

Energy and latency performance of geographic random forwarding for ad hoc and sensor networks

Michele Zorzi[†] and Ramesh R. Rao[‡]

[†]UdR CNIT, Università di Ferrara – via Saragat, 1 – 44100 Ferrara, Italy

ph.: +39-0532-974840 – fax: +39-0532-974870 – e-mail: zorzi@ing.unife.it

[‡]Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407, USA

Abstract—In this paper, we describe a novel forwarding technique based on geographical location of the nodes involved and random selection of the relaying node via contention among receivers. A collision avoidance scheme based on this idea is described in detail, and an approximate analysis is provided. The proposed scheme is compared with STEM, and is shown to perform significantly better for sufficient node density.

I. INTRODUCTION

In ad hoc and sensor networks, it is critically important to save energy.¹ Many papers have recently appeared which propose MAC, routing, and topology maintenance schemes which try to save energy based on aggressive power-off strategies. In fact, it has been recognized that the only way a node can save substantial energy is to power off the radio, since transmitting, receiving and listening to an idle channel are functions which require roughly the same amount of power. As a consequence of this key observation, MAC and routing strategies need to be revisited since, for example, CSMA-based access schemes need all nodes to continuously listen to the channel while, on the other hand, nodes which power off their radio may end up not being reachable and/or aware of activity in the network. The main problem in this scenario is therefore that of combining protocols which minimize the amount of time the radio is on with effective strategies for MAC and routing.

An example of how topology can be maintained in the presence of sleeping nodes is provided by SPAN in [1], where the authors propose that in a dense network several disjoint sets of nodes be identified, each able to guarantee connectivity and bandwidth to all nodes. As long as one of these sets is active at any given time, the network is connected; on the other hand, since when one set is active the others can sleep, the percentage of time a node must be active is drastically reduced. STEM [2] provides a way to establish communications in the presence of sleeping nodes. Each sleeping node wakes up periodically to listen. If a node wants to establish communications, it starts sending out beacons polling a specific user. Within a bounded time, the polled node will wake up and receive the poll, after which the two nodes are able to communicate. An interesting feature of STEM is that a dual radio setup is envisioned, with separate frequencies used for wakeup and actual data transmission.

This work has been partially supported by the European Commission under contract IST-2001-34734 “EYES.”

¹While in this paper we will mostly refer to sensor networks, many of the ideas presented can also be applied to ad hoc networks.

GAF [3] is similar to SPAN in a sense, since it envisions the use of only a fraction of nodes at any given time. The specific approach of GAF is to divide the area in square regions, called grids, in such a way that any two nodes in neighboring grids are within range of each other. With this provision, grids can be treated as equivalent (or virtual) nodes, in the sense that all nodes in the same grid can be interchangeably used for routing purposes. The price to pay for this guarantee is that the hop length is significantly smaller than the radio range (by a factor of $\sqrt{5}$ [3]). This may result in inefficiency in terms of latency and energy consumption (more hops than possibly needed).

A common characteristic of the above schemes is that, at the MAC layer and often also at the routing layer, when a node decides to transmit a packet (as the originator or a relay) it specifies the MAC address of the neighbor to which the packet is being sent. Knowledge of the network topology (though in many cases only local in extent) is required since a node needs to know its neighbors and possibly some more information related to the availability of routes to the intended destination. This topological information can be acquired at the price of some signaling traffic, and becomes more and more difficult to maintain in the presence of network dynamics (e.g., nodes which move or turn off without coordination).

In this document, we describe a transmission scheme based on geographical routing where packets are relayed on a best-effort basis, i.e., the actual node which acts as a relay is not known a priori by the sender, but rather is decided after the transmission has taken place. This idea leverages on the fact that in the wireless environment broadcast is free (from the sender’s point of view) and that in the presence of randomly changing topologies a node may not be aware of which of its current neighbors is in the best position to act as a relay. In a sense, this is like doing contention at the receiver’s end, which is untraditional because in classic schemes it is the transmitter which contends for the channel. Here, since the intended recipient is not specified, multiple nodes may be able to receive the packet, and a *receiver contention scheme* is therefore needed to avoid duplication.

II. COLLISION AVOIDANCE SCHEME

We consider a scheme which uses carrier sense before transmission, which partially avoids collisions but gives no guarantee against the hidden terminal problem. Notice that the fact that nodes are not always on makes traditional RTS/CTS-based collision avoidance mechanisms ineffective since a node may wake up after the CTS was issued. This could be solved

by requiring a long idle channel time to be detected before a transmission can start (essentially enough for the whole packet exchange to complete, which is of course very wasteful) or by synchronizing all nodes as in [4], which requires additional signaling and complexity.

The solution we adopt here is the use of busy tones [5], [6]. It was observed in [2] that there exist sensor nodes equipped with two radios. In [2] the availability of separate channels for the data traffic and the wakeup signaling is useful to facilitate protocol operation, in particular to avoid that prolonged beacon periods interfere with data traffic. In our case, we use the second radio to let the receiving node issue a busy tone, which is a way to effectively prevent collisions at the receiver. Notice that we can trade off energy and latency as in [2] by using a pulsed busy tone with some duty cycle, with the requirement that the sensing time be increased in order to avoid that silent intervals of the busy tone are interpreted as idle channel.

The protocol will then work as follows. When a node has a packet to send, it listens to both frequencies. If either is active, the node backs off. If both are inactive, the node transmits. The collision avoidance feature of this scheme is based on the RTS/CTS message exchange. However, unlike the traditional RTS message which is addressed to a specific node, in this case any node within range can respond to it, with nodes closer to the destination doing so with higher priority. Therefore, the CTS message is also subject to contention, since multiple nodes may decide to respond to the same RTS at the same time. A more detailed explanation of the protocol, including the contention mechanism, is given next.

A. Detailed description

We now describe in detail the protocol operation from the transmitter and the receiver side. The current description is for a specific solution, and many variants are possible which improve the performance while being more complicated to explain. In this section we choose a simple version to highlight the main points.

1) *Transmitter*: When a sleeping node has a packet to send, it transitions to the active state and monitors both frequencies for τ seconds. If either frequency is busy, the node backs off and reschedules an attempt at a later time. If on the other hand both frequencies are sensed idle during this entire interval, the node transmits a broadcast RTS message, which contains the location of the intended destination as well as that of the sender. After sending the RTS, the transmitting node listens for CTS messages from potential relays. In each of the CTS slots following the end of the RTS message, the transmitting node acts as follows: i) if only one CTS message is received, it starts transmission of the data packet, whose initial part acts as a CTS confirmation for the node which issued the CTS; ii) if it receives no CTSs, it will send a CONTINUE message and listen again for CTSs, timing out after N_p empty CTS slots (which forces the node to abort the handshake and to reschedule it at a later time); iii) if it hears a signal but is unable to detect a meaningful message, it will assume that a collision took place, and will send a COLLISION message which will trigger the start of a collision resolution algorithm (to be described later) and will listen again for CTSs.

After packet transmission, an immediate ACK is expected. If it is correctly received, it completes the transaction and the node can go back to sleep. If the transmitter does not get an ACK within a given time, it times out and declares

the transaction failed. It will then reschedule the same packet for future transmission. After N_{MaxAtt} failed attempts for the same packet, the transmitter will give it up and generate an error message for the higher layers.

Notice that with the above rules the protocol does not lead to transmitter deadlock, as it will never wait indefinitely for CTSs or ACKs. Only in the case of completed transaction will the transmitter consider the packet as successfully passed to the next hop. A remaining problem with this scheme is that packet duplication may occur. In fact, if the final ACK is lost the relay node is now in charge of packet delivery whereas the transmitter will not be aware of this fact and will retry the packet transmission. This ambiguity does not compromise the correctness of the scheme and can be solved by intermediate nodes when an additional copy of the same packet is received and discarded. This requires that nodes keep memory of recent transmissions. If this is not possible or desirable, as well as in the case in which the duplicate packet goes through a different set of nodes, packet duplication will be detected at the destination, which leads to some inefficiency which on the other hand is mitigated by the fact that losing an ACK when the packet was successful is a low probability event and the overall performance impact may be expected to be limited.

2) *Receiver*: Each node will (more or less) periodically wake up and put itself in the listening mode. If nothing happens throughout the listening time, whose duration may be fixed or random, the node goes back to sleep. On the other hand, if the node detects the start of a transmission, it goes into the receiving state.

Upon detecting the start of a message, a listening node starts receiving. At the same time, it activates the busy tone on the signaling frequency for a duration T_{RTS} . If no valid RTS is received, the node goes back to the listening state, where it stays for the originally scheduled duration. On the other hand, if a valid RTS is received, the node reads the information in it and determines its own priority as a relay. This priority is based on the relative location of the node itself compared to the distance between the transmitter and the intended final destination. Specifically, assume the following: the portion of the coverage area of the transmitter which is closer to the intended destination than the transmitter itself is divided in N_p regions $\mathcal{A}_1, \dots, \mathcal{A}_{N_p}$ such that all points in \mathcal{A}_i are closer to the destination than all points in \mathcal{A}_j for $j > i, i = 1, \dots, N_p - 1$. (Possible choices of these regions may be to take all with the same area or to quantize the advancement in N_p equal levels.)

In the first CTS slot after the RTS, all nodes in \mathcal{A}_1 will send a CTS message, while all others will be silent. All nodes will then listen for the message from the transmitter in the latter part of the CTS slot. If a packet start is heard (which contains the identification of the node which sent the CTS), only the designated node will continue to receive, whereas all others will go back to sleep. Notice that going back to the listening state is not a good strategy since these nodes are in the coverage area of the transmitter and therefore will be unable to serve as relays for any other nodes. In the interest of energy saving, the best thing to do is to go back to sleep regardless of any previous schedule (if the listening interval is significantly longer than a complete transaction, nodes could just interrupt their listening and resume it at the end of the transmission).

If in the second part of the first CTS slot a CONTINUE message is heard, it means that there are no nodes in \mathcal{A}_1 , and all nodes in \mathcal{A}_2 will contend in the second CTS slot. If an

ABORT message is received, the transmitter has reached the maximum allowed number of CTS slots and the transaction is aborted.

If on the other hand a COLLISION message is received, this means that more than one CTS was generated in the CTS slot. All nodes who did not transmit will drop out (they recognize that higher priority nodes are present) while those involved in the collision will start the collision resolution algorithm. Each colliding node will decide with probability 0.5 whether to continue or drop. Who decides to continue will send a CTS in the next slot. Three events are possible: i) only one node sends, transmitter starts packet transmission and all others drop; ii) more than one CTSs, transmitter sends a COLLISION message, those who did not send drop out, those involved in the collision decide whether to continue as before; iii) no CTS is heard, a CONTINUE message is sent by the transmitter, and all nodes who did not select the current slot decide again independently whether to continue as before. This procedure will terminate in few slots with high probability. In order to force it to be limited, the transmitter can send an ABORT message if the collision is not resolved within $N_{MaxColl}$ CTS slots. Finally, any node which receives a message it does not understand will drop out.

Nodes which heard the RTS correctly will follow the sequence of steps above, and they are guaranteed to either become the relay node or to drop out at some point. The event that two nodes think they are the designated relay can be completely avoided if the start of the packet contains the full relay node's address, and is very unlikely in a simplified scheme where at the start of the packet a random number (included in the CTS as temporary short address) is transmitted.

III. ANALYSIS

We now give an approximate analysis, in order to gain some understanding of the basic mechanisms and sources of energy consumption of our scheme, as well as to compare it to STEM.

Note that a node in the network, while mostly sleeping, wakes up for two reasons, namely if it has a packet to transmit and if it is time to listen according to the wakeup scheme. Note that in the latter case there are three possibilities, i.e.: i) nothing is received and the node goes back to sleep after a specified amount of time; ii) activity is detected but the node is not selected as a relay; and iii) the node acts as a relay. Clearly these three possibilities correspond to different times and energy consumptions.

Consider a long time interval of duration t . The total average energy consumed during this time can be expressed as follows:

$$E_{tot} = N_T E_T + N_\ell E_\ell + T_s P_s \quad (1)$$

where N_T and N_ℓ are the average number of times (during t) the node transmits a packet and wakes up to listen, respectively, while E_T, E_ℓ is the average amount of energy consumed following either event.

A. Packet transmission

In case of a packet transmission, the transmitting node sends an RTS message, of duration T_{RTS} , and listens for CTSs until it hears some signal. If N_p CTS slots go by without any CTS heard, the sender times out and retries. Let N be the average number of nodes in the coverage area, $M = dN$ the average number of such nodes *listening* (d is the duty cycle of the

listening activity of each node), and ξ the fraction of those nodes which are considered as relays.² The probability that there are no nodes who can answer the RTS is then $e^{-\xi M}$. Each cycle where no nodes respond then involves, besides the RTS, N_p empty CTS slots, i.e., the sender transmits one RTS, listens (without receiving anything) for $N_p T_{CTS}$ and transmits (CONTINUE messages) for $N_p T_{CTSr}$. On average, there are $(e^{\xi M} - 1)^{-1}$ such cycles, followed by a successful handshake, which involves one RTS, x CTSs, and $(x - 1)$ CTS replies, since the reply to the last (successful) CTS is the start of the data packet itself. The average number of CTS slots in a successful handshake, x , can be computed as follows. Let $\lambda_0 = \xi M / N_p$. A successful handshake starts with i empty CTS slots with probability $e^{-i\lambda_0}$ ($0 \leq i < N_p$), followed by σ_k slots with probability $e^{-\lambda_0} \lambda_0^k / k!$, where σ_k is the number of slots needed to resolve a "collision" in which $k \geq 1$ nodes are involved. Note that, with the binary splitting strategy used to resolve collisions, the average number of slots $s_k = E[\sigma_k]$, obeys the following recursive relationship [7]

$$s_k = \begin{cases} 1 & k = 1 \\ \frac{1 + 2^{-k} \sum_{i=1}^{k-1} \binom{k}{i} s_i}{1 - 2^{-k+1}} & k > 1 \end{cases} \quad (2)$$

Then, we have

$$\begin{aligned} x &= \sum_{i=0}^{N_p-1} i e^{-i\lambda_0} + \frac{\sum_{k=1}^{\infty} \frac{e^{-\lambda_0} \lambda_0^k}{k!} s_k}{1 - e^{-\lambda_0}} \\ &= \frac{e^{-\lambda_0}}{1 - e^{-\lambda_0}} - \frac{N_p e^{-N_p \lambda_0}}{1 - e^{-N_p \lambda_0}} + \frac{\sum_{k=1}^{\infty} \frac{e^{-\lambda_0} \lambda_0^k}{k!} s_k}{1 - e^{-\lambda_0}} \end{aligned} \quad (3)$$

Finally, the sender transmits the packet for T_D and receives the ACK for T_{ACK} . Note that these messages are transmitted/received on two different radios but, if we ignore switching times, there is exactly one radio on at any given time during these cycles. The total time the transmitting node radio is on is then given by

$$\begin{aligned} t_T &= (e^{\xi M} - 1)^{-1} (T_{RTS} + N_p (T_{CTS} + T_{CTSr})) \\ &\quad + T_{RTS} + x T_{CTS} + (x - 1) T_{CTSr} + T_D + T_{ACK} \end{aligned} \quad (4)$$

and if we assume that the power spent in each of these functions (transmit, receive and listen) is the same P for all, the total energy spent every time a node wants to transmit a packet is

$$E_T = t_T P \quad (5)$$

and the contribution of the energy associated to packet transmission to the total average power consumption E_{tot}/t is

$$\frac{N_T E_T}{t} = \lambda P t_T \quad (6)$$

where λ is the packet arrival rate at each node.

B. Listening/Receiving

Each node wakes up periodically with duty cycle d , and stays on for a time T_L . The average rate of packet arrivals in its coverage area is λN , and the probability that no activity is detected is $p_0 = e^{-\lambda N T_L}$. In this case, the node just spends the amount of time T_L listening and goes back to sleep. On the other hand, with probability $1 - p_0$, someone in the coverage

²Note that nodes who are placed opposite with respect to the destination should not be used as a general rule.

area will start an RTS. Before being able to know whether it can be considered as a relay, the node must receive this RTS. Since the arrival time of this RTS is uniformly distributed within the listening interval, the actions involved are listening to the channel for $T_L/2$ on average, and receiving for T_{RTS} . Note that as soon as the node detects channel activity, it turns on its busy tone, so that a transmit activity for T_{RTS} must be accounted for as well.

Given that an RTS is started, with probability $1 - \xi$ the node will not be in the portion of the coverage area facing the destination, and will drop out immediately after receiving the RTS. In this case, there is no additional activity involved. On the other hand, with probability ξ the node will participate in the contention, along with other nodes whose number is a Poisson r.v. with mean ξM . Since all participating nodes have the same probability of being the winner, the probability that the node wins the contention is found as

$$\sum_{k=0}^{\infty} \frac{1}{k+1} \frac{e^{-\xi M} (\xi M)^k}{k!} = \frac{1 - e^{-\xi M}}{\xi M} \quad (7)$$

In this case, i.e., the node wins the contention, it is involved at most in sending x CTSs, receiving $(x - 1)$ CTS replies, receiving the data packet and finally sending the ACK. When the node is receiving, i.e., for a time equal to $(x - 1)T_{CTSr} + T_D$, the busy tone is on.

If on the other hand the node participates in the contention but loses it (with conditional probability $(\xi M - 1 + e^{-\xi M})/\xi M$), the activity involved is upper bounded by receiving $(x - 1)$ CTS replies and transmitting continuously (CTSs or busy tone) for $(x - 1)(T_{CTS} + T_{CTSr})$. Note in fact that nodes losing the contention do not necessarily participate until the end, and with certainty do not transmit in the very last CTS slot (in which somebody else is successful).

In summary, the total average active time of the radio (counting twice the times when both radios are on) can be found as

$$\begin{aligned} t_\ell &= p_0 T_L + (1 - p_0) \left[\frac{T_L}{2} + 2T_{RTS} \right. \\ &+ \frac{1 - e^{-\xi M}}{M} (xT_{CTS} + 2(x - 1)T_{CTSr} + 2T_D + T_{ACK}) \\ &+ \left. \frac{\xi M - (1 - e^{-\xi M})}{M} (x - 1)(T_{CTS} + 2T_{CTSr}) \right] \\ &= T_L + (1 - p_0) [\xi(x - 1)(T_{CTS} + 2T_{CTSr}) \\ &+ 2T_{RTS} - \frac{T_L}{2} + \frac{1 - e^{-\xi M}}{M} (T_{CTS} + 2T_D + T_{ACK})]. \quad (8) \end{aligned}$$

For reasonable scenarios, the probability that upon wakeup the node ends up being involved in a data exchange will have to be small (the whole idea being to avoid heavy load of the nodes). In order to have network stability we must have $\lambda N T_{DATAex} < 1$, i.e., the average number of users in transmission state per coverage area must be less than unity (T_{DATAex} is the total time for a data transfer from RTS to ACK). If we assume that $T_L \ll T_{DATAex}$, we have that $\lambda N T_L \ll 1$. In this case, we have $p_0 \simeq 1$ and $1 - p_0 = 1 - e^{-\lambda N T_L} \simeq \lambda N T_L$.

If once again we assume that an active radio consumes a power P regardless of its being in transmit, receive or listen mode, the total average contribution to the total average power consumption E_{tot}/t can be found as $E_\ell N_\ell/t$. For an unloaded network, we would have $N_\ell/t = d/T_L$, while in general it is

true that $N_\ell/t \leq d/T_L$, and the bound is tight for low traffic. In this case, we can write

$$\begin{aligned} \frac{E_\ell N_\ell}{t} &\simeq \frac{E_\ell d}{T_L} = \frac{d P t_\ell}{T_L} \\ &= dP + \lambda P \left[-\frac{M T_L}{2} + \xi M (x - 1)(T_{CTS} + 2T_{CTSr}) \right. \\ &\quad \left. + 2M T_{RTS} + (1 - e^{-\xi M})(T_{CTS} + 2T_D + T_{ACK}) \right] \quad (9) \end{aligned}$$

where we used the fact that

$$\frac{dP(1 - p_0)}{T_L} \simeq \frac{dP \lambda N T_L}{T_L} = \lambda P M \quad (10)$$

C. Sleeping

The total amount of energy consumed while sleeping is given by $T_s P_s$, where T_s is the total amount of time the two radios are off. Notice that since in the above analysis we never accounted for sleeping times in between active periods of the radios, T_s must include those times as well. In any event, we can affirm that the contribution of sleeping time to the overall average power consumption is

$$\frac{T_s P_s}{t} \leq P_s \quad (11)$$

where P_s is the power consumed when both radios are off. In view of the fact that in the envisioned scenarios the radios must be sleeping most of the time, we have $t - T_s \ll t$, and therefore the above bound is tight and can be used as a reasonable approximation.

D. Total average energy consumption

We can find the total normalized average energy consumption to be

$$\psi_0 = \frac{E_{tot}}{P t} = \frac{1}{P} \left(\frac{N_T E_T}{t} + \frac{E_\ell N_\ell}{t} + \frac{T_s P_s}{t} \right) \quad (12)$$

where the expressions for the three terms are given above. ψ_0 is the total energy consumed in time t , divided by the energy which would be consumed by a radio which is always on (transmitting, receiving or monitoring the channel).

E. Latency

We define here latency as the time it takes from when a node starts the packet transmission handshake to when the transmission of the actual data packet starts. In our scheme, this corresponds to t_T minus the time for data and ACK, i.e.,

$$\begin{aligned} \bar{T}_S &= (e^{\xi M} - 1)^{-1} (T_{RTS} + N_p (T_{CTS} + T_{CTSr})) \\ &\quad + T_{RTS} + x T_{CTS} + (x - 1) T_{CTSr} \quad (13) \end{aligned}$$

F. Analysis of STEM

A similar analysis can be carried out for the STEM scheme. We consider here STEM-B [2].

Energy consumption The average energy consumption can still be divided into three terms. In a packet transmission, the sender sends beacons until the intended recipient wakes up and receives one. At that point, packet exchange takes place via 802.11-like MAC. Nodes wake up every T seconds for T_L . The average time the beacon needs to be sent is then given by $(T - T_B)/2 + B_1$ [2], and if we assume that $T_L = T_B + B_1$

we obtain $(T - T_L)/2 + 1.5B_1$, where $T = T_L/d$. The total average amount of time the node is powered on is therefore given by

$$t_T = \frac{T_L}{2d} - \frac{T_L}{2} + 1.5B_1 + T_{CTS} + T_D + T_{ACK} \quad (14)$$

where we assumed that the beacon acts as RTS. The contribution to the average energy consumption due to packet transmission is then given by $\lambda P t_T$.

After waking up, a node will be addressed with probability $1 - p_0$, where p_0 is the probability that no activity is detected. Note that in this case nodes are explicitly addressed, and therefore the rate at which messages for a specific node are generated is lower than before (where on the other hand all nodes in the coverage area would receive the RTS). However, since a beacon is for a specific node, the interval of time during which a new message can be generated is now T rather than T_L . The message arrival rate is then given by $\frac{\lambda N T}{N} = \frac{\lambda T_L}{d}$. Notice that this is the average fraction of listening periods in which a node gets a message, and as before in the envisioned scenario we expect this number to be small.

After waking up, a node will listen for T_L and go back to sleep with probability $p_0 = e^{-\lambda T_L/d}$. With probability $1 - p_0 \simeq \lambda T_L/d$, the node will be involved in receiving a message. In this case, note that in STEM the listen time is $T_L = T_B + B_1$, where T_B is the period with which beacons are sent and B_1 is the length of a beacon. Since the beacon start time is uniformly distributed within T_B , the average time to receive a beacon is $T_B/2 + B_1 = (T_L + B_1)/2$. After receiving a beacon, the node's radio is involved in sending a CTS, receiving a data packet and sending an ACK. The total activity time for listening/receiving is therefore

$$\begin{aligned} t_\ell &= p_0 T_L + (1 - p_0) \left[\frac{T_L + B_1}{2} + T_{CTS} + T_D + T_{ACK} \right] \\ &\simeq T_L + \frac{\lambda T_L}{d} \left[\frac{B_1 - T_L}{2} + T_{CTS} + T_D + T_{ACK} \right] \quad (15) \end{aligned}$$

Finally, as before, we approximate the contribution of sleep mode to the overall average power as P_s .

The total average energy consumption in STEM can therefore be computed as

$$\psi_s = \frac{E_{tot}}{P t} \leq \lambda t_T + \frac{dt_\ell}{T_L} + \frac{P_s}{P} \quad (16)$$

Latency. If we define latency as the time from when a beacon is initiated to the time an ACK for it is successfully received (and therefore data exchange can start), we have as in [2] (we assume here $\varepsilon = 0$, which corresponds to minimum listening time T_L)

$$\bar{T}_S = B_{1+2} + \frac{T - T_B}{2} = \frac{T - T_L}{2} + 1.5B_1 + B_2 \quad (17)$$

IV. PERFORMANCE COMPARISON

In this section, we give some numerical results for the schemes considered, and provide a comparison between them.

First of all, notice that there are three types of parameters in the above formulas:

- **fixed parameters**, i.e., parameters which are expected to be decided once for all and will be considered as constant: in particular, we choose $N_p = 4, \xi = 0.4, T_{SIG}/T_D = 0.1$ (we assume here for simplicity that all signaling packets are of the same length T_{SIG});

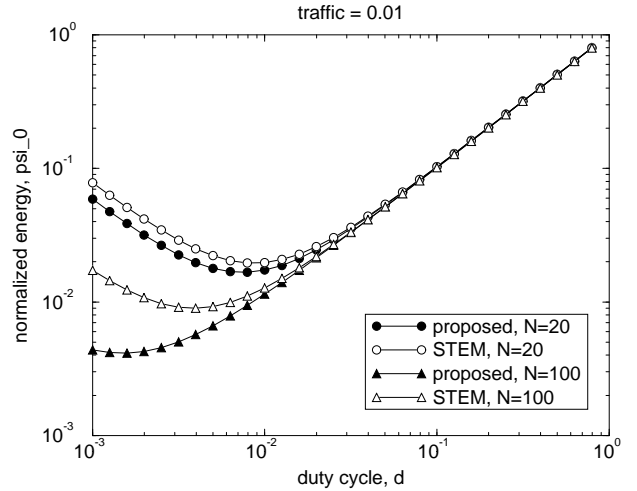


Fig. 1. Average normalized energy consumption, ψ_0 , vs. duty cycle, d . Proposed scheme and STEM compared. $N = 20, 100$, network load 0.01.

- **external parameters**, i.e., parameters which are common to all schemes and provide the scenario in which those schemes are compared: in particular, the node density and the network traffic; here, we use the average number of nodes per coverage area, N , as a measure of the network density, and the average normalized traffic per coverage area, $\lambda N T_D$ as a measure of the network load;
- **parameters of the specific schemes**, i.e., parameters which play different roles in different schemes and can be chosen differently according to the scheme selected; for example, the listening time or the duty cycle may not be the same in the proposed scheme and in STEM. In reality, these parameters would be subject of protocol optimization, and a fair comparison should take into account that they can be independently selected.

As to the parameter optimization, note the following. In STEM, the minimum listening time is $T_B + B_1 = 3T_{SIG}$. Since it is obvious that the best choice is to select T_L as small as possible, we set $T_L = 3T_{SIG}$ here. The remaining independent parameter is the duty cycle. In our scheme, if we upper bound the energy consumption by neglecting the negative term $-MT_L/2$ in (9), the listening time no longer appears explicitly in the expressions, and can therefore be ignored. In this case also, the duty cycle is the only remaining independent parameter.

Results are shown in Figures 1 through 4, in which the performance of our proposed scheme is compared with that of STEM. Figures 1 and 2 show the normalized energy performance, ψ_0 , vs. the duty cycle, d . In both schemes, for large duty cycle, the energy consumption is dominated by the listening activity, as expected. As the duty cycle is decreased, other sources of energy consumption are important. In particular, the fact that the transmitter must spend energy to “find a neighbor” (either via the beacon as in STEM or by repeated attempts in our scheme) becomes dominant, and more so as the network load is higher. Note that in all cases our scheme outperforms STEM, with more significant gains when the node density is large.

It should be noted that the choice of the duty cycle does not have to be the same in the two schemes, as they may

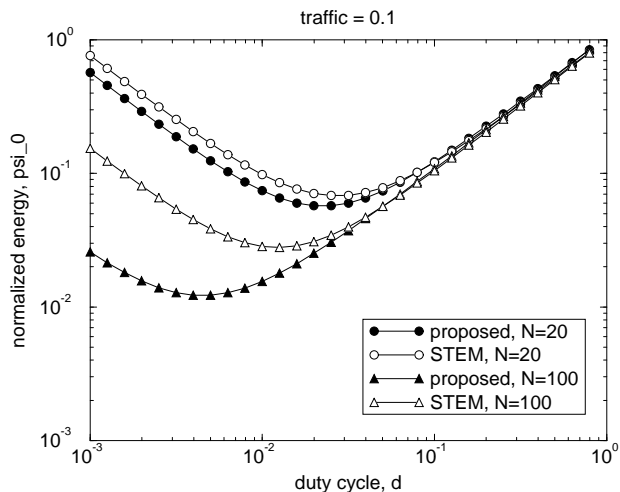


Fig. 2. Average normalized energy consumption, ψ_0 , vs. duty cycle, d . Proposed scheme and STEM compared. $N = 20, 100$, network load 0.1.

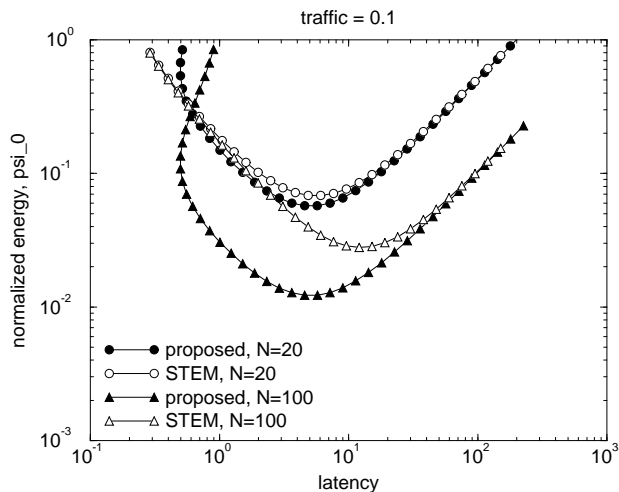


Fig. 4. Average normalized energy consumption, ψ_0 , vs. latency. Proposed scheme and STEM compared. $N = 20, 100$, network load 0.1.

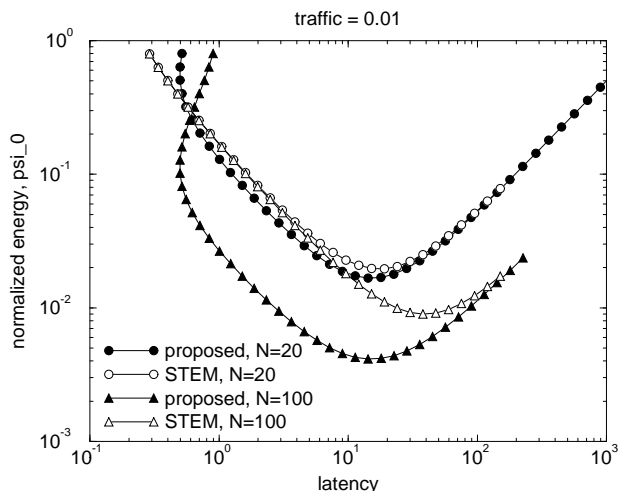


Fig. 3. Average normalized energy consumption, ψ_0 , vs. latency. Proposed scheme and STEM compared. $N = 20, 100$, network load 0.01.

be independently optimized (notice that the minimum energy occurs for different values of d for the two schemes). However, it is clear from the figures that for dense networks the minimum energy consumption achievable by our scheme is significantly smaller than that in STEM. To shed some more light on this issue, we plot the trade-off between energy consumption and latency in Figures 3 and 4. The curves are generated by spanning the range of values of the duty cycle (curves are traveled right to left by increasing the duty cycle). For very low duty cycle and not enough nodes, both schemes perform badly since the latency associated to long sleep times is unacceptable, and the persistence in looking for a relay results in degraded energy performance as well. On the other hand, in the interesting region in which there is a real trade-off between energy and latency, it is seen how our scheme performs better than STEM, more so for denser networks. More specifically, although for relatively sparse networks the improvement is marginal, for networks with $N = 100$ nodes

per coverage area we can gain almost an order of magnitude in latency for comparable energy or in energy for comparable latency. The proposed scheme therefore appears as a promising alternative for low-power networking.

V. CONCLUSIONS

In this paper, we described a novel forwarding technique based on geographical location of the nodes involved and random selection of the relaying node via contention among receivers. A collision avoidance scheme based on this idea was described in detail, and an approximate analysis was provided. The proposed scheme was compared with STEM, and was shown to perform significantly better for sufficient node density.

Items for future research include a more detailed analysis, comparison with other proposed schemes, and optimization of the parameters.

REFERENCES

- [1] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, Robert Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in Ad Hoc wireless networks," ACM/IEEE MobiCom2001, July 2001.
- [2] Curt Schurgers, Vlasios Tsitsis, Saurabh Ganeriwal, Mani Srivastava, "Optimizing sensor networks in the energy-latency-density design space," IEEE Trans. on Mobile Computing, vol. 1, Jan-Mar 2002, p. 70-80.
- [3] Ya Xu, John Heidemann, Deborah Estrin, "Geography-informed energy conservation for Ad Hoc routing," ACM/IEEE MobiCom2001, July 2001.
- [4] Wei Ye, J. Heidemann, D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," Proceedings of INFOCOM 2002, New York NY, June 2002, p. 1567-1576.
- [5] F.A. Tobagi, L. Kleinrock, "Packet switching in radio channels. II. The hidden terminal problem in carrier sense multiple-access and the busy-tone solution," IEEE Trans. Commun., vol. 23, pp. 1417-33, Dec. 1975.
- [6] Z.J. Haas, Jing Deng, "Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks," IEEE Trans. Commun., vol. 50, pp. 975-85, June 2002.
- [7] M. Zorzi, R.R. Rao, "Integrated MAC and routing by geographic random forwarding for ad hoc and sensor networks," UCSD Technical Report, September 2002.