2) Assume the proposition holds when the hop-length of the shortest route between a Route Request packet forwarder and the source is k ($k \geq 1$). That is to say, by $I_{max} \times k$ time units, all nodes whose shortest route to the source is of hop-length k have received a Route Request packet containing a route of hop-length k. Based on the induction hypothesis, an additional I_{max} time units guarantee that every neighbor of these nodes will wake up at least once and receive a Route Request packet containing a route of hop-length k. Hence, when the hop-length of the shortest route between a Route Request packet forwarder and the source is $k+1$, by $I_{max} \times (k+1)$ time units, the forwarder must have received a Route Request packet from an upstream forwarder containing a route of hop-length k. With the length-aware packet buffering mechanism, the Route Request packet forwarded by this node contains a shortest route of hop-length $k+1$ to the source. Proof ends.

In practice, factors such as wireless collision and network congestion may lower the effectiveness of Delayed Selection, which will be discussed in Section V.

Delayed Selection is easy to implement without requiring global time synchronization. In our implementation, the total buffering time of a Route Request packet (i.e., T_{buf}) is stored in the packet header. When a node forwards a Route Request packet, it updates the T_{buf} of the packet by adding the difference between the current time and the time the packet was received. If the packet transmission fails, when the node retransmits the packet, the total buffering time of the packet is likewise updated.

B. The Duty-Cycled Selection Optimization

The *Duty-Cycled Selection* optimization is motivated by two observations on forwarding the Route Request packets in asynchronous duty-cycling networks. First, due to duty cycling, when a node wakes up, multiple neighbors of this node may have accumulated route discovery packets and began forwarding them to this node at the same time, resulting in a sudden influx of packets to this node. The order of forwarding the newly received route discovery packets impacts the qualify of the routes discovered. Second, due to the bursty traffic characteristic of duty cycling networks, the receive-then-forward paradigm of the conventional routing protocols often interferes with the reception of the route discovery packets, causing packet losses and suboptimal route discovery.

Hence, with the Duty-Cycled Selection optimization, after a node wakes up, it will not commence forwarding packets until it finishes receiving packets from other nodes; this not only allows the node to potentially collect more Route Request packets before deciding which one to forward but also prevents packet transmissions from interfering with packet receptions. Unlike Delayed Selection, Duty-Cycled Selection only requires a node to withhold forwarding packets until the end of receiving all packets during a wakeup.

Furthermore, Duty-Cycled Selection optimizes the forwarding of the packets received during a wakeup by prioritizing forwarding the route discovery packets containing superior routes, thereby enabling superior routes to be discovered and to arrive at intermediate forwarders and the route discovery source more quickly. Specifically, for the Route Request packets received in a wakeup that belong to the same route discovery, the Duty-Cycled Selection optimization selects and only forwards the Route Request packet containing the best route (e.g., the route with the best ETX or with the shortest hop-length). In addition, it sorts the Route Reply packets based on their routing metric (e.g., ETX or hop-length) and forwards the packets with the best routing metric first.

C. The Reply Updating Optimization

In duty-cycling networks, between forwarding a Route Request packet toward the route discovery destination and forwarding the corresponding Route Reply packet toward the route discovery source, a node may have learned a better route to the route discovery source and destination from the later received and overheard route discovery packets. Based on this observation, when a Route Reply is sent back to the route discovery source following the reverse of a route received by the destination, the *Reply Updating* technique enables an intermediate forwarder to optimize the route in every Route Reply packet using the route information learned by this node.

Specifically, upon receiving or overhearing a route discovery packet, a node records in its local route cache the best routes (e.g., best-ETX routes and shortest-hop-length routes) to the nodes on the route in the packet. When a node receives a Route Reply packet, if the route embedded in it includes a suboptimal route from the node to the source or the destination of this route discovery as opposed to the best route known by this node, the node replaces the suboptimal route in the packet with the best route known. For example, in Figure 1, by the time N5 receives the Route Reply packet from N6, it has received a Route Request packet forwarded by N3 and learned the best route from N1 to N5, (i.e., N1→N3→N5), which enables N5 to use the best route to replace the inferior route N1→N2→N4→N5 in the Route Reply packet. The resulting route N1→N3→N5→N6→N7 has a lower ETX and shorter hop-length than the original route in the Route Reply packet, which is N1→N2→N4→N5→N6→N7.

D. The Adaptive Backoff Optimization

When a node wakes up, multiple neighboring nodes may send packets to the node at the same time, which often leads to wireless collisions and packet retransmissions. In order to quickly resolve the collisions and to prioritize sending Route Request packets that contain high quality routes, our *Adaptive Backoff* optimization enables a Route Request packet sender to choose its backoff time for the packet transmission based on the quality of the route in the packet.

Let h be the hop length and ETX be the ETX of a Route Request packet. Let CW be the congestion window size, $rand$ be a random integer, and $slotTime$ be the unit of backoff time. If the goal is to discover routes with the shortest hop-length, the backoff time for transmitting a Route Request packet is computed as follows:

$$(min(h, 10) \times \frac{CW}{10} + rand \,\%\, CW) \times slotTime.$$

If the goal instead is to discover routes with the smallest ETX, backoff time is computed as follows (Γ is a configurable parameter that denotes the maximum route ETX):

$$(min(ETX, \Gamma) \times \frac{CW}{\Gamma} + rand \,\%\, CW) \times slotTime.$$

Given the first component of these two equations (i.e., $min(h, 10) \times \frac{CW}{10} \times slotTime$ and $min(ETX, \Gamma) \times \frac{CW}{\Gamma} \times slotTime$, respectively), the sender of a Route Request containing a route with small ETX or hop-length backs off less on average than does the sender of a Route Request containing a route with large ETX or hop-length. The second component, $(rand \,\%\, CW) \times slotTime$, spreads the backoff times of different senders, including nodes sending Route Reply and data packets. To allow a receiver to receive the packets, the receiver stays awake for a little longer than $2 \times CW \times slotTime$ to wait for the transmissions.

The backoff time for sending a data packet or a Route Reply packet is computed simply as $(rand \,\%\, CW) \times slotTime$, which makes a data or Route Reply packet more likely to win wireless medium during packet retransmissions than a Route Request packet. During packet retransmissions, Adaptive Backoff favors data and Route Reply packets as they have less redundancy than Route Request packets.

V. EVALUATION

A. Simulation Methodology

We evaluated our route discovery optimizations through extensive simulations using the *ns-2* network simulator in two types of asynchronous duty-cycling wireless networks: *random* and *grid*. The random networks cover $1000\,\text{m} \times 1000\,\text{m}$ with 100 randomly-placed nodes. The grid network is a 10×10 grid of 100 nodes, in which a node is within the transmission range (250 m) of only its two horizontal and two vertical neighbors. Compared with the random networks, the grid network is much sparser; on average, a node in the grid network has 3.6 neighbors, while in the random networks, a node on average has 15.6 neighbors.

In the simulations, each node wakes up at independently chosen random times, with random wakeup intervals ranging from 500 ms to 1500 ms; a node wakes up once per second on average. If this wakeup interval length is decreased, route discovery latency decreases while node duty cycle increases.

We evaluated eight combinations of our route discovery optimizations, as well as the existing basic DSR route discovery procedure (denoted as *no-optimization*). In each figure below, *DS* denotes Delayed Selection, *DCS* denotes Duty-Cycled Selection, *RU* denotes Reply Updating, and *AB* denotes Adaptive Backoff; combinations of our route discovery optimizations are denoted by combinations of these abbreviations (e.g., *DS-DCS-RU-AB* denotes the combination of all four optimizations). The MAC protocol used in our simulations is RI-MAC [13].

We evaluated the following six metrics: *first route length* is the hop-length of the first route discovered during each route discovery; *first route latency* is the time taken to discover this first route; *minimum route length* is the hop-length of the minimum-length route discovered during each route discovery; and *minimum route latency* is the time taken to discover this minimum-length route; *minimum route ETX* is the ETX of the minimum-ETX route discovered during each route discovery; and *node duty cycle* is the ratio of time a node is active to the entire simulation time.

In our simulations, we estimated the ETX between each pair of nodes based on the simulated RSSI over that link, using the packet delivery ratio correlation results presented by Srinivasan and Levis [12]. This technique could be used in an actual deployed network based on historical measurements of the RSSI of normally received packets, and could be supplemented by having a node occasionally remain awake for a full cycle (or more) to obtain additional RSSI samples from the node's neighbors (e.g., from those neighbors' wakeup beacons). It is also possible to use explicit probing to measure ETX, although this is less desirable due to the increased energy and network bandwidth consumption required.

The source and destination of each route discovery in our simulations are randomly chosen. For both the hop-length and ETX metrics, the results shown are organized according to the hop-length of the theoretical shortest route between this source and destination. The hop-length of the theoretical shortest route between any two nodes in the random networks ranges from 1 to 7, and in the grid network, from 2 to 18. When evaluating a route discovery optimization, for every possible length l of theoretical shortest routes, 100 route discovery source and destination pairs are randomly chosen with the length of the theoretical shortest routes between the source and destination of all chosen pairs being l. For each chosen pair of nodes, a route discovery is initiated by the source to find a route to the corresponding destination. The average and standard deviation of the measured metrics are shown in the figures below.

B. Results in Random Networks

Figure 3 shows the hop-length of the first routes discovered in the random networks. Without our route discovery optimizations, the routes discovered by DSR in duty-cycling networks were substantially longer than the theoretical shortest routes: the first routes discovered were on average 53% longer than the

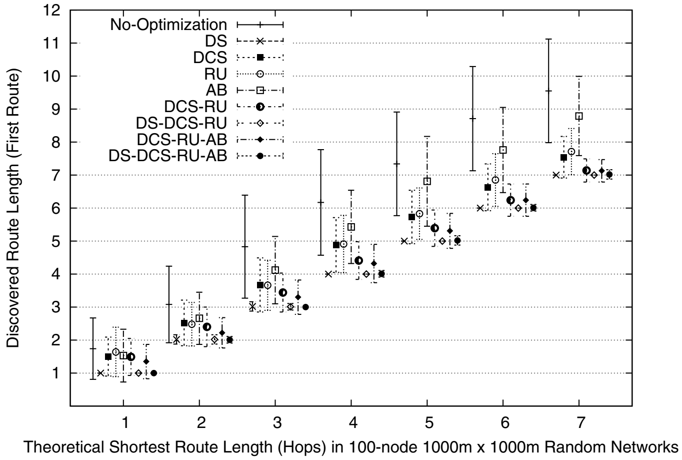

Fig. 3. Hop-length of the first routes discovered in 100-node random 1000 m × 1000 m networks.

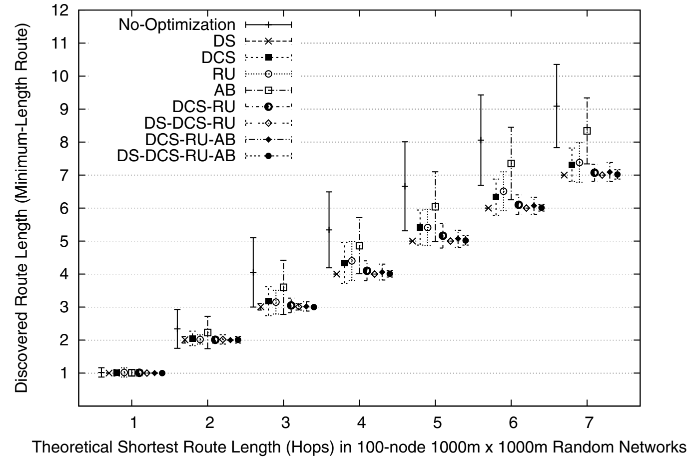

Fig. 4. Hop-length of the minimum-length routes discovered in 100-node random 1000 m × 1000 m networks.

theoretical shortest routes; only 21% of the first routes were of the same length as the theoretical shortest routes.

In contrast, the first routes discovered when using all our route discovery optimizations were only 0.2% longer than the theoretical shortest routes. Among all first routes discovered by the Delayed Selection optimization, more than 99% of them were of the same hop-length as the theoretical shortest routes and the rest of them ($< 1\%$) were only one hop longer than the theoretical shortest routes. The rare occasion that Delayed Selection failed to discover a theoretical shortest route was due to the wireless collisions that caused the loss of the Route Request packet following the theoretical shortest route. All route discovery optimization combinations that included Delayed Selection achieved a similar high performance on discovering the first routes, but other combinations also significantly improved the first routes.

Figure 4 shows the hop-length of the minimum-length routes discovered in the random networks. As the minimum-length route is the shortest route discovered in a route discovery, the minimum-length routes are at least as short as the first routes discovered but no shorter than the theoretical shortest routes. The minimum-length routes discovered by DSR without using our optimizations (i.e., *no-optimization*) were on average 26% longer than the theoretical shortest routes. Furthermore, as the hop-length of the theoretical shortest route increases, so does the gap between the hop-length of the minimum-length routes discovered by no-optimization and the hop-length of the theoretical shortest route. This is because with no-optimization, when the hop-length of the theoretical shortest route increases, a Route Request packet is more likely to experience detours due to asynchronous node wakeup timings.

Compared with *no-optimization*, our route discovery optimizations were able to discover significantly better minimum-length routes. The route discovery optimization combinations that included Delayed Selection found the minimum-length routes with the same hop-length as the theoretical shortest routes more than 99% of time, and the average hop-length of the minimum-length routes discovered by these route discovery

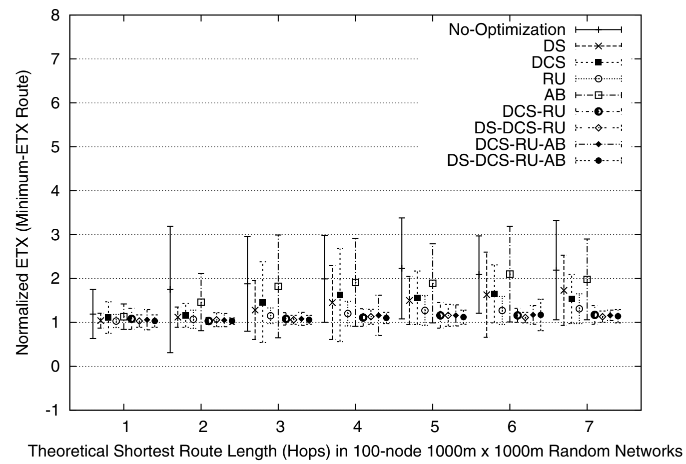

Fig. 5. Normalized ETX of the minimum-ETX routes discovered in 100-node random 1000 m × 1000 m networks.

optimization combinations was only 0.1% longer than the theoretical shortest routes. The combination of Duty-Cycled Selection, Reply Updating, and Adaptive Backoff achieved high performance too, discovering minimum-length routes with the same hop-length as the theoretical shortest routes 96% of time. The average hop-length of the minimum-length routes discovered by these three optimizations was less than 1% longer than the theoretical shortest routes.

Figure 5 shows the ETX of the minimum-ETX routes discovered in the random networks. The ETX of a route is presented normalized by the ETX of the theoretical optimal-ETX route between this source and destination. In the random networks, when no-optimization was used, the ETX of the minimum-ETX routes discovered was on average 90% larger than that of the theoretical optimal-ETX routes. In sharp contrast, with all our route optimizations enabled, the ETX of the minimum-ETX routes discovered was only 9% larger than that of the theoretical optimal-ETX routes. With DCS-RU-AB enabled, the ETX of the minimum-ETX routes discovered was on average 12% larger than that of the theoretical optimal-ETX routes.

Figure 6 shows the latency of discovering the first routes in random networks. The route discovery optimization combination of Duty-Cycled Selection, Reply Updating, and Adaptive Backoff achieved the smallest first route discovery latency for two reasons. First, these route discovery optimizations shortened the route discovered so a route discovery packet traversed fewer hops. Second, the Duty-Cycled Selection optimization prioritized sending the route discovery packets that contained short routes, which consequently sped up the discovery of the short first routes. When the theoretical shortest route length was 1 (i.e., the source and destination were neighbors), the first route discovery latency of the optimization combinations including Delayed Selection was less than that for the other optimization combinations because with the Delayed Selection optimization, the neighboring nodes of the source that received a Route Request packet would postpone forwarding the received packet until the source had finished broadcasting its Route Request packet, thereby reducing the wireless collisions caused by the source and another neighbor of the destination sending the Route Request packets to the destination at the same time. With no-optimization, once a node received a Route Request packet from the source, it immediately began rebroadcasting the packet, which in this case often interfered with the destination receiving the Route Request packet from the source, increasing the route discovery latency. As the theoretical shortest route length between the route discovery source and destination rose, the first route discovery latency of the Delayed Selection optimization surpassed that for other route optimizations because Delayed Selection waited for potentially better routes at each intermediate forwarder.

Figure 7 shows the average node duty cycle in the random networks. The average node duty cycle was low since a node was in sleeping state most of the time. However, when using our route discovery optimizations, node duty cycle was further reduced since the optimizations decrease the length of the routes discovered, reducing the number of transmissions needed to forward the packets.

C. Results in Grid Network

Figure 8 show the length of the minimum-length routes discovered in the grid duty-cycling network. The grid network is much sparser than the random networks, thereby having a far lower route diversity and less wireless collisions than the random networks. In the grid network, our route discovery optimizations also achieved very good results. 99% of the first routes and 100% of the minimum-length routes discovered by the Delayed Selection optimization were of the same length as the theoretical shortest routes. 96% of the minimum-length routes discovered by the combination of Duty-Cycled Selection, Reply Updating and Adaptive Backoff were of the same length as the theoretical shortest routes.

Compared with no-optimization, our route discovery optimizations lowered node duty cycle in the grid network since they shortened the routes discovered and improved the forwarding of route discovery packets. Owing to the node sparsity in the grid network, a node had fewer route discovery packets to forward, so the average node duty cycle of the grid network was lower than that of the random networks.

Figure 9 shows the normalized ETX of the minimum-ETX routes discovered in the grid network. In the grid network, when no-optimization was used, the ETX of the minimum-ETX routes discovered was on average 9% larger than that of the theoretical optimal-ETX routes. The gap between the ETX of the minimum-ETX routes discovered in the grid network and the ETX of the theoretical optimal-ETX routes was smaller than the gap between these two metrics in the random networks, which was attributed to the sparsity of the grid network. On average, a node in the grid network has 3.6 neighbors while a node in the random networks has 15.6 neighbors. Thus, compared with the random networks, the grid network had far fewer feasible routes between a source node and a destination node so the route discovery packets in the grid network were much less likely to follow suboptimal routes. For example, when the route discovery source and destination were neighbors in the grid network, the minimum-ETX route discovered between them was often directly from the source to the destination. But if the source and the destination were neighbors in the random networks, there were far more feasible routes between them, and depending on the nodes' wakeup timings, the minimum-ETX routes discovered can be much worse than the theoretical optimal-ETX routes.

In contrast to no-optimization, our route discovery optimizations were very effective in discovering routes with near optimal ETX: When all four optimizations were enabled, the minimum-ETX routes discovered were identical to the theoretical optimal-ETX routes. With only DCS-RU-AB enabled, the ETX of the minimum-ETX routes discovered was only 1% larger than that of the theoretical optimal-ETX routes.

VI. Conclusion

In this paper, we presented four optimizations for on-demand route discovery in asynchronous duty-cycling wireless networks: Delayed Selection, Duty-Cycled Selection, Reply Updating, and Adaptive Backoff. These techniques are fully distributed and can be easily applied to existing route discovery protocols without introducing extra routing message exchanges.

We have evaluated the performance of the combinations of these route discovery optimization techniques on two widely-used routing metrics (route hop-length and route ETX) through extensive *ns-2* simulations in asynchronous duty-cycling networks. Compared with the conventional route discovery protocol, these optimization techniques substantially improved the routes discovered. For example, whereas the routes discovered in the random networks by the conventional route discovery protocol were over 50% longer hop count than the theoretical shortest routes and had an ETX 90% larger than the ETX of the optimal-ETX routes, the routes discovered using our optimization techniques were only 0.2% longer hop count than the theoretical shortest routes and had an ETX only 9% larger than the ETX of the theoretical optimal-ETX routes. Our route discovery optimizations also lowered the route discovery latency and node energy consumption over the conventional route discovery protocol. Furthermore, as the network density and route length increased, the performance advantage of our

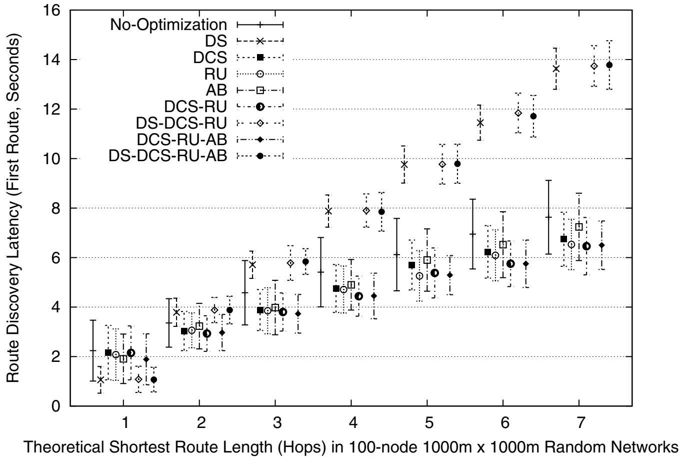

Fig. 6. Discovery latency of the first routes in 100-node random $1000\,\mathrm{m} \times 1000\,\mathrm{m}$ networks.

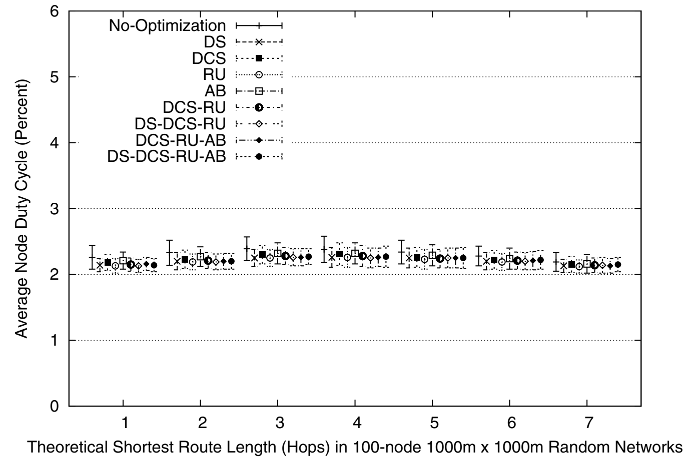

Fig. 7. Average node duty cycle in 100-node random $1000\,\mathrm{m} \times 1000\,\mathrm{m}$ networks.

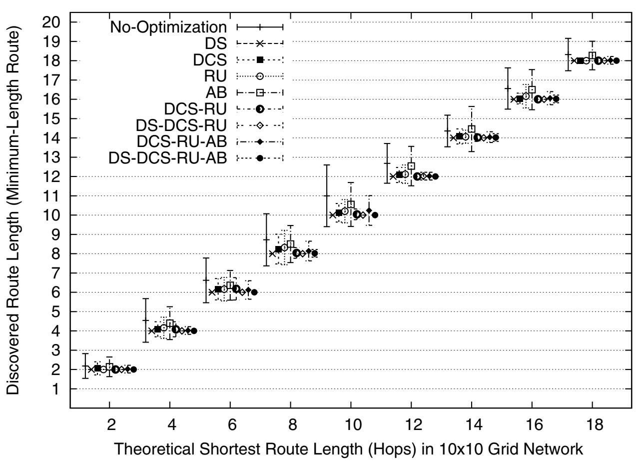

Fig. 8. Hop-length of the minimum-length routes discovered in the 10×10 grid network.

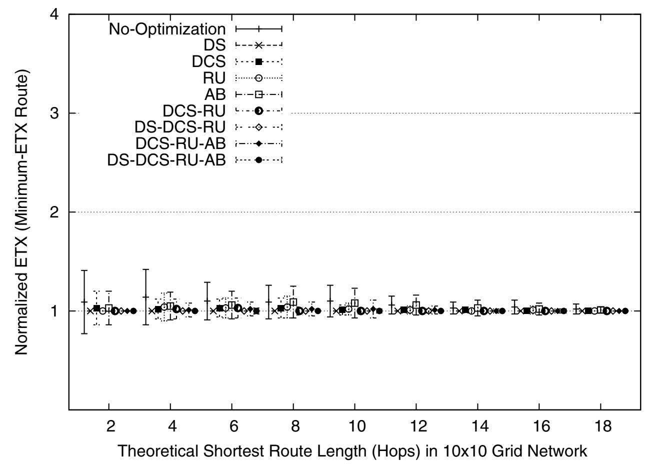

Fig. 9. Normalized ETX of the minimum-ETX routes discovered in the 10×10 grid network.

optimization techniques over the conventional route discovery protocol likewise increased.

REFERENCES

[1] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *SenSys 2006*, pages 307–320, November 2006.

[2] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom 2003*, pages 134–146, September 2003.

[3] Shu Du, Amit Kumar Saha, and David B. Johnson. RMAC: A routing-enhanced duty-cycle mac protocol for wireless sensor networks. In *INFOCOM 2007*, pages 1478–1486, May 2007.

[4] Prabal Dutta, Stephen Dawson-Haggerty, Yin Chen, Chieh-Jan Mike Liang, and Andreas Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *SenSys 2010*, pages 1–14, November 2010.

[5] Amre El-Hoiydi and Jean-Dominique Decotignie. WiseMAC: An ultra low power MAC protocol for multi-hop wireless sensor networks. In *ALGOSENSORS 2004*, pages 18–31, July 2004.

[6] Mukul Goyal, Emmanuel Baccelli, Matthias Philipp, Anders Brandt, and Robert Cragie. Reactive discovery of point-to-point routes in low power and lossy networks. Internet Draft draft-ietf-roll-p2p-rpl-04, work in progress, 2011.

[7] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181, 1996.

[8] Shouwen Lai and Binoy Ravindran. On distributed time-dependent shortest paths over duty-cycled wireless sensor networks. In *INFOCOM 2010*, pages 1685–1693, March 2010.

[9] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM 1994*, pages 234–244, August 1994.

[10] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In *IEEE WMCSA*, pages 90–100, 1999.

[11] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 350–355, March 2002.

[12] Kannan Srinivasan and Philip Levis. Rssi is under appreciated. In *Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets)*, May 2006.

[13] Yanjun Sun, Omer Gurewitz, and David B. Johnson. RI-MAC: A receiver initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *SenSys 2008*, pages 1–14, November 2008.

[14] Lei Tang, Yanjun Sun, Omer Gurewitz, and David B. Johnson. EM-MAC: A dynamic multichannel energy-efficient mac protocol for wireless sensor networks. In *MobiHoc 2011*, May 2011.

[15] Lei Tang, Yanjun Sun, Omer Gurewitz, and David B. Johnson. PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks. In *INFOCOM 2011*, pages 1305–1313, April 2011.

[16] Yu-Chee Tseng, Chih-Shun Hsu, and Ten-Yueng Hsieh. Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. *Computer Networks*, 43:317–337, October 2003.

[17] Rong Zheng, Jennifer C. Hou, and Lui Sha. Asynchronous wakeup for ad hoc networks. In *MobiHoc 2003*, pages 35–45, June 2003.