A new contention-based MAC protocol for geographic forwarding in ad hoc and sensor networks*

Michele Zorzi[†]

[†]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

August 29, 2003

Abstract

In this paper, we consider a novel forwarding technique based on geographical location of the nodes involved and random selection of the relaying node via contention among receivers. A new collision avoidance scheme based on this idea is described in detail, and an approximate analysis is provided. The proposed scheme is compared with a similar solution basd on busy tones, as well as with STEM, and is shown to perform well for sufficient node density. Compared to the previously proposed protocol, the one presented here only needs one radio, thereby greatly simplifying the hardware.

1 Introduction and system model

Energy consumption 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.

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

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.

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 more hops are needed to cover a given distance) and potentially large delays may be introduced in STEM (in order to wait for a node to wake up).

We proposed an alternative solution, called Geographic Random Forwarding (GeRaF) in [4], 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 to assess its own priority in acting as a relay for that message. Based on the knowledge of the positions of the sensors involved, each node which hears a message can determine which region it belongs to. 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.

The collision avoidance MAC scheme proposed in [4] relies on busy tones to solve the hidden terminal problem, thereby needing two radios. In this paper we propose a modified scheme which only needs a single radio and uses an appropriately tuned sensing time to avoid collisions. A simple analysis is reported and comparisons are made with the original scheme [4] and with STEM [3]. The numerical results show that, although the presented scheme exhibits a somewhat degraded performance compared to the busy-tone solution, it still does better than STEM in dense networks, while requiring a simpler hardware.

We assume here that nodes are randomly distributed throughout the network according to a Poisson process and that propagation can be characterized by coverage circles (extension to Rayleigh fading is under study). The node density is expressed in terms of average number of nodes within the coverage area, N. Each node will be active only a fraction d of the time, so that the actual number of available relays within coverage is $M = dN$. In addition, we assume that for relaying purposes only a fraction $\xi = 0.4$ of the coverage area is actually considered, which approximately corresponds to the portion of it whose points are closer to the destination than the transmitter ("relay area" — we assume here that packets are never allowed to move farther away from the destination).

2 A new collision avoidance scheme for GeRaF

We consider a scheme which uses carrier sense before transmission. Traditional carrier sense schemes only partially avoid collisions and give no strict guarantee against the hidden terminal problem. Indeed, the fact that nodes are not always on makes traditional RTS/CTS-based collision avoidance mechanisms ineffective since a node may wake up after a CTS was issued. This could be solved by synchronizing all nodes as in [5], which requires additional signaling and complexity. As an alternative, busy tones could be used as in GeRaF [4], but this has the potentially significant drawback of requiring an additional radio. While sensor nodes with two radios have been reported in the literature [3], in some cases it may be necessary or desirable to be able to operate with only one. In this section, we propose a MAC protocol similar to GeRaF [4], but able to work with a single radio ("GeRaF-1R").

The scheme is based on geographic random forwarding [4], where data packets are routed by selecting the relay node which is most favorably located towards the destination. This selection is made based on the relative location of transmitter, relay and destination. More specifically, after receiving a request-to-send (RTS) packet, all active nodes within range will evaluate their own priority in acting as a relay based on their location, and contend with other potential relays to become the designated relay for that packet. This *receiver contention* procedure is what makes our protocol substantially different from other carrier-sense based schemes. The idea is that of using whatever node is available at any given time without waiting for a specific node to wake up or trying to acquire topology knowledge in a dynamic environment.

The protocol works as follows. When a node has a packet to send, it listens to the radio channel. If no activity is sensed for a time duration T_{sens}, the node can start its transmission activity. Otherwise, the node backs off and schedules a reattempt at a later time. The collision avoidance feature of this scheme is based on the relationship between the sensing time and the transmission schedule followed by active nodes. More specifically, the sensing time should be long enough to overlap with both the sender's transmissions and the receivers' replies during the packet exchange. This avoids that periods in which a node is receiving are interpreted as idle channel by its neighbors.

2.1 Transmitter

When a sleeping node has a packet to send, it transitions to the active state and monitors the radio channel for T_{sens} seconds. If during that time some activity is detected, the node backs off and reschedules an attempt at a later time. If on the other hand the channel is 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 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 ("empty cycle"); 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 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.

In order to compute the amount of time the radio is active (transmitting, receiving or listening) as a result of the generation of a packet, consider the following (a similar approach was used in [4]). First of all, there is a chance that no relays are found in the relay area, which leads to an empty cycle. The average number of such empty cycles is $(e^{\xi M}-1)^{-1}$, where ξM is the average number of nodes in the relay area, and this accounts for a total radio activity time equal to

$$(e^{\xi M}-1)^{-1}(T_{sens}+T_{RTS}+N_p(T_{CTS}+T_{CTSr})) \tag{1}$$

where $(T_{sens}+T_{RTS}+N_p(T_{CTS}+T_{CTSr}))$ is the time during which the radio is on in each of those cycles. Note in fact that for an empty cycle we have a listening activity for T_{sens} and then an RTS packet (transmitting), followed by exactly N_p CTS slots which remain empty (listening) and N_p CTS replies (transmitting). Finally, note that unlike in [4] we count here all times involved only once, since there is only one radio.

After a number of empty cycles (if any) we have a successful cycle, i.e., one in which there is at least one active node in the relay area. [1] The duration of the successful cycle is found as

$$T_{sens}+T_{RTS}+xT_{CTS}+(x-1)T_{CTSr}+T_D+T_{ACK} \tag{2}$$

where the average number of CTS slots needed to finally have a contention winner is denoted by x (see [4] for its detailed expression). The reason we count $x-1$ CTS replies is because the start of the data packet acts itself as the CTS reply for the successful CTS slot.

[1] We assume here that the dynamics of the node sleeping activity is such that after an empty cycle the next attempt at sending the RTS will see an independent set of relays.

The total radio activity time related to the generation of one packet is then expressed as

$$\begin{aligned} t_T &= (e^{\xi M}-1)^{-1}(T_{sens}+T_{RTS}+N_p(T_{CTS}+T_{CTSr})) \\ &+T_{sens}+T_{RTS}+xT_{CTS}+(x-1)T_{CTSr}+T_D+T_{ACK} \end{aligned} \tag{3}$$

Note that we consider here the typical case in which the sensor network is mostly sensing and the generated traffic is low, so that we can ignore the event that when a node has a packet to transmit it will find the channels busy. A more refined model as in [4] can be used to study the case of higher traffic, but such a study is out of the scope of this paper.

2.2 Receiver

Each node will (more or less) periodically wake up and put itself in the listening mode. If nothing happens throughout the listening time, T_L, 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. 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 subdividing the relay region into 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 $j > i, i = 1, \ldots, N_p - 1$ [4].

Let $\mathcal{A}_i$ the region the node belongs to. That means that the first opportunity that node has to volunteer as a relay is in the i-th CTS slot, assuming that no CTSs were sent in CTS slots 1 through $i-1$. In fact, if some CTSs were sent before CTS slot i, this would mean that some higher priority region is non-empty and therefore the node in question should drop out. Therefore, our node of priority i will listen for the CTS replies in slots 1 through $i-1$, until either it hears something different from a CONTINUE message (in which case it drops out) or it receives $i-1$ CONTINUE messages (in which case it sends its own CTS in slot i).

If the node participates in the contention (i.e., it belongs to the one with highest priority among the non-empty regions), two events may happen. If it is the only one sending a CTS in the i-th CTS slot, it is the winner and will receive the packet start. Otherwise, if other users also send a CTS in the same slot, a collision occurs (the CTS reply will be "COLLISION") and a binary splitting collision resolution algorithms is executed. In this case, all nodes involved will decide with probability 1/2 whether or not to send again in the next slot. If nobody sends, this random decision is again executed in the following slot. Otherwise, only those who have sent will survive and all others will drop out, until there is a single survivor.

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.

Following an approach as in [4], we can evaluate the average time during which the receiver's radio is active following a wake-up event:

$$\begin{aligned} t_\ell &= p_0 T_L + (1-p_0)\left[\frac{T_L}{2} + T_{RTS} \right. \\ &+ \frac{1-e^{-\xi M}}{M}\left(xT_{CTS} + (x-1)T_{CTSr} + T_D + T_{ACK}\right) \\ &\left. \qquad + \frac{\xi M - (1-e^{-\xi M})}{M}(x-1)(T_{CTS} + T_{CTSr})\right] \\ &= T_L + (1-p_0)\left[\xi(x-1)(T_{CTS} + T_{CTSr}) \right. \\ &\left. + T_{RTS} - \frac{T_L}{2} + \frac{1-e^{-\xi M}}{M}\left(T_{CTS} + T_D + T_{ACK}\right)\right]. \end{aligned} \tag{4}$$

where the first term $p_0 T_L$ corresponds to no activity detected ($p_0 = e^{-\lambda N T_L}$ is the probability of this event). If on the other hand some activity is detected (after having listened for $T_L/2$ on average and having received the RTS), three cases are possible: i) the node is not in the relay area (with probability $1-\xi$) and drops off immediately; ii) the node is in the relay area but loses the contention; iii) the node is in the relay area and wins the contention. The rest of the expression in brackets in (4) accounts for cases ii) and iii).

2.3 Collision avoidance feature

In this section, we discuss the collision avoidance property of the proposed scheme, and show how to choose T_{sens} in order to guarantee that no collisions occur. Since we cannot rely on all nodes having received the RTS/CTS, we must provide a mechanism to avoid that a node which wakes up (and therefore is unaware of previous RTS/CTS exchanges) produces interference at nodes already active in a packet exchange. In order to do so, we use a sensing time T_{sens} which is the amount of time a node has to listen for an idle channel before being allowed to transmit a RTS packet. Collisions are avoided if T_{sens} exceeds the maximum length of time a node may be continuously listening without transmitting. That is, if a node detects an idle channel for T_{sens}, this means that the channel is in fact idle (i.e., there are no hidden terminals).

First of all, it is important to identify which nodes need to be protected and when. A slightly conservative approach is to block the following nodes

- all nodes within range of the transmitter for the whole duration (from RTS to ACK)
- all nodes within range of the receivers as long as the receivers are involved in the contention.

The latter bullet means that each contender should be protected as long as it is still a valid contender, whereas it no longer needs to be protected after it has dropped off. For the actual winner, this means to be protected through the whole exchange.

In the transmitter activity, the radio transmits the RTS, all replies, and the data packet, whereas it receives the CTSs and the ACK. As long as T_{sens} exceeds the duration of CTS and ACK packets, a sensing node will always detect a busy channel if it is within range of the transmitter, which avoids the hidden terminal problem for the transmitter.

Consider now a receiver (potential relay). If this receiver is located in the priority region of choice (the one with highest priority among those non-empty), it must be able to follow all sender's replies and to contend.

If i is the priority region in question, the receiver will not be sending anything until it is its turn to send a CTS, i.e., in the i-th CTS slot. This means that T_{sens} must be at least $T_{RTS} + (i-1)(T_{CTS} + T_{CTSr})$ in order to protect this receiver. In order for this to work in all cases, we must choose $T_{sens} \geq T_{RTS} + (N_p - 1)(T_{CTS} + T_{CTSr})$ for the worst case. Note that this choice will also protect all those potential relays that do not get to contend but drop off after hearing a packet start or a COLLISION message.

If the receiver sending the CTS is the only one conteding, it is the winner and the data packet transmission starts. In order for the packet to be correctly received, we must have $T_{sens} \geq T_D$, i.e., a user who senses the channel after a CTS has been successfully sent must not be allowed to send an RTS while the data packet is being transmitted.

If on the other hand the receiver contends with others, in the collision resolution algorithm CTSs are sent by the surviving terminals, so that who continues contending is protected by its own CTS, as long as $T_{sens} \geq T_{CTSr}$, while whoever does not send a CTS drops off and therefore does not need protection. There is only one case in which this does not apply: in the collision resolution algorithm, following a collision all nodes involved decide whether or not to continue with probability 1/2. It is possible that none of the nodes involved decides to continue. In order for the scheme to lead to a winner, in this particular case the resulting empty CTS slot is not counted, and all nodes involved in the collision in the previous CTS slot will make another independent decision. This may lead to a number of empty CTS slots in the unlikely event that this happens a number of times in a row. In theory, this could result in an arbitrarily long idle channel time (as perceived by nodes within range of the receivers but not of the transmitter), which may therefore exceed T_{sens}. On the other hand, the probability that this happens is very small, and if T_{sens} is equal to a few CTS durations this event can be neglected. [2]

In view of the above discussion, we have that the length of T_{sens} must be no shorter that T_D and $T_{RTS} + (N_p - 1)(T_{CTS} + T_{CTSr})$, i.e., we should choose

$$T_{sens} = \max\{T_D, T_{RTS} + (N_p - 1)(T_{CTS} + T_{CTSr})\} \tag{5}$$

[2] One could deterministically force a CTS after a certain number of empty CTS slots during the collision resolution procedure, thereby avoiding even this low-probability event.

which typically leads to $T_{sens} = T_D$. With this choice, even in the absence of busy tones, the hidden terminal problem is completely avoided in this scenario, and therefore the protocol correctly operates with a single radio.

Note that the introduction of T_{sens} is expected to produce some performance degradation since it implies a longer active time for the transmitter's radio, as well as some additional latency. This is the price to pay for the hardware simplicity. In order to mitigate this negative effect, one could accept some probability of being hit by hidden terminals if the resulting performance degradation is outweighed by the benefit of avoiding a long T_{sens}. Alternatively, one could force some additional messages to go out so as to reduce T_{sens}. For example, for long packets one can divide them into pieces, and separate these pieces by short gaps during which no useful data is transmitted by the sender while the receiver sends out a pulse of energy (the equivalent of a "busy tone"), which reduces the requirements on T_{sens} which must now be as long as a single piece instead of as the whole packet. The effectiveness of this solution depends on the energy cost of switching the radio, which needs careful evaluation and is left for future study. We will show in this paper that even the simple scheme with no busy tones at all can achieve reasonable performance in realistic scenarios.

3 Analysis

We now give an approximate analysis as in [4], 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.

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

$$E_{tot} = N_T E_T + N_\ell E_\ell + T_s P_s \tag{6}$$

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 time spent in sleep mode, while P_s is the power corresponding to that state.

We focus here on the typical situation in which the network is sensing most of the time, and messages are produced only infrequently. Therefore, we ignore the event that users who want to transmit cannot do so because the channel is already busy.

In this case, the total time associated to the transmitting activity of a node is then given by (3) and if we assume that the power spent in transmit, receive and listen mode is the same P for all, the total energy spent every time a node wants to transmit a packet is

$$E_T = t_T P \tag{7}$$

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

$$\frac{N_T E_T}{t} = \lambda P t_T \tag{8}$$

where t_T is given in (3) and λ is the packet arrival rate at each node.

Similarly, for the receive activity, the total average contribution to the total average power consumption E_{tot}/t can be found as $E_\ell N_\ell / t$ and is bounded by

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

where we used the fact that

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

Finally, since the node will be sleeping most of the time, we can use the approximation $T_s/t \simeq 1$.

The total normalized average energy consumption to then given by

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

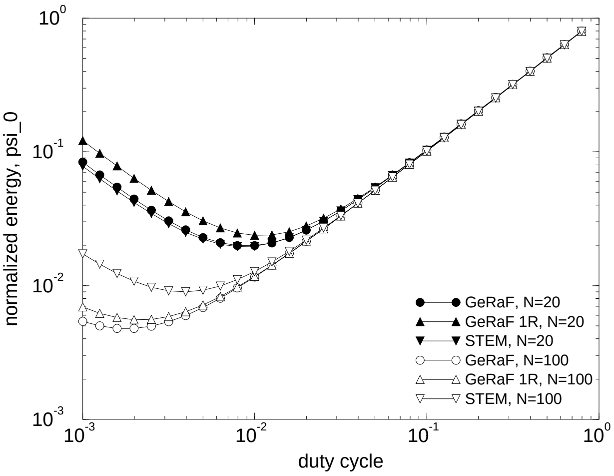

Figure 1: Average normalized energy consumption, ψ_0, vs. duty cycle, d. GeRaF, GeRaF-1R and STEM compared. $N = 20, 100$, network load 0.01.

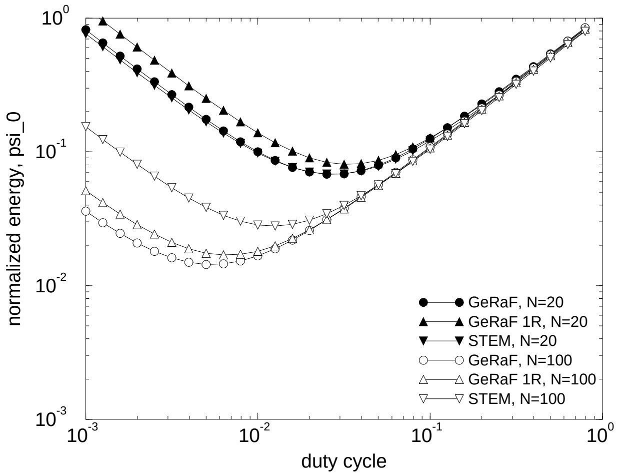

Figure 2: Average normalized energy consumption, ψ_0, vs. duty cycle, d. GeRaF, GeRaF-1R and STEM compared. $N = 20, 100$, network load 0.1.

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).

Latency, defined as the time it takes from when a node starts the packet transmission handshake to when the transmission of the actual data packet starts, can be computeed as t_T minus the time for data and ACK, i.e.,

$$\begin{aligned}\overline{T}_S &= (e^{\xi M} - 1)^{-1}(T_{sens} + T_{RTS} + N_p(T_{CTS} + T_{CTSr})) \\ &\quad + T_{sens} + T_{RTS} + xT_{CTS} + (x-1)T_{CTSr} \qquad (12)\end{aligned}$$

4 Performance comparison

In this section, we give some numerical results for the schemes considered, and provide a comparison between them. For the new MAC proposed (GeRaF-1R) we use the above analysis, whereas for STEM and the classic version of GeRaF we use the results in [4]. In all the following results, 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

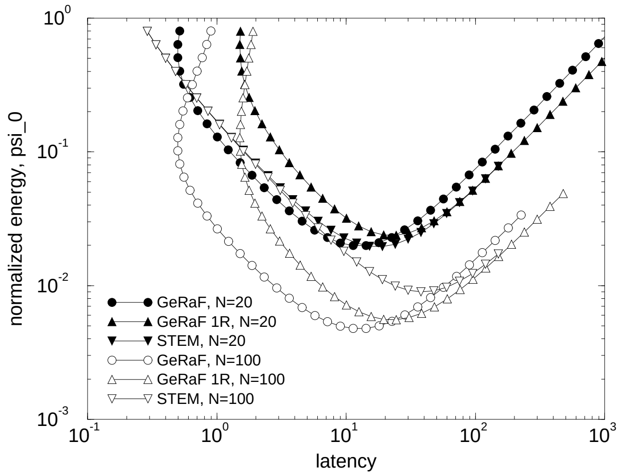

Figure 3: Average normalized energy consumption, ψ_0, vs. latency (in units of T_D). GeRaF, GeRaF-1R and STEM compared. $N = 20, 100$, network load 0.01.

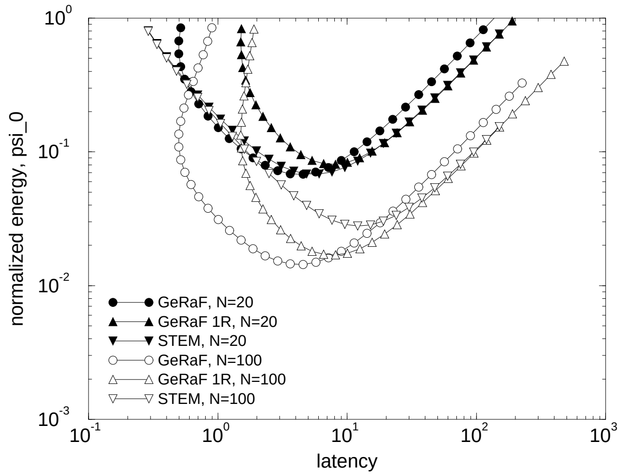

Figure 4: Average normalized energy consumption, ψ_0, vs. latency (in units of T_D). GeRaF, GeRaF-1R and STEM compared. $N = 20, 100$, network load 0.1.

T_{SIG}).

Results for the average normalized energy consumption are shown in Figures 1 and 2. The loss in performance due to the single-radio operation compared to the scheme with the busy tones [4] can be quantified. It is important to note that essentially the conclusions in [4] still hold, i.e., the proposed scheme outperforms STEM [3] whenever the node density is sufficient, which in this case corresponds to more than about 20 neighbors. [3]

The tradeoff between energy and latency is shown in Figures 3 and 4. While it is apparent that for small duty cycles (upper left region of the curves) the performance hit may be significant, we are more interested in the low-energy portion of the curves, where the latency degradation is not exceedingly large.

An approximate energy optimization can be carried out following [4] to compute the optimal duty cycle as a function of the system parameters. The minimum energy as a function of the node density if reported in Figure 5, where one can see that the minimum energy is only slightly larger in the case with a single radio than it is in the case with two radios. In both cases, STEM is outperformed when the node density is sufficient, due to the different slope of the minmium energy vs. node density curves.

[3]Note that these neighbors are not all active at the same time, but only during a fraction *d*.

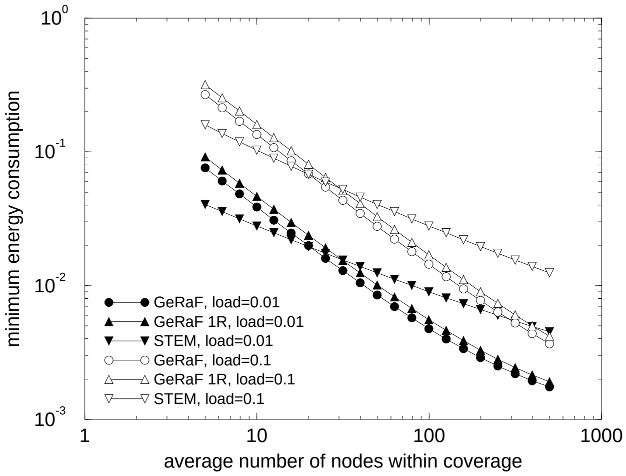

Figure 5: Optimal normalized average energy consumption, ψ, vs. average number of nodes within coverage, N. Network load 0.01 and 0.1.

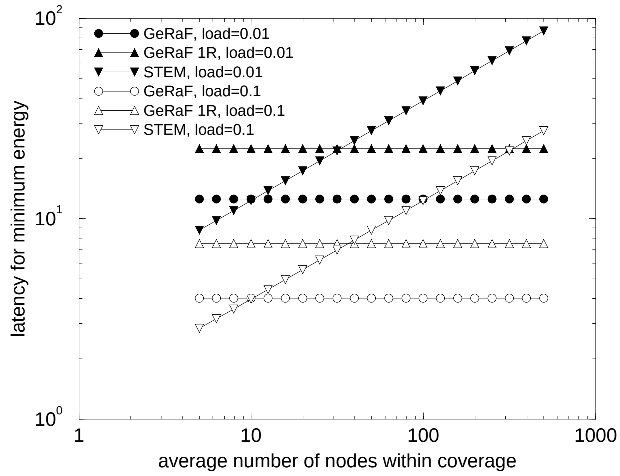

Figure 6: Latency (in units of T_D) corresponding to optimal energy consumption vs. average number of nodes within coverage, N. Network load 0.01 and 0.1.

Finally, the latency which corresponds to the minimum energy point is shown in Figure 6. While this result seems to be very negative for the one-radio scheme, it should be recalled that it corresponds to the very minimum for the energy. Looking back at Figures 3 and 4, one can see that the energy minimum is not very sharp, which means that by allowing a small energy loss we can reduce the latency to acceptable values. Therefore, from results such are those in Figures 3 and 4 we can conclude that in the interesting region of operation the one-radio version of GeRaF works as well as the one with busy tones, while requiring a significantly simpler hardware.

5 Conclusions

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 this idea was described, and an approximate analysis was provided. Unlike previously proposed schemes based on busy tones, our protocol does not need a second radio. The proposed scheme was compared with its two-radio version as well as with STEM, and was shown to perform well 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] Ya Xu, John Heidemann, Deborah Estrin, “Geography-informed energy conservation for Ad Hoc routing,” ACM/IEEE MobiCom2001, 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] M. Zorzi, R.R. Rao, Geographic Random Forwarding (GeRaF) for ad hoc and sensor networks: energy and latency performance, in *IEEE Trans. on Mobile Computing*, vol. 2, n. 4, Oct.-Dec. 2003. Also in Proc. IEEE WCNC’03.

[5] 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.