Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance

Michele Zorzi, *Senior Member, IEEE,* and **Ramesh R. Rao,** *Senior Member, IEEE*
to appear in the *IEEE Transactions on Mobile Computing*, vol. 2, n. 4, Oct.-Dec. 2003

***Abstract*— In this paper, we study a novel forwarding technique based on geographical location of the nodes involved and random selection of the relaying node via contention among receivers. We provide a detailed description of a MAC scheme based on these concepts and on collision avoidance, and report on its energy and latency performance. A simplified analysis is given first, some relevant tradeoffs are highlighted and parameter optimization is pursued. Further, a semi-Markov model is developed which provides a more accurate performance evaluation. Simulation results supporting the validity of our analytical approach are also provided.**

I. INTRODUCTION

Energy conservation is one of the key technical challeges in sensor networks and ad hoc networks. It is necessary to devise communications and networking schemes which make judicious use of the limited energy resources without compromising the network connectivity and the ability to deliver data to the intended destination. In addition, sensor nodes are often subject to further constraints in terms of CPU power, memory space, etc., which call for simple algorithms and schemes whose memory needs are modest.

One of the main mechanisms to save energy is the use of sleep modes at the MAC layer, by which nodes are put to sleep as often as possible. This must be done in such a way that connectivity is preserved, since if too many nodes are sleeping at the same time, the network may end up being disconnected. In the recent literature, several schemes have been proposed which address this problem. For example, SPAN [1] tries to coordinate the sleeping activity of the nodes so that a connecting backbone is always present. GAF [2] identifies groups of nodes which are equivalent from a routing point of view, i.e., in each group it is sufficient that a single node is awake at any given time. STEM [3], on the other hand, provides a means to communicate with a node currently asleep, by implementing a *rendez-vous* mechanism based on beacon transmissions. As to the MAC itself, most papers in the literature assume either TDMA-based schemes [4], or multi-channel setups in which parallel transmissions can be performed without interference [5], [6], or variants of classic contention-based schemes, usually based on RTS/CTS handshake in order to mitigate the hidden terminal problem [7], [8].

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 addition, the proposed schemes do have some performance problems, e.g., the radio range is significantly underutilized in GAF (which means that more hops are needed to cover a given distance) and potentially large delays may be introduced in STEM (in order to wait for a given node to wake up).

We propose here an alternative solution, called Geographic Random Forwarding (GeRaF, pronounced as “giraffe”), which is based on the assumption that sensor nodes have a means to determine their location, and that the positions of the final destination and of the transmitting node are explicitly included in each message. In this scheme, a node which hears a message is able (based on its position towards the final destination) to assess its own priority in acting as a relay for that message. All nodes who received a message may volunteer to act as relays, and do so according to their own priority. This mechanism tries to choose the best positioned nodes as relays. In addition, since the selection of the relays is done *a posteriori*, no topological knowledge nor routing tables are needed at each node, but the position information is enough. Geographic routing is used here to enable nodes to be put to sleep and waken up without coordination, and to integrate routing, MAC and topology management into a single layer. This basic idea is described in some more detail in a companion paper [9], where the multihop performance of the scheme is also studied.

In this paper, a collision avoidance protocol based on this idea is described in detail. We provide a detailed analysis of the energy and latency performance of this protocol. First, a simplified analysis is given, which leads to closed-form expressions and, via further approximations, to simple parameter optimization rules. The energy-latency tradeoff is explored, and some performance results are presented, showing that the proposed solution is a promising alternative for low-power networking. As a first step, we focus on the fundamental behavior of GeRaF, and we consider STEM [3] as the most

This work has been partially supported by the European Commission under the EYES project (contract IST-2001-34734) and by the Italian government under the FIRB VICOM project. Parts of this work have been presented at IEEE WCNC 2003 and IEEE VTC 2003 Spring.

Corresponding author: CNIT and Dipartimento di Ingegneria, 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; ph.: +1 858 822 4572 – fax: +1 858 822 5197 – e-mail: rrao@ucsd.edu

appropriate scheme to which we compare, since it is the one which is most closely related to our approach. More extensive comparisons with other energy-conserving schemes, including [7], [10], are the subject of future work.

Further, a more complete approach is followed in which a semi-Markov model for protocol operation is developed and solved. This second approach is more accurate, especially in networks which are not very dense. Comparison between the results of the two approaches shows that the simplified analysis can accurately predict the behavior in a large fraction of the range and, more importantly, it accurately predicts the optimal value of the duty cycle of the sleeping behavior of the nodes. Finally, some preliminary simulation results are shown to confirm the validity of our analysis.

II. Collision avoidance MAC 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 [7], which requires additional signaling and complexity.

The solution we adopt here is the use of busy tones [11], [12]. It was observed in [3] that there exist sensor nodes equipped with two radios. In [3] 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, no prolonged beacon periods are present, and therefore we could use the second radio to let the receiving node issue a busy tone, which is a way to effectively prevent collisions at the receiver. More precisely, on the first frequency (“data” frequency) all message exchanges occur, whereas the second frequency (“busy tone” frequency) is used for busy tones only. Notice that we could trade off energy and latency as in [3] 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. In addition, these pulsed busy tone messages could act as partial ACKs thereby allowing recovery of smaller pieces of the message if individual CRCs are available (consideration of these variations as well as their performance implications are left for future study). Note that this is possible since the transmitter also has two radios which can be used independently (in particular, one to transmit the data and the other to receive the busy tone/partial ACK messages).

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. The issuance of CTS messages in response to an RTS is done in such a way as to give priority to nodes which provide a larger advancement towards the final destination, as detailed below.

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 may 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 enters 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 its own. After sending the RTS, the transmitting node listens in the subsequent slots 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 CTS 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. Notice that if the receiver is an intermediate node towards the destination, in a scheme in which a packet exchange is immediately initiated the RTS message itself could serve as the ACK. Here, on the other hand, we assume that an explicit ACK is used. 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. While listening for CTSs and for the ACK, the node transmits the busy tone to prevent interference from hidden terminals.

Notice that with the above rules the protocol does not lead to transmitter deadlock, as the sender 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 transmission of the same packet. 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 multiple copies of the same packet go through disjoint sets of nodes, packet duplication will be detected at the destination, which leads to some inefficiency that 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. Note that the randomness of the events involved makes the sleep process not exactly periodic. The sleep time will be considered as a constant in the following, for convenience of explanation and of analysis. In reality, more sophisticated schemes could be envisioned, in which sleep times could be random or could depend on the battery status (i.e., nodes with less charge tend to sleep longer). These variants are left for future study. [1]

Upon detecting the start of a message, a listening node starts receiving. At the same time, it activates the busy tone on the busy tone 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 $\mathcal{A}_1, \ldots, \mathcal{A}_{N_p}$ such that all points in $\mathcal{A}_i$ are closer to the destination than all points in $\mathcal{A}_j$ for $j > i, i = 1, \ldots, N_p - 1$. (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 node. 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 or not to continue. 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 are sent in the same slot, transmitter sends a COLLISION message, those who did not send drop out, those involved in the collision decide whether or not to continue as before, until the collision is resolved; 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, or made 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. In order to avoid the hidden terminal problem, each node involved in the above procedure will keep the busy tone active until it drops out or, if it is the winner, until the whole data packet has been received.

We stress the fact that the above protocol choices (e.g., the details of the collision resolution algorithm) are made just to give an example of how it is possible to provide the related functionality. It is not our goal here to optimize these schemes, but rather to show that our proposal is able to achieve satisfactory performance. Even better performance could be obtained if further optimization is pursued, and this is left as an interesting topic for further work.

III. Approximate analysis

We first develop an approximate analysis of the GeRaF collision avoidance scheme described in the previous section, in order to gain some understanding of the basic mechanisms and sources of energy consumption, as well as to compare it to STEM.

[1] Another optimization, not considered here, would be to let a node sense both frequency upon wakeup and immediately go back to sleep if either is busy, since in this case it is impossible for it to act as a relay.

We assume that nodes are distributed throughout the network according to a Poisson process in two dimensions, with intensity ρ nodes per unit area. A node in the network, while mostly sleeping, wakes up for two reasons, namely if it has a packet to transmit or if it is time for it 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 amounts of active time and energy consumption.

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

$$E_{tot} = N_T E_T + N_\ell E_\ell + T_s P_s \tag{1}$$

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. T_s is the total amount of time the node spends in the sleep mode, and P_s is the corresponding power. In the following, the various terms in (1) are evaluated.

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 $N = \rho\pi$ 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. [3] The average number of CTS slots in a successful handshake, x, can be computed as follows. Let $\lambda_0 = \xi M / N_p$ be the average number of available relays in each priority region. [4] A successful handshake starts with i empty CTS slots with probability $e^{-i\lambda_0}$ $(0 \le i < N_p)$, 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 $k \ge 1$ 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

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

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

[3]For simplicity, and with no loss in generality, we ignore here the fact that only a limited number of attempts is allowed, i.e., we take $N_{MaxAtt} = N_{MaxColl} = \infty$.

[4]In order to simplify the notation, here we define the priority regions as having the same area. Extension to the general case is straightforward.

In summary, the joint probability of having exactly $j \ge 0$ empty cycles, followed by a non-empty cycle with i empty slots $(0 \le i < N_p)$ plus one non-empty slot in which $k \ge 1$ CTSs are sent, is given by

$$\left(e^{-N_p \lambda_0}\right)^j e^{-i\lambda_0} \frac{e^{-\lambda_0}\lambda_0^k}{k!}, \quad j \ge 0, 0 \le i < N_p, k \ge 1 \tag{3}$$

Then, for the length of the non-empty cycle we have

$$\begin{aligned} x &= E[i + \sigma_k] \\ &= \sum_{j=0}^{\infty}\sum_{i=0}^{N_p-1}\sum_{k=1}^{\infty}(i + s_k)\left(e^{-N_p\lambda_0}\right)^j e^{-i\lambda_0}\frac{e^{-\lambda_0}\lambda_0^k}{k!} \\ &= \frac{1 - e^{-\lambda_0}}{1 - e^{-N_p\lambda_0}}\sum_{i=0}^{N_p-1} i e^{-i\lambda_0} + \frac{\sum_{k=1}^{\infty}\frac{e^{-\lambda_0}\lambda_0^k}{k!}s_k}{1 - e^{-\lambda_0}} \\ &= \frac{e^{-\lambda_0}}{1 - e^{-\lambda_0}} - \frac{N_p e^{-N_p\lambda_0}}{1 - e^{-N_p\lambda_0}} + \frac{\sum_{k=1}^{\infty}\frac{e^{-\lambda_0}\lambda_0^k}{k!}s_k}{1 - e^{-\lambda_0}} \end{aligned} \tag{4}$$

Finally, the sender transmits the packet for T_D and receives the ACK for T_{ACK}. Note that these messages are all transmitted/received on the data frequency, while transmission on the busy tone frequency is activated during CTS slots and the ACK. For later analytical convenience, we ignore the carrier sense activity, which would involve an additional listening time of τ, which is a very reasonable approximation since $\tau \ll T_D$. The total time the transmitting node is on (counting twice the times during which both radios are active) is then given by

$$\begin{aligned} t_T &= (e^{\xi M} - 1)^{-1}(T_{RTS} + N_p(2T_{CTS} + T_{CTSr})) \\ &+ T_{RTS} + 2xT_{CTS} + (x-1)T_{CTSr} + T_D + 2T_{ACK} \end{aligned} \tag{5}$$

If we assume that the power spent in each of these functions (transmit, receive and listen) is the same P for all, [5] the total energy spent every time a node wants to transmit a packet is $E_T = t_T P$, and the contribution of the energy associated to packet transmission to the total average power consumption E_{tot}/t is

$$\frac{N_T E_T}{t} = \lambda P t_T \tag{6}$$

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 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

[5]This assumption is clearly not critical since one can precisely distinguish among the three functions, thereby assigning the exact power to each if needed. On the other hand, even though possibly not the same, these power levels have been observed to be comparable, and we expect no additional insight from a more accurate definition of the power levels. The assumption therefore appears to be reasonable, and is made here to limit the size of the parameter space.

to the data 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} on the busy tone frequency 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

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

In this case, i.e., the node wins the contention, it is involved at most in sending x CTSs (with x as given in (4)), 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

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

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 $1-p_0 = 1-e^{-\lambda N T_L} \simeq \lambda N T_L$.

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 $N_\ell E_\ell/t$. 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

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

where we used the fact that

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

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 $\frac{T_s P_s}{t} \leq P_s$, where P_s is the overall power consumed when both radios are in sleep mode. 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

$$\psi_0 = \frac{E_{tot}}{Pt} = \frac{1}{P}\left(\frac{N_T E_T}{t} + \frac{N_\ell E_\ell}{t} + \frac{T_s P_s}{t}\right) \tag{11}$$

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 single radio which is always on (transmitting, receiving or monitoring the channel), as is typical in traditional CSMA-based protocols.

In order to simplify the expressions, consider the case in which $T_{RTS} = T_{CTS} = T_{CTSr} = T_{ACK} = T_{SIG}$. In this case, we obtain

$$\begin{aligned} \frac{N_T E_T}{t} \;&=\; \lambda P\left[(e^{\xi M}-1)^{-1}(3N_p+1)T_{SIG} \right. \\ &\left. + (3x+2)T_{SIG} + T_D\right] \end{aligned} \tag{12}$$

$$\begin{aligned} \frac{N_\ell E_\ell}{t} \;&\simeq\; dP + \lambda P\left[-\frac{MT_L}{2} + 2(1-e^{-\xi M})T_D \right. \\ &\left. + \left(3\xi M(x-1) + 2M + 2(1-e^{-\xi M})\right)T_{SIG}\right] \end{aligned} \tag{13}$$

$$\frac{T_s P_s}{t} \simeq P_s \tag{14}$$

and

$$\begin{aligned} \psi_0 \;&\simeq\; d + \frac{P_s}{P} + \lambda\left[\left(3 - 2e^{-\xi M}\right)T_D - \frac{MT_L}{2} \right. \\ &+ \left(3\xi M(x-1) + 2M + 2(1-e^{-\xi M})\right. \\ &\left.\left. + 3x + 2 + (e^{\xi M}-1)^{-1}(3N_p+1)\right)T_{SIG}\right] \end{aligned} \tag{15}$$

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, and can be computed similarly to (5)[6] to obtain

$$\begin{aligned} T_{lat} &= (e^{\xi M}-1)^{-1}(T_{RTS}+N_p(T_{CTS}+T_{CTSr})) \\ &\quad +T_{RTS}+xT_{CTS}+(x-1)T_{CTSr} \\ &= \left((e^{\xi M}-1)^{-1}(1+2N_p)+2x\right)T_{SIG} \end{aligned} \tag{16}$$

F. Analysis of STEM

A similar analysis can be carried out for the STEM scheme. We consider here STEM-B [3]. The basic principle of STEM is the following. Nodes are expected to sleep most of the time, and to periodically wake up to listen. If a node wants to send a packet to one of its neighbors, it starts polling it by sending beacon messages which carry the intended recipient's identity. Since the intended recipient is guaranteed to wake up within a finite amount of time, this polling period ends successfully and results in the communication link between the two nodes involved to be restored. Once this is done, the packet transfer can occur. The details of the beacon message as well as a more complete description of the scheme can be found in [3].

1) 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. [7] Nodes wake up every T seconds for T_L. The average time the beacon needs to be sent is then given by $(T-T_B)/2+B_1$, where T_B is the period with which beacons are sent and B_1 is the length of a beacon [3]. 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

$$\begin{aligned} t_T &= \frac{T_L}{2d}-\frac{T_L}{2}+1.5B_1+T_{CTS}+T_D+T_{ACK} \\ &= \frac{T_L(1-d)}{2d}+T_D+3.5T_{SIG} \end{aligned} \tag{17}$$

where we assumed that the beacon acts as RTS. The contribution to the average power 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

$$\frac{\lambda N T}{N} = \frac{\lambda T_L}{d} \tag{18}$$

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 $T_L = T_B + B_1$. Since the beacon start time is uniformly distributed within T_B, the average time to receive a beacon is $T_B/2+B_1 = (T_L+B_1)/2 = (T_L+T_{SIG})/2$. 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

$$\begin{aligned} t_\ell &= p_0T_L+(1-p_0)\left[\frac{T_L+B_1}{2}+T_{CTS}+T_D+T_{ACK}\right] \\ &= T_L+(1-p_0)\left[-\frac{T_L}{2}+T_D+2.5T_{SIG}\right] \\ &\simeq T_L+\frac{\lambda T_L}{d}\left[-\frac{T_L}{2}+T_D+2.5T_{SIG}\right] \end{aligned} \tag{19}$$

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

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

$$\begin{aligned} \psi_s &= \frac{E_{tot}}{Pt} \le \lambda t_T + \frac{dt_\ell}{T_L}+\frac{P_s}{P} \\ &= \lambda\left(T_D+3.5T_{SIG}+\frac{T_L(1-d)}{2d}\right) \\ &\quad +d\left(1+\frac{\lambda}{d}\left[-\frac{T_L}{2}+T_D+2.5T_{SIG}\right]\right)+\frac{P_s}{P} \\ &= \lambda\left(2T_D+6T_{SIG}+\frac{T_L(1-2d)}{2d}\right)+d+\frac{P_s}{P} \end{aligned} \tag{20}$$

Note that for a given value of λ this expression is independent of N. This leads to the conclusion that STEM as considered here is unable to benefit from an increased node density. A somewhat more fair comparison would be to consider STEM combined with GAF as proposed in [3], so that higher densities of node deployment could be exploited to reduce the energy consumption. This scheme would however have two significant drawbacks, namely a poorer utilization of the coverage radius because of the node organization into grids, and the need for explicit signaling among nodes in order to make the GAF mechanism work. As a first step, in the following results we focus on the original version of STEM. Detailed evaluation of the combination of STEM and GAF, as well as a quantitative assessment of the above phenomena, are out of the scope of the present paper and are left for future study.

2) 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 [3] (we assume here $\varepsilon = 0$, which corresponds to minimum listening time T_L)

$$\begin{aligned} T_{lat} &= B_{1+2}+\frac{T-T_B}{2}=\frac{T-T_L}{2}+1.5B_1+B_2 \\ &= \frac{T_L(1-d)}{2d}+2.5T_{SIG} \end{aligned} \tag{21}$$

[6]Specifically, unlike in (5) we do not consider the time for data and ACK, and do not count twice the times when the busy tone is active.

[7]Note that STEM could be combined with various access protocols, such as for example S-MAC [7]. In this setting, we assume a simple contention-based scheme which enables direct comparison with GeRaF.

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 the number of priority classes $N_p = 4$, the relative size of the relay region $\xi = 0.4$, the relative power consumption in sleep mode, $P_s/P = 0.001$, and $T_{SIG}/T_D = 0.1$ (we assume here for simplicity that all signaling packets are of the same length T_{SIG}); **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 GeRaF and in STEM. In reality, these parameters would be the 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 only remaining independent parameter is the duty cycle. In GeRaF, if we upper bound the energy consumption by neglecting the negative term $-MT_L/2$, 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.

The performance evaluation can therefore be carried out in terms of energy consumption and latency as a function of the duty cycle, with the listening times chosen as just explained. Curves of energy and latency vs. duty cycle, as well as latency vs. energy (while varying the duty cycle), can be provided. As an example, some results are shown in Figures 1 through 4, in which the performance of GeRaF 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 as in GeRaF) becomes dominant, and more so as the network load is higher and the node density is smaller. Note that GeRaF outperforms STEM 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 be independently optimized (notice from Figs. 1 and 2 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 GeRaF may be significantly smaller than that in STEM. To shed some more light on this issue, we plot the trade-off between energy consumption and latency (normalized to the duration of a data packet, T_D) 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 both schemes, we can observe a region in which there is a real trade-off between energy and latency, whereas there exists a saturation point beyond which there is no trade-off as both schemes perform poorly: the latency associated to long sleep times is unacceptable, and the persistence in looking for a relay results in degraded energy performance as well. In the trade-off region, the relative performance of GeRaF and STEM depends on the node density: GeRaF performs better than STEM for sufficiently dense networks, while the opposite is true when the density is small. As shown in the figure, although for relatively sparse networks ($N = 20$) GeRaF and STEM perform approximately the same, for networks with

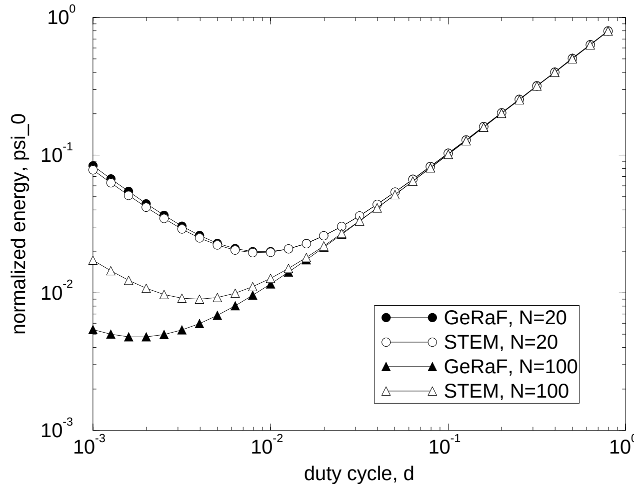

Fig. 1. Average normalized energy consumption, ψ_0, vs. duty cycle, d. GeRaF and STEM compared. $N = 20, 100$, network load 0.01.

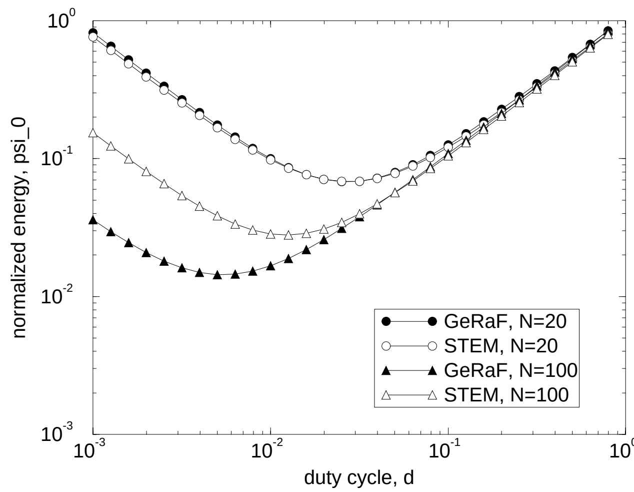

Fig. 2. Average normalized energy consumption, ψ_0, vs. duty cycle, d. GeRaF and STEM compared. $N = 20, 100$, network load 0.1.

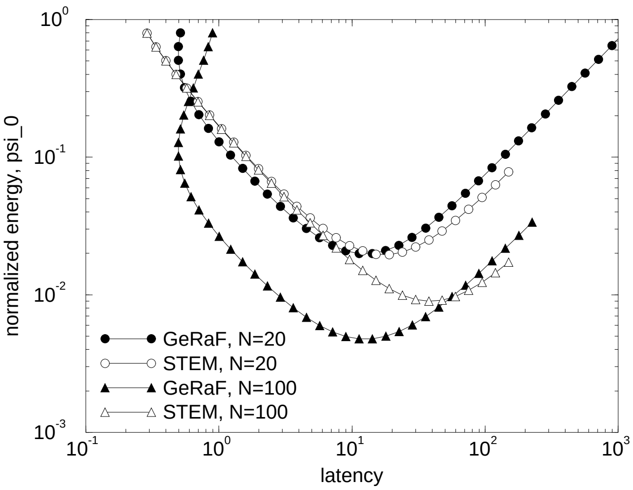

Fig. 3. Average normalized energy consumption, ψ_0, vs. latency (in units of T_D). GeRaF and STEM compared. $N = 20, 100$, network load 0.01.

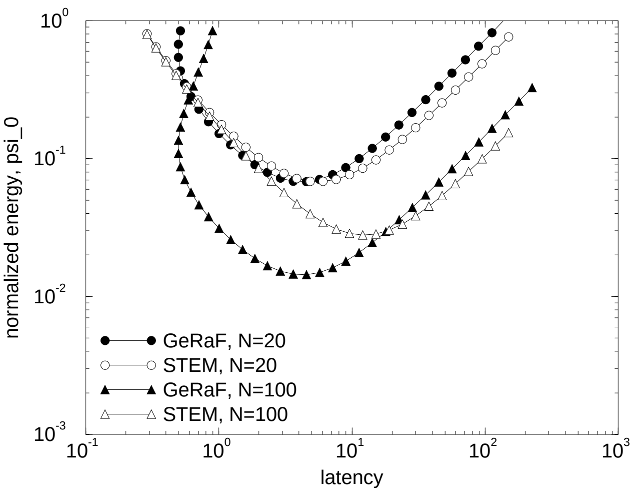

Fig. 4. Average normalized energy consumption, ψ_0, vs. latency (in units of T_D). GeRaF and STEM compared. $N = 20, 100$, network load 0.1.

$N = 100$ nodes per coverage area GeRaF can gain over STEM almost an order of magnitude in latency for comparable energy or in energy for comparable latency. As already mentioned, STEM could be improved by coupling it with GAF, which on the other hand has significant drawbacks in terms of additional signaling and increased number of hops. The proposed scheme therefore appears as a promising alternative for low-power networking.

A. Discussion on the traffic model

In the above results, we have adopted a traffic model in which, when comparing different values of N, we have assumed that the average network traffic remains constant. This is justified by the fact that in this paper we focus on the use of high node densities to save energy without introducing too much latency. In this scenario, deploying more nodes does not lead to more nodes generating more traffic, but rather to more nodes sleeping for a larger fraction of the time, so that the average activity (in terms of both communications activity and data generation) within a given area is unchanged. The other option, i.e., using more nodes to enable more data transfer, is not the main focus here. In any event, note that the curves in Figures 1 through 4 can be used in the latter case as well by just scaling the load by the constant value of N. The comparison between GeRaF and STEM still holds, and therefore most of the above conclusions still apply. In particular, GeRaF is better than STEM for sufficiently dense networks, while the opposite is true for small N.

Another issue related to the traffic model is the packet generation at the sensor nodes. The model considered here assumes that isolated packets are generated, while in the presence of other models (e.g., bursts of packets) the scheme should be changed, e.g., by creating an association between a node and the relay who wins the contention in order to avoid multiple contentions for packets in the same burst.

V. Energy optimization

From the plots shown, there appears to be a minimum in the energy consumption, i.e., there exists an optimal value of the duty cycle which minimizes the energy cost. Here, we investigate the optimization of this parameter.

Since the full expression of ψ_0 for GeRaF is too complex, as a first step we look for some accurate approximation. To this aim, in Figs. 5–7 we plot the following quantities:

$$t_1 = \lambda \left[(e^{\xi M} - 1)^{-1} (3N_p + 1) T_{SIG} + T_D \right] \quad (22)$$

$$t_2 = \lambda (3x + 2) T_{SIG} \quad (23)$$

$$t_3 = \lambda \left[-\frac{M T_L}{2} + 2(1 - e^{-\xi M}) T_D + \left(2M + 2(1 - e^{-\xi M}) \right) T_{SIG} \right] \quad (24)$$

$$t_4 = 3\lambda \xi M (x - 1) T_{SIG} \qquad t_5 = d \quad (25)$$

In addition, we plot the normalized sleep power, P_s/P (sl), the total normalized traffic (tr) estimated as $\lambda N T_{DATAex}$, and the total consumption (tot)

$$\psi_0 = t_1 + t_2 + t_3 + t_4 + t_5 + P_s/P \quad (26)$$

The reason we plot the normalized traffic is that one of the assumptions on which our analysis is based is that the traffic be low, so that the approximations made hold. Plotting the traffic allows us to see in which region the results presented are reasonable and where they may be too pessimistic. We will elaborate on this issue in Section 6, where a more detailed model will be developed whose accuracy does not rely on the low traffic assumption.

From the various figures (as well as from the results obtained in many other cases, not shown here, which follow the same general trend as in these examples), it is clear that for large values of d the contribution of t_5 (energy spent listening) dominates, whereas for low duty cycles the dominant term is t_1, which corresponds to the energy spent looking for a relay. From Fig. 7 we also note that for small traffic and dense networks the effect of the sleep power cannot be neglected. [8]

[8] Note that the case $N = 500$, which may seem extreme, is shown here to demonstrate that the accuracy of the approximation is good over a wide range of the parameter values.

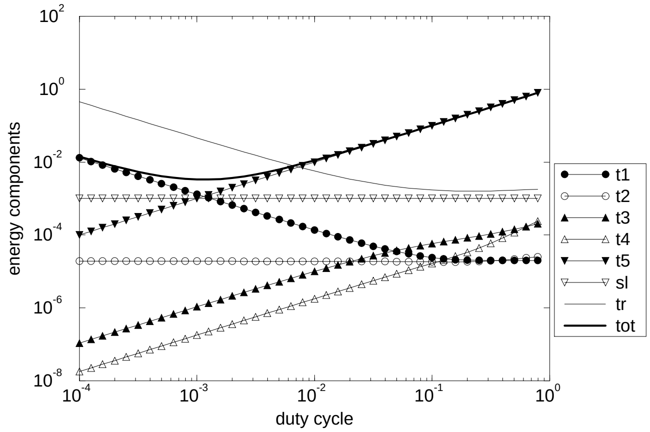

Fig. 5. Components of the normalized average energy consumption vs. duty cycle. $N = 50$, network load 0.001.

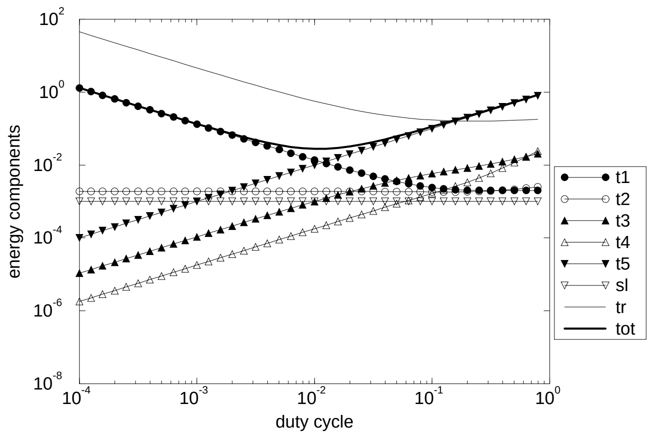

Fig. 6. Components of the normalized average energy consumption vs. duty cycle. $N = 50$, network load 0.1.

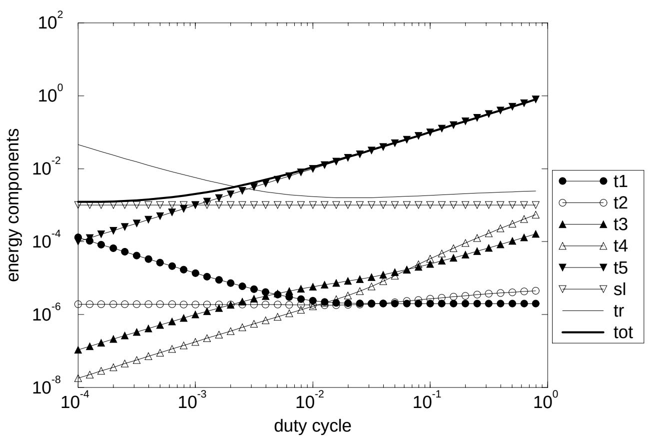

Fig. 7. Components of the normalized average energy consumption vs. duty cycle. $N = 500$, network load 0.001.

In all cases shown, as well as in all our extensive evaluations not depicted here, terms t_2, t_3, t_4 are negligible over the whole range of values considered. Therefore, in order to study the behavior of ψ_0, we can use the approximation

$$\psi_0 \simeq t_1 + t_5 + P_s/P$$

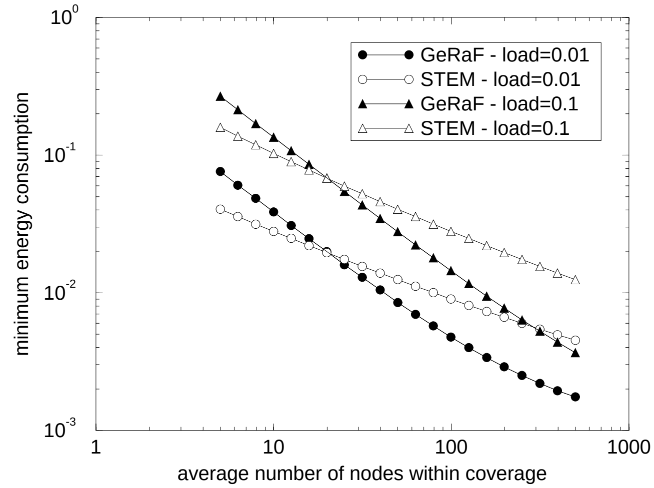

Fig. 8. Optimal normalized average energy consumption, ψ, vs. average number of nodes within coverage, N. Network load 0.01 and 0.1.

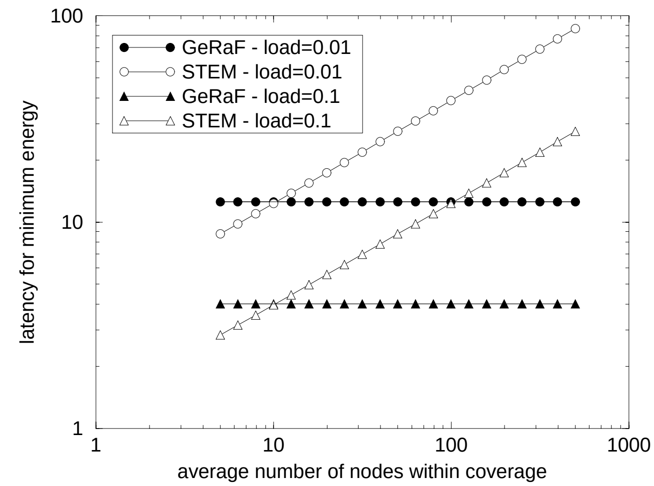

Fig. 9. Latency (in units of T_D) corresponding to optimal energy consumption vs. average number of nodes within coverage, N. Network load 0.01 and 0.1.

$$= \lambda\left[(e^{\xi M}-1)^{-1}(3N_p+1)T_{SIG} + T_D\right] + d + P_s/P \qquad (27)$$

whose accuracy has been tested with excellent results. Based on this expression, and recalling that $M = dN$, we can just differentiate ψ_0 with respect to d and set the derivative to zero, obtaining that the optimal choice of the duty cycle as a function of the various parameters involved is given by

$$d_{opt} = \frac{\log w}{\xi N} \qquad (28)$$

where

$$w = \frac{\alpha + 2 + \sqrt{\alpha(\alpha+4)}}{2}, \qquad \alpha = \lambda(3N_p+1)\xi N T_{SIG} \qquad (29)$$

From the expression of the energy consumption of STEM we obtain the equation

$$\frac{\partial \psi_s}{\partial d} = \lambda T_L\left(-\frac{1}{2d^2}\right) + 1 = 0 \quad \Longrightarrow \quad d_{opt} = \sqrt{\frac{\lambda T_L}{2}} \qquad (30)$$

A. Results

Figure 8 shows the optimized performance of GeRaF and of STEM versus the average number of nodes within coverage, N. In each curve, the value of the total average network load is kept constant, i.e., λ decreases as we move to the right. For each value of N, the optimal duty cycle is computed according to the above formulas and used to compute the energy performance. We note from the above expressions that for GeRaF d_{opt} is approximately inversely proportional to N, as is λ for fixed network load (recall that the network load is defined as $\lambda N T_D$). Therefore, we expect the minimum energy consumption to be inversely proportional to N, as the figure shows. In STEM, on the other hand, d_{opt} is proportional to $\sqrt{\lambda}$, i.e., inversely proportional to $\sqrt{N}$. Looking at the expression for ψ_s, the behavior is itself inversely proportional to $\sqrt{N}$, as the figure shows. Notice that the slopes of the curves corresponding to the two schemes are therefore different, and while STEM shows superior performance for a relatively small number of nodes per coverage area, when the network is more dense GeRaF outperforms STEM, as expected. This is due to the fact that since *any* node can act as a relay, a higher density provides a higher probability that such a node wakes up.

We remark here that the results of Figure 8 are energy-optimized without taking into account latency. A fair comparison should therefore consider latency as well. Figure 9 shows the latency performance which for each N corresponds to choosing d_{opt}. The figure clearly shows that, in the considered case of fixed network traffic, the latency of GeRaF is constant, since choosing d_{opt} (roughly) inversely proportional to N results in a constant value of M, the number of available relays within coverage, which is the key factor in determining latency. On the other hand, in STEM the optimal choice of d results in a value of latency which is proportional to $\sqrt{N}$, as the figure shows. Clearly, when N is larger than about 15-20, STEM has worse energy and latency performance than GeRaF. For a more complete investigation of the trade-off between energy and latency, refer to the already discussed Figures 3 and 4.

Similar results for the case in which λ is kept constant (and therefore the average network load increases with N) are shown in Figures 10 and 11. It can be seen that in this situation STEM chooses a fixed point in the energy-latency space, whereas GeRaF can still benefit from increased density, as in the previous case.

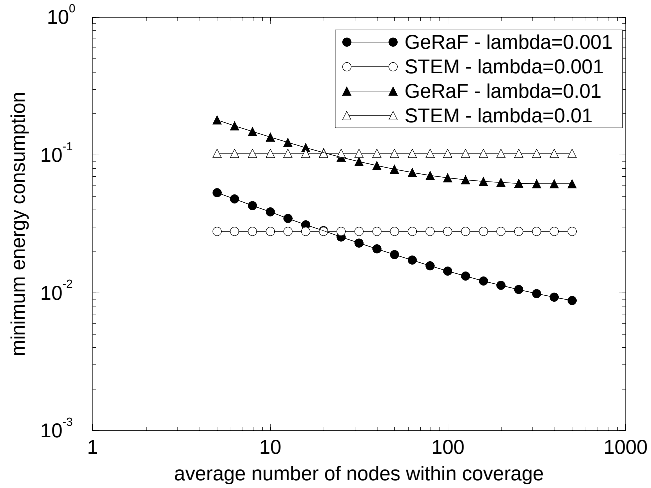

Fig. 10. Optimal normalized average energy consumption, ψ, vs. average number of nodes within coverage, N. Traffic per node $\lambda = 0.001$ and 0.01.

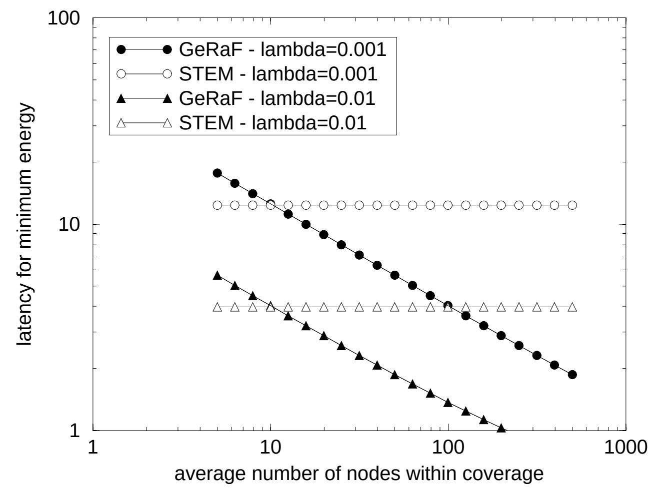

Fig. 11. Latency (in units of T_D) corresponding to optimal energy consumption vs. average number of nodes within coverage, N. Traffic per node $\lambda = 0.001$ and 0.01.

VI. Semi-Markov model

In the previous subsection we have mentioned that the simple analysis presented can be expected to apply only for low traffic in the network, as it assumes that when a packet is ready for transmission the medium is never busy, and neglects other issues which could become relevant if the channel is occupied for a significant fraction of the time. This is the reason why, for low duty cycles and significant traffic per node, the approximate expression (actually, upper bound) for the energy consumption may increase beyond the normalized value of two, which is the maximum possible since the worst possible case is that both radios are always on.

It is therefore important to validate the goodness of the previous analysis and to determine the range of values of the parameters in which the above results are meaningful. In order to do so, we develop a more complete model in which the effect of multiple attempts is accounted for. More specifically, we track the evolution of a node, which can be in one of the following states: *transmitRTS* : the node has decided to start a handshake and sends an RTS; *transmitpkt* : a handshake has been successfully completed and packet transmission starts; *packetready* : the node has a packet ready for transmission; *sleep* : the node is in sleep mode; *listen* : the node is in idle listening mode; *receiveRTS* : an RTS has started while the node was listening, and the node starts receiving it; *receivepkt* : the node has won contention to be a relay and now receives the packet.

We build the transition structure among these states according to the various events which can occur. For each transition we can determine the associated energy consumption as well as other relevant metrics. The resulting semi-Markov model can be solved to obtain the performance results of interest.

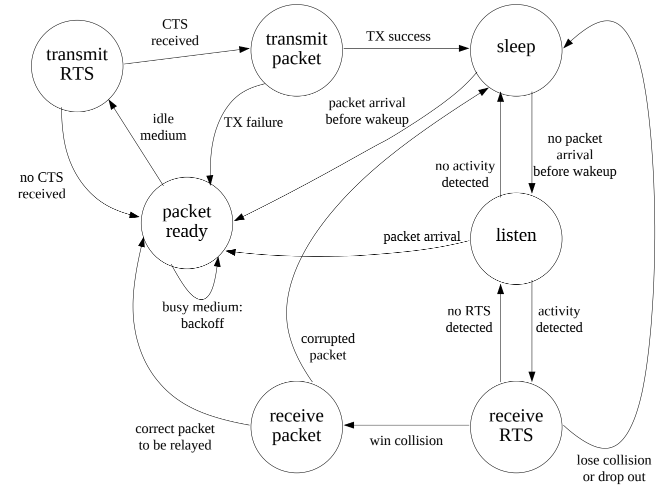

Fig. 12. Transition diagram of the semi-Markov model for GeRaF.

Notice that even this model is not completely accurate, as an exact model would require to keep track of the state of all nodes jointly, a clearly impossible task. However, it does capture important behaviors, e.g., the fact that in a loaded network a node may spend more time in a given state than anticipated.

In the sequel, we derive the transition structure for the semi-Markov model. The model has seven states and sixteen transitions, and is depicted in Figure 12.

transmitRTS **–** This state corresponds to a node which has decided to start a handshake and sends an RTS, i.e., medium sensing has been successfully performed.

The possible transitions in this case are the following. With probability $1 - e^{-\xi M}$, there is a relay available, the node receives a CTS and enters the *transmitpkt* state. Given that there is at least one active user in the relay area, the average time to solve the contention (in number of CTS slots) is given as $x = x_0 + x_1$, where

$$x_0 = \left(1 - e^{-\xi M}\right)^{-1} \sum_{i=0}^{N_p - 1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) \left(1 - e^{-\lambda_{i+1}}\right) i \quad (31)$$

is the average number of empty CTS slots (no relays in the corresponding area), and

$$x_1 = \left(1 - e^{-\xi M}\right)^{-1} \sum_{i=0}^{N_p - 1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) \sum_{k=1}^{\infty} \frac{e^{-\lambda_{i+1}} \lambda_{i+1}^k}{k!} s_k \quad (32)$$

is the average number of CTS slots from the one in which at least one CTS is sent to when the collision is resolved (i.e., a single CTS is sent).[9]

The radio on the data channel is always on, transmitting one RTS and $x - 1$ CTS replies, receiving x_1 CTSs, and listening for x_0 CTSs. [10] On the other hand, the second radio (only used for busy tones) is active only when the first radio does not transmit, i.e., during the x CTS slots. We arrange the time spent by each radio in the four modes (transmit, receive, listen, sleep) into a matrix, where the two rows correspond to the two radios and the columns correspond to the four modes in the given order. Note that the sum of the elements of each row must be the same and is equal to the total average time associated to the considered transition. In this case we have

$$\left(\begin{array}{cccc} \begin{array}{c} T_{RTS} + \\ (x-1) T_{CTSr} \end{array} & x_1 T_{CTS} & x_0 T_{CTS} & 0 \\ & & & \\ x T_{CTS} & 0 & 0 & \begin{array}{c} T_{RTS} + \\ (x-1) T_{CTSr} \end{array} \end{array} \right)$$

The other possibile transition is to state *packetready* , and occurs when no CTS is received, i.e., no relays could be found and the node backs off. This event has probability $e^{-\xi M}$, and the associated times are

$$\left(\begin{array}{cccc} T_{RTS} + N_p T_{CTSr} & 0 & N_p T_{CTS} & T_{backoff} \\ N_p T_{CTS} & 0 & 0 & \begin{array}{c} T_{RTS} + N_p T_{CTSr} \\ + T_{backoff} \end{array} \end{array} \right)$$

transmitpkt **–** This state is entered when a successful handshake has been completed. A data packet is then transmitted and an ACK received. Two events are possible here, i.e., the packet is successfully received with probability P_{succ}, for which

$$\left(\begin{array}{cccc} T_D & T_{ACK} & 0 & 0 \\ T_{ACK} & 0 & 0 & T_D \end{array} \right)$$

and which leads to the *sleep* state; or, the packet is corrupted with probability $1 - P_{succ}$, for which

$$\left(\begin{array}{cccc} T_D & 0 & T_{ACK} & T_{backoff} \\ T_{ACK} & 0 & 0 & T_{backoff} + T_D \end{array} \right)$$

and which leads to *packetready* since a new attempt needs to be scheduled.

packetready **–** This state corresponds to the node having a packet ready for transmission. The action taken is sensing the channel. If the channel is sensed idle for $T_{sensing}$, a transition occurs to state *transmitRTS* , otherwise, the node returns in *packetready* after some backoff time with average $T_{backoff}$. Let P_{idle} be the probability that the channel is sensed idle. Then, with probability P_{idle} there is a transition to *transmitRTS* with times

$$\left(\begin{array}{cccc} 0 & 0 & T_{sensing} & 0 \\ 0 & 0 & T_{sensing} & 0 \end{array} \right)$$

whereas with probability $1 - P_{idle}$ there is a transition to *packetready* with times

$$\left(\begin{array}{cccc} 0 & 0 & T_{sensing} & T_{backoff} \\ 0 & 0 & T_{sensing} & T_{backoff} \end{array} \right)$$

[9]In this section we specifically use λ_j to denote the average number of nodes in $\mathcal{A}_j$, so that the analysis applies to the case in which the priority regions are generally defined.

[10]Note that the actual number of CTSs which the transmitter receives may be less than x_1, since some of the CTS slots during the resolution of a contention may be empty. What we consider here is a conservative approximation, which may be expected to be very tight since the power for receive and listen is almost the same and, in addition, as shown in Section 5, the contribution of this transition is small.

Note that the evaluation of P_{idle} is not easy since the medium is sensed busy if any node is transmitting within range, as well as if any node is sending the busy tone in response to transmissions by other nodes (which need not be within range of the node considered). Notice that a node will be affected by any transmission which started up to T_{DATAex} earlier, where T_{DATAex} is the total time for a data transfer from RTS to ACK. The area in which these data exchanges may affect our node is at least one coverage circle around the node (transmitters are directly heard on the data channel) and at most a circle with twice the radius (where transmitters may trigger busy tones in the coverage area of our node). Therefore, we can say that $e^{-4\lambda N T_{DATAex}} \leq P_{idle} \leq e^{-\lambda N T_{DATAex}}$. These bounds have been found to be tight in the cases of interest.

sleep **–** In this state, both radios are turned off. The state is exited when either a new packet is generated by the node or the node is scheduled to wake up. If λ is the packet arrival rate at each node, and T_{sleep} is the sleep time (assumed constant), a transition to *packetready* occurs with probability $1-e^{-\lambda T_{sleep}}$, with times

$$\begin{pmatrix} 0 & 0 & 0 & x \\ 0 & 0 & 0 & x \end{pmatrix}$$

where

$$x = \frac{1}{\lambda} - \frac{T_{sleep}}{e^{\lambda T_{sleep}} - 1} \tag{33}$$

is the average time to the next packet arrival given that it arrives before sleeping time expires. The other possible transition exiting state *sleep* is to state *listen* (corresponding to no packet arrivals at the node throughout the sleeping period), with probability $e^{-\lambda T_{sleep}}$ and times

$$\begin{pmatrix} 0 & 0 & 0 & T_{sleep} \\ 0 & 0 & 0 & T_{sleep} \end{pmatrix}$$

listen **–** In this state, the node monitors the data channel. If no activity is detected (i.e., no RTS is started) and no packet is locally generated, the node goes back to sleep after T_L, otherwise it takes the appropriate actions. The rate at which new packets are generated by the node itself is λ, whereas the rate at which RTSs are generated by nodes within coverage of the listening node is λN. Therefore, the rate at which the listening period is interrupted is $\lambda(N+1)$. The average time to the next packet arrival at any of the nodes in range (included the node itself), given that one such packet arrives before listening time expires is given by

$$x = \frac{1}{\lambda(N+1)} - \frac{T_L}{e^{\lambda(N+1)T_L} - 1} \tag{34}$$

The probabilities that this packet arrival occurs at the node or at one of its neighbors are $1/(N+1)$ and $N/(N+1)$, respectively.

The possible transitions are therefore to state *sleep* with probability $e^{-\lambda(N+1)T_L}$ and times

$$\begin{pmatrix} 0 & 0 & T_L & 0 \\ 0 & 0 & 0 & T_L \end{pmatrix}$$

and to states *receiveRTS* and *packetready* with probabilities $\frac{N(1-e^{-\lambda(N+1)T_L})}{N+1}$ and $\frac{1-e^{-\lambda(N+1)T_L}}{N+1}$, respectively, and times

$$\begin{pmatrix} 0 & 0 & x & 0 \\ 0 & 0 & 0 & x \end{pmatrix}$$

receiveRTS **–** In this state, after having sensed the start of a RTS, the node receives it. If the node belongs to the relay area for the corresponding packet, it contends for being a relay, otherwise it goes back to sleep. The following events need to be considered: 1. the node is not in the relay area; 2. the node is in the relay area but drops out before it sends a CTS; 3. the node is in the relay area, sends a CTS and loses the collision; 4. the node is in the relay area, sends a CTS and is the winner.

The node is not in the relay area with probability $1-\xi$. In this case, the only time involved is receiving the RTS. Note that while receiving the RTS the node transmits the busy tone on the busy tone frequency.

Conditioned on the node being in the relay area, the probability that the node is in $\mathcal{A}_i$ is proportional to the area of $\mathcal{A}_i$. Let

$$b_i = P[\text{node in } \mathcal{A}_i | \text{node in relay area}] = \frac{\text{area of } \mathcal{A}_i}{\text{relay area}} \tag{35}$$

and

$$c_i = P[\text{node in } \mathcal{A}_j, j=1,\ldots,i | \text{node in relay area}] = \sum_{j=1}^{i} b_j \tag{36}$$

The node is in the relay area but drops out in slot $i+1$ before sending a CTS if there are i empty CTS slots, the $(i+1)$-st is not empty and the node is in $\mathcal{A}_m, m > i+1$. The probability of this event is

$$\xi \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) \left(1 - e^{-\lambda_{i+1}}\right) (1 - c_{i+1}) \tag{37}$$

so that the average probability that a node drops out is

$$p_d = \xi \sum_{i=0}^{N_p-1} (1 - c_{i+1}) \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) \left(1 - e^{-\lambda_{i+1}}\right) \tag{38}$$

and the conditional average number of slots involved is

$$x_d = \frac{\xi}{p_d} \sum_{i=0}^{N_p-1} (1 - c_{i+1}) \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) \left(1 - e^{-\lambda_{i+1}}\right) (i+1) \tag{39}$$

Note that the node never transmits a CTS in any of these slots.

The node is in the relay area, sends a CTS in slot $i+1$ and loses the collision if there are i empty CTS slots, the node selects slot $i+1$, other k nodes select slot $i+1$, and one of them is the winner. The probability of this event is found as

$$\xi \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \frac{e^{-\lambda_{i+1}} \lambda_{i+1}^k}{k!} \frac{k}{k+1} \tag{40}$$

in which we exploited the fact that any of the $k+1$ contending users has the same probability of being the winner. The probability that the node loses the contention is therefore

$$\begin{aligned} p_l &= \xi \sum_{i=0}^{N_p-1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \sum_{k=0}^{\infty} \frac{e^{-\lambda_{i+1}} \lambda_{i+1}^k}{k!} \frac{k}{k+1} \\ &= \xi \sum_{i=0}^{N_p-1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \left(1 - \frac{1-e^{-\lambda_{i+1}}}{\lambda_{i+1}} \right) \end{aligned} \quad (41)$$

and the conditional average number of slots involved is bounded by $x_l = x_{l0} + x_{l1}$, with

$$x_{l0} = \frac{\xi}{p_l} \sum_{i=0}^{N_p-1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \left(1 - \frac{1-e^{-\lambda_{i+1}}}{\lambda_{i+1}} \right) i \quad (42)$$

$$x_{l1} = \frac{\xi}{p_l} \sum_{i=0}^{N_p-1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \sum_{k=0}^{\infty} \frac{e^{-\lambda_{i+1}} \lambda_{i+1}^k}{k!} \frac{k s_{k+1}}{k+1} \quad (43)$$

where s_k is the average time to solve a collision involving k nodes, and is an upper bound here since the node drops out before the collision is resolved. Note that x_{l0} is the average number of slots in which the node certainly does not transmit a CTS, whereas x_{l1} is the average number of slots in which the node *may* transmit a CTS (during collision resolution, the node may decide not to transmit).

The node is in the relay area, sends a CTS in slot $i+1$ and wins the collision if there are i empty CTS slots, the node selects slot $i+1$, other k nodes select slot $i+1$, and our node is the winner. As before, the probability of this event is found as

$$\xi \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \frac{e^{-\lambda_{i+1}} \lambda_{i+1}^k}{k!} \frac{1}{k+1} \quad (44)$$

The probability that the node wins the contention is therefore

$$\begin{aligned} p_w &= \xi \sum_{i=0}^{N_p-1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \sum_{k=0}^{\infty} \frac{e^{-\lambda_{i+1}} \lambda_{i+1}^k}{k!} \frac{1}{k+1} \\ &= \xi \sum_{i=0}^{N_p-1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \left(\frac{1-e^{-\lambda_{i+1}}}{\lambda_{i+1}} \right) \end{aligned} \quad (45)$$

and the conditional average number of slots involved is bounded by $x_w = x_{w0} + x_{w1}$, where

$$x_{w0} = \frac{\xi}{p_w} \sum_{i=0}^{N_p-1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \left(\frac{1-e^{-\lambda_{i+1}}}{\lambda_{i+1}} \right) i \quad (46)$$

$$x_{w1} = \frac{\xi}{p_w} \sum_{i=0}^{N_p-1} \left(\prod_{j=1}^{i} e^{-\lambda_j} \right) b_{i+1} \sum_{k=0}^{\infty} \frac{e^{-\lambda_{i+1}} \lambda_{i+1}^k}{k!} \frac{s_{k+1}}{k+1} \quad (47)$$

As before, x_{w0} is the average number of slots in which the node certainly does not transmit a CTS, whereas x_{w1} is the average number of slots in which the node *may* transmit a CTS (during collision resolution, the node may decide not to transmit, and can still continue and be the winner if no other nodes transmit).

Of the four events above, the first three lead to the *sleep* state. The corresponding probability is $p_d + p_l + (1-\xi)$, the conditional average number of slots in which the node certainly does not transmit a CTS is bounded by

$$x_0 = \frac{p_d x_d + p_l x_{l0}}{p_d + p_l + (1-\xi)} \quad (48)$$

and the conditional average number of slots in which the node may transmit a CTS is bounded by

$$x_1 = \frac{p_l x_{l1}}{p_d + p_l + (1-\xi)} \quad (49)$$

(Note that if the node is not in the relay area, it drops out immediately.) The time for this transition is therefore

$$\begin{pmatrix} x_1 T_{CTS} & T_{RTS} + x T_{CTSr} & 0 & x_0 T_{CTS} \\ \begin{matrix} T_{RTS} + x T_{CTSr} \\ + x_0 T_{CTS} \end{matrix} & 0 & 0 & x_1 T_{CTS} \end{pmatrix}$$

where $x = x_0 + x_1$.

On the other hand, if the node wins the contention, it will go to state *receivepkt* , with probability p_w and time

$$\begin{pmatrix} x_{w1} T_{CTS} & T_{RTS} + x_w T_{CTSr} & 0 & x_{w0} T_{CTS} \\ \begin{matrix} T_{RTS} + x_w T_{CTSr} \\ + x_{w0} T_{CTS} \end{matrix} & 0 & 0 & x_{w1} T_{CTS} \end{pmatrix}$$

where $x_w = x_{w0} + x_{w1}$.

In the above expressions, we assumed that the busy tone starts being active when the RTS starts and is kept active until the node drops out or wins the contention, with the exception of the time during which the data radio is transmitting (i.e., when the node sends CTSs).

As a final remark, note that we are ignoring here the transition from state *receiveRTS* to *listen* which corresponds to the event "no RTS detected" in the diagram of Figure 12, since we assume that once the start of an RTS is detected, the message is in fact an RTS, so the probability of the event "no RTS detected" is zero.

receivepkt **–** In this state, a node receives a data packet while transmitting the busy tone. With probability P_{succ} this packet transmission will be correct and the node will transition to state *packetready* with times

$$\begin{pmatrix} T_{ACK} & T_D & 0 & 0 \\ T_D & 0 & 0 & T_{ACK} \end{pmatrix}$$

whereas with probability $1 - P_{succ}$ this packet transmission will be corrupted, the node will go back to sleep and no ACK will be generated, which corresponds to a transition to state *sleep* with times

$$\begin{pmatrix} 0 & T_D & 0 & 0 \\ T_D & 0 & 0 & 0 \end{pmatrix}$$

Note that in the above we assumed that a node which correctly receives a packet will immediately try to forward it, i.e., a correct packet reception leads to state *packetready* .

A. Performance analysis

From the transition structure developed above, it is possible to find many metrics of interest. In particular, it is possible to assign to each transition an average energy cost, by appropriately weighing the average times the two radios spend in the different functions. By doing so, it is possible to find the average energy consumption by using the theory or renewal reward processes [13]. Note that in the above analysis the amounts of time either radio spends in each state are separately accounted, so that it would be straightforward to consider different power consumptions among the different modes of operation and between the two radios.

It is also possible to find the average latency, defined as the time from when a packet is generated to when the successful transmission of the packet starts. In this case, the latency is given by the first passage time from state *packetready* to state *sleep* , minus $(T_D + T_{ACK})$. The first passage time from state i to state j of a semi-Markov chain, θ_{ij}, can be found by solving the following set of equations [13]:

$$\theta_{ij} = \tau_i + \sum_{r \neq j} P_{ir} \theta_{rj} \tag{50}$$

for all states i, where τ_i is the average time spent in state i when it is entered, [11] and P_{ij} are the transition probabilities of the embedded Markov chain.

Similarly, it is possible to find the average energy cost of correctly delivering a packet by finding the "first passage time" from state *packetready* to state *sleep* , where instead of the actual time metrics, we use on each transition the energy metrics.

Example results from this analysis are plotted in Figs. 13 and 14 where the normalized energy consumption from the simplified analysis and that from the semi-Markov model are compared. Although for low duty cycles the behavior may be significantly different when the network is not dense, the simplified analysis can accurately predict the behavior in a large fraction of the range and, more importantly, it accurately predicts the location of the minimum, i.e., the value of d_{opt} given above is in fact accurate in all cases, as also confirmed by our other extensive evaluations not shown here due to space constraints.

Similarly, a semi-Markov model can be developed for STEM as well. Note that the more complicated part of the above analysis is the one due to the CTS contention, which is not part of STEM. The resulting analysis is therefore much simpler for STEM than it is for GeRaF, and is not pursued here due to space constraints.

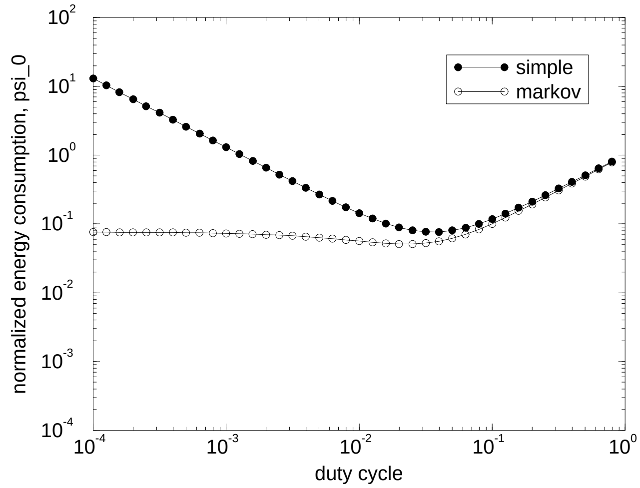

Fig. 13. Normalized average energy consumption, ψ_0, vs. duty cycle. Simple analysis and semi-Markov model compared. $N = 5$, network load 0.01.

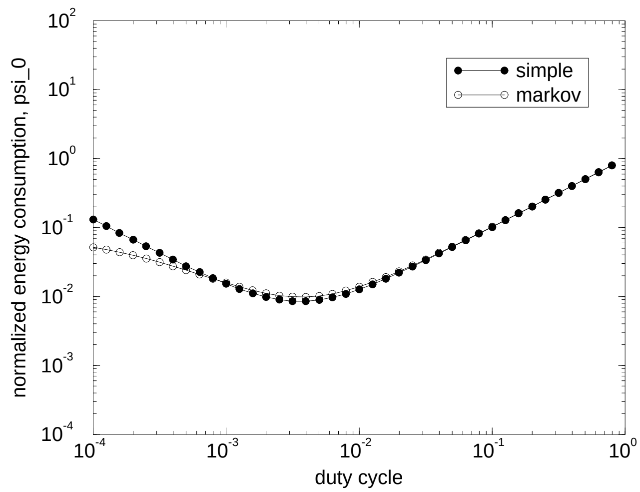

Fig. 14. Normalized average energy consumption, ψ_0, vs. duty cycle. Simple analysis and semi-Markov model compared. $N = 50$, network load 0.01.

VII. Simulation results

In the previous section, we have provided some validation of the simple analysis by means of a more sophisticated model which takes into account the actual time a node spends in any given state. On the other hand, the semi-Markov analysis is itself not exact, since it focuses on the behavior of a single node while essentially assuming that all other nodes operate in steady-state. In order to further verify that the insights provided by our analysis are correct, we set up a simulation program (written in C++) which implements all the details of the proposed MAC and forwarding mechanisms. In particular, the simulation takes correctly into account all interactions among nodes (e.g., backoff effects). On the other hand, in order to first focus on the effects of the detailed MAC mechanisms, the radio modeling is still limited to circular coverage areas. The simulated scenario consists of a square area in which nodes are uniformly distributed. In order to observe multihop behavior, we set the side of the square to eight times the coverage radius (in the simulation we used a 400×400 meter area and a coverage radius of 50 m). The number of nodes is then chosen based on the value of N selected. The data rate is 19.2 kbps, the data packet is 1000 bits and all signaling packets are 100 bits. In view of the low data rate, the sensing time is ignored. The relay area is divided

[11] This time is computed for each state i by taking the time matrices for all outgoing transitions, averaging them with respect to all possible destinations, and taking the sum of either row [13].

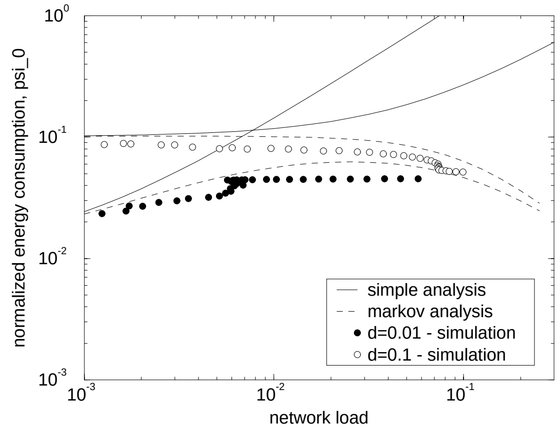

Fig. 15. Normalized average energy consumption, ψ_0, vs. the network load. Simple analysis semi-Markov model compared with simulation points. $N = 5$, duty cycle 0.01 and 0.1.

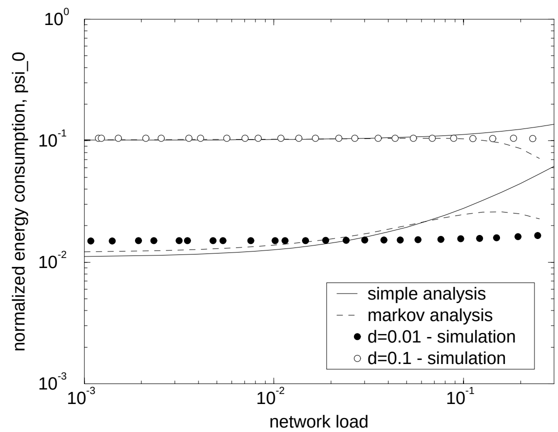

Fig. 16. Normalized average energy consumption, ψ_0, vs. the network load. Simple analysis semi-Markov model compared with simulation points. $N = 50$, duty cycle 0.01 and 0.1.

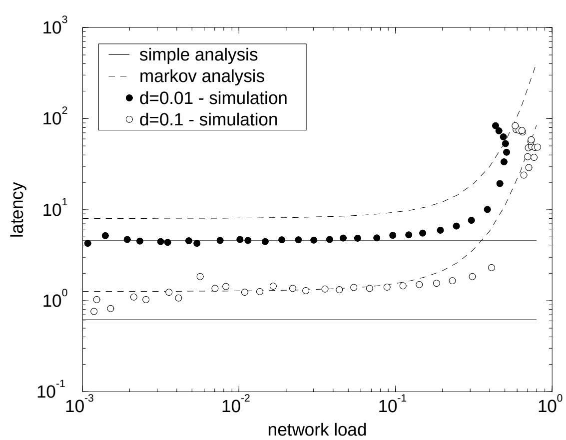

Fig. 17. Latency (in units of T_D) vs. the network load. Simple analysis semi-Markov model compared with simulation points. $N = 50$, duty cycle 0.01 and 0.1.

in four regions as detailed in the protocol description.

While we defer a detailed simulation study of the protocol, which can be expected to reveal many aspects of the protocol details which cannot be studied by analysis, in this section we provide some preliminary results to show that the performance results predicted by our analytical approach are confirmed by simulation points. Figures 15 and 16 report the normalized energy consumption of GeRaF, ψ_0, as a function of the network load, $\lambda N T_D$, for two different network densities, $N = 5$ and $N = 50$, and for two values of the duty cycle, $d = 0.01$ and 0.1. The curves shown are obtained by both analytical approaches presented in this paper, and simulation points are included. [12] From these figures we can observe that, as already pointed out, the simple analytical model may significantly overestimate the energy consumption for sparse networks, while on the other hand the more detailed semi-Markov model closely matches the simulation points, and correctly predicts the network behavior in the whole range of parameters presented. The latency results of Figure 17 for $N = 50$ and for $d = 0.01$ and 0.1 also show that the analytical approaches identify the right trend. These results show the accuracy of the semi-Markov analysis, and help in determining where the simplified analysis (which makes it possible to write closed-form expressions) can be applied.

A detailed simulation campaign is needed in order to verify the sensitivity of the system behavior with respect to the fine details of the protocol, as well as to understand the protocol behavior in extreme situations (e.g., very sparse networks or very high load) where the analysis may become less accurate.

VIII. Conclusions and future work

In this paper, we considered 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 the idea of geographic random forwarding was proposed, and an approximate analysis of its energy and latency performance was provided. The proposed scheme was compared with STEM, and was shown to perform significantly better for sufficient node density. Optimization of the duty cycle for the two schemes was performed, and the obtained optimized performance results were compared. Finally, a more accurate analysis via an elaborate semi-Markov model for the sensor node evolution was proposed and both analytical approaches were validated by means of simulations.

Future work involves several refinements and extensions of the above work, as well as validation of the analytical results by more comprehensive simulations where the simplifying assumptions employed in the analysis are relaxed and more realistic channel models are considered. In particular, issues which should be considered are the development of a semi-Markov model for STEM, the consideration of various other metrics, the coupling of the energy-latency analysis with the multihop scenario, and the effect of different traffic models, e.g., when packets are generated in bursts or by nodes which are in the same geographical area.

[12] The results shown are those obtained from single simulation runs. The statistical significance has been checked by observing that simulations with different seeds provide essentially the same results.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for the many insightful comments which have greatly contributed to the improvement of the paper. Thanks are also due to Mr. Raffaele Rugin for his help in producing the simulation results.

REFERENCES

[1] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, Robert Morris, Span: An energy-efficient coordination algorithm for topology maintenance in Ad Hoc wireless networks, In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[2] Ya Xu, John Heidemann, Deborah Estrin, Geography-informed energy conservation for Ad Hoc routing, In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[3] Curt Schurgers, Vlasios Tsiatsis, 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.

[4] Eugene Shih et al., Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[5] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie. Protocols for self-organization of a wireless sensor network. *IEEE Personal Communications*, 7(5):16–27, 2000.

[6] Chunlong Guo, Lizhi Charlie Zhong, and Jan M. Rabaey. Low Power Distributed MAC for Ad Hoc Sensor Radio Networks. In *Proceedings of IEEE GlobeCom 2001*, San Antonio, 2001.

[7] Wei Ye; Heidemann, J.; Estrin, D., An energy-efficient MAC protocol for wireless sensor networks, *Proceedings of INFOCOM 2002*, New York NY, June 2002, p. 1567–1576.

[8] Alec Woo and David Culler. A transmission control scheme for media acces in sensor networks. In *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, Rome, Italy, July 2001.

[9] M. Zorzi, R.R. Rao, Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: multihop performance, in *IEEE Trans. on Mobile Computing*, this issue.

[10] J.P. Monks, V. Bharghavan, W.W. Hwu, A power controlled moltuple access protocol for wireless packet networks, *Proceedings of INFOCOM 2001*, pp. 219–228.

[11] 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.

[12] 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.

[13] R.A. Howard, *Dynamic probabilistic systems*, John Wiley & Sons, 1971.

Michele Zorzi (S'89–M'95–SM'98) was born in Venice, Italy, in 1966. He received the Laurea Degree and the Ph.D. in Electrical Engineering from the University of Padova, Italy, in 1990 and 1994, respectively. During the Academic Year 1992/93, he was on leave as a graduate student at the University of California, San Diego (UCSD), where he did research on multiple access in mobile radio networks. In 1993, he joined the faculty of the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy. After spending three years with the Center for Wireless Communications at UCSD, in 1998 he joined the School of Engineering of the Università di Ferrara, Italy, where he is currently a Professor. His present research interests include performance evaluation in mobile communications systems, multiple access in mobile radio networks, ad hoc and sensor networks, and energy constrained communications protocols.

Dr. Zorzi is the Editor-In-Chief of the *IEEE Wireless Communications Magazine*, and currently serves on the Editorial Boards of the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Mobile Computing*, the *Wiley Journal of Wireless Communications and Mobile Computing* and the *ACM/URSI/Kluwer Journal of Wireless Networks.* He was also guest editor for special issues in the *IEEE Personal Communications Magazine* (Energy Management in Personal Communications Systems) and the *IEEE Journal on Selected Areas in Communications* (Multi-media Network Radios). E-mail: zorzi@ing.unife.it

Ramesh R. Rao received his Honours Bachelor's degree in Electrical and Electronics Engineering from the University of Madras in 1980. He did his graduate work at the University of Maryland, College Park, Maryland, receiving the MS degree in 1982 and the Ph. D. degree in 1984. Since then he has been on the faculty of the department of Electrical and Computer Engineering at the University of California, San Diego, where he is currently Professor and Director of the San Diego Division of the California Institute of Telecommunications and Information Technology. Prior to this appointment, he served as the Director of the UCSD Center for Wireless Communications and was was the Vice Chair of Instructional Affairs in the Department of Electrical and Computer Engineering.

His research interests include architectures, protocols and performance analysis of wireless, wireline and photonic networks for integrated multi-media services.

He served as the Editor of the Information Theory Society Newsletter from 1993 to 1995 and is the founding Web Editor of the Information Theory Society web site. He has been a senior member of the IEEE since 1990 and was elected to the IEEE Information Theory Society Board of Governors (1997 to 1999 and 2000 to 2002). He is the Editor for Packet Multiple Access of the IEEE Transactions on Communications and is a member of the Editorial Board of the ACM/Baltzer Wireless Network Journal as well as IEEE Network magazine.