In the Proceedings of the IEEE International Conference on Communications (ICC), 2004.

dFSK: *Distributed* Frequency Shift Keying Modulation in Dense Sensor Networks

An-swol Hu Sergio D. Servetto

***Abstract*— We consider the problem of reachback communication in wireless sensor networks: multiple sensors are deployed on a field, and they collect local measurements of some random process which then need to be encoded and reproduced at a remote location. In this paper we present the design of a distributed modulation scheme for reachback, dubbed dFSK (for *distributed* Frequency Shift Keying). In dFSK, all nodes in the network first agree on a common stream of bits to send. Then, nodes listen to a few transmissions by other nodes to form an estimate of when to schedule their own transmissions. The aggregate waveform that results from superimposing a large number of these weak transmissions has a pre-specified set of zero-crossings. It is through the location of these zeros that information is conveyed to the far receiver. This mode of operation is analogous to that of standard FSK, where it is through spectral properties of the transmitted symbol (and hence closely related to the rate of its zero-crossings) that information is conveyed to the far receiver.**

I. INTRODUCTION

A. Problem Setup

Consider the following problem setup. Multiple sensors are deployed on a field, and each one of them collects a sample of some stochastic process that unfolds over the field. Then, once done processing their data, all these nodes cooperate to send information back to a far away fusion center where the data is to be analyzed and possibly acted upon. In this paper, we model this scenario as a problem of multiaccess communication where multiple correlated sources to be reproduced at a common receiver are sent over a multiple access channel by an array of cooperative encoders [1]. From a practical point of view this is a most interesting problem since typically each individual sensor/encoder will not have enough resources to generate a strong information bearing signal that can be detected reliably at the far receiver. As a result, without active cooperation among sensors, we see little hope of being able to solve this problem. Thus, our main goal in this paper is to develop a constructive method to allow all nodes in the network to cooperatively generate a strong information bearing signal to communicate with a far receiver.

One possible avenue that we wish to explore in the development of a constructive mechanism for reachback is motivated by some of our recent work on the sensor *broadcast* problem [4]. There, we established the feasibility and constructed algorithms for nodes in a network as in the setup above to cooperate and allow all nodes to obtain a rate-constrained encoding of the entire field of readings picked up by all sensors. Therefore, it is only natural to ask the question of how applications could benefit from this new functionality. Specifically in the context of the sensor reachback problem, a sensor broadcast protocol would allow all nodes to agree on a common stream of bits to send, meaning that all these nodes can then synchronize their transmissions so as to generate a strong signal at the far point, a signal that results from coherently superimposing all the weak signals generated by each of the sensors. This setup is illustrated in Fig 1.

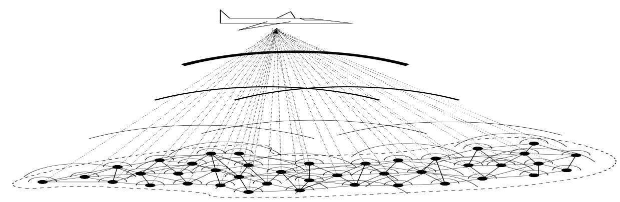

Fig. 1. Cooperation among sensors to reach back to a far receiver. A good analogy to describe the role of a sensor broadcast protocol in the context of reachback is that of a *symphonic orchestra*. When all instruments in the orchestra play independently, all we hear is noise; but when they all play according to a common script, the music from all instruments is combined into a coherent play. The bits distributed during sensor broadcast play the role of that common script for a later cooperative transmission step.

In a sensor networking context, we believe models for multiple access that assume synchronous operation among transmitters fail to capture one of the fundamental difficulties sensor networks must contend with—time synchronization. Without yet getting into details of how the synchronization problem could be solved, let us say that for the model described above, and under the assumption that antipodal signaling is employed, we model the signal $R_N(t)$ at the output of the channel as

$$R_N(t) = bA_N(t) + n(t) = b\left(c_N \sum_{i=1}^{N} p(t-T_i)\right) + n(t),$$

where: $b \in \{-1, 1\}$ is one bit of information shared among all transmitters (obtained, for example, as a result of one execution of a sensor broadcast protocol); c_N is a constant that depends only on the number of nodes N, used to scale transmissions appropriately so as to maintain bounded the total amount of power radiated by the entire network; $[T_1 ... T_N]$ is a vector of random relative offsets; and $n(t)$ is Gaussian noise. Under these assumptions, a solution to the reachback problem consists of a mechanism for the network to generate $R_N(t)$'s from which a far receiver can form reliable estimates of the bits b's.

B. Time Synchronization in Dense Sensor Networks

From the discussion above, it should be clear that the main challenge we are confronted with is that of giving a character-

The authors are with the School of Electrical and Computer Engineering, Cornell University. URL: http://cn.ece.cornell.edu/. Work supported by the National Science Foundation, under awards CCR-0238271 (CAREER), CCR-0330059, and ANI-0325556.

ization of the basic network pulse $A_N(t)$. But that challenge we have already encountered in previous work, where the goal was to achieve network-wide synchronization of all the nodes in the network [3]. Under the exact same assumptions as in the model above, our goal in that work was to define an aggregate signal $A_N(t)$ that could be heard by all nodes in the network simultaneously, and that would contain the information needed to synchronize all nodes. We accomplished that goal by setting up a distributed estimation problem, which gave us random T_i's, and studied the behavior of $A_N(t)$ in the asymptotic regime as $N \to \infty$. Our main result in [3] was to show that, by properly setting up the distributed estimation step, we would be able to obtain a *deterministic* limit aggregate waveform $A_\infty(t)$, with the following properties:

- If the basic pulse p is continuous and bounded, then under some mild technical conditions on the distributions involved, $A_\infty(t)$ is everywhere continuous.[1]
- We have exact characterizations of the points t in which $A_\infty(t)$ takes on positive, negative, and zero values.

In particular, knowledge of the zero-crossing locations $\zeta = \{t \in \mathbb{R} : A_\infty(t) = 0\}$ is what allowed us to solve the synchronization problem. By appropriately setting up the distributed estimation problem, we were able to show that we can make $\zeta = \{nT : n \in \mathbb{Z}\}$, for some separation T between roots. Since all nodes can detect these equispaced zero-crossings of $A_\infty(t)$ at the same time, the synchronization problem can be solved using this information.

This previous work on the time synchronization problem provides the natural motivation for the approach taken in this paper to designing a suitable distributed modulation scheme for the sensor reachback problem. Indeed, if we are able to control, to some extent, the location of the roots of $A_\infty(t)$, it is only natural to think of using that ability not only to synchronize the network, but also to communicate information to a far receiver. Therefore, in this paper we show how to do exactly that: how to modulate data bits onto the time synchronization signal, using it as a "distributed carrier wave".

C. Main Contributions and Organization

The main contribution presented in this paper is the design of a transmitter for cooperative sensor reachback. Cooperation is done in the form of each node individually solving a simple estimation problem for when to transmit pulses over a Gaussian multiple access channel so that the aggregate waveform that results from the transmissions of all nodes has zero-crossings at desired locations. Information is then conveyed to the far receiver by the locations of these zeros.

The rest of this paper is organized as follows. In Section II we describe the basic model for the generation of the data that nodes can observe, based on which they get to form estimates of their delays. Then, in Section III we review and explain a number of results from [3] dealing with the generation of a network-wide time synchronization signal in a distributed manner. dFSK is then presented as a natural extension of that previous work. In Section IV we show how this timing signal can be used as a distributed carrier wave, onto which data bits can be modulated. Final remarks, including a brief discussion on possible receivers, are presented in Section V.

[1]The requirement that p be continuous and bounded we know to be sufficient but not necessary—more general conditions under which the result holds are currently under study.

II. System Model

An essential component of our solution is the set up of a distributed estimation problem, by which each node i gets to estimate its own transmission delay T_i, so as to generate an appropriate $A_\infty(t)$. To solve this estimation problem, the first step is to give a model for the generation of data based on which time estimates will be computed.

In the context of the time synchronization problem studied in [3], the generation of data observed at node i is governed by the following state equations:

$$\begin{aligned} s^{c_1}_{n'+1,1} &= s^{c_1}_{n',1} + 1 \\ t^{c_i}_{n,i} &= \alpha_i(s^{c_1}_{n',1} - \bar{\Delta}_i) + \Psi_i(s^{c_1}_{n',1}), \end{aligned} \tag{1}$$

where:

- $s^{c_k}_{n,i}$ denotes the time of the n-th transition of the operational counter $s_i(t)$, described in the timescale of c_k (the clock at the k-th node). $s_i(t)$ is explained in section III-A.
- $t^{c_k}_{j,i}$ is the time (in terms of the timescale of c_k) at which the i-th node sees the j-th transmission.
- α_i accounts for a constant frequency offset in the two clocks and $\bar{\Delta}_i$ models the fact that the two clocks may have been started at different times.
- $\Psi_i(t)$ is a zero-mean, white Gaussian noise process with variance σ^2.

The first equation describes the hidden state: there is one fixed but arbitrary node in the network, called "node 1", whose clock increments at integer times (according to its own time scale). The second equation relates the observations at node i with the clock transition times of node 1: assuming $n' - n = k \in \mathbb{N}$ (an unknown number of transmissions take place before node i starts hearing anything), then the expression is saying that the pulse seen by node j at $t^{c_j}_{n,j}$ is ocurring at $s^{c_j}_{n+k,1}$. The estimate of each T_i computed at each node i will be a function of a vector of m of these observations $t^{c_k}_{j,i}$.

Please note: there are many assumptions involved in the setup of these state equations, which we are not able to cover in detail in this paper due to lack of space. Those issues have been discussed extensively in [3]—the reader is referred to that work for more details.

III. Asymptotically Optimal Time Synchronization

dFSK is most easily explained as a natural extension to a solution we proposed recently to the problem of achieving network-wide time synchronization [3]. Therefore, before providing details about dFSK in the next section, here we present a summary of the most relevant results from [3].

A. Properties of the Synchronized Network

The clocks c_1 and c_i, for all i will be free running clocks that will have a synchronized "operational" counter built on top of them. We want the operational counter at the ith node to increment at integer values of t (the time of node 1) and hold a value equal to $s_1(t)$, the operational counter of node 1 which increments at integer values of t. The actual synchronization mechanism proposed in [3] to solve this problem setup will not be described here. Instead we will outline the properties of a network that is already synchronized before describing in Section III-B how this synchronization is maintained.

As seen in the state equations (1), it is clear that since there exists a random jitter in the clock of any node i relative to c_1 it is impossible to have node i increment its operational counter $s_i(t)$ at *exactly* the time $s_1(t)$ increments. As a result, in the synchronized network, each node i makes a minimum variance, unbiased estimate of when $s_1(t)$ increments. For every node i, the variance of the estimate does not increase with increasing integer values of t.

Consider a network with N nodes uniformly distributed over the $[0,1]\times[0,1]$ plane. To achieve and maintain synchronization, each node periodically transmits a shifted version of the pulse, $(A/N)p'(t)$,

$$p'(t) = \begin{cases} 1 & -\tau < t < 0 \\ 0 & t = 0 \\ -1 & 0 < t < \tau \end{cases}$$

where the duration of the pulse is 2τ and $\tau << 1$. A is the maximum possible amplitude of the pulse.

The synchronization mechanism described in [3] builds a network where node 1 transmits $(A/N)p'(t-\tau_o)$ and increments $s_1(t)$ at every integer value of $t=\tau_o$ while every node i transmits $(A/N)p'(t-T_i)$, where T_i is a minimum variance unbiased estimate of τ_o written in terms of c_1. To make the time scale changes more understandable, consider c_1 and c_i defined as $c_{1,t}=t$ and $c_{i,t}=\alpha_i(t-\bar{\Delta}_i)+\Psi_i(t)$. If node 1 transmits $(A/N)p'(t-\tau_o)$, then $E(T_i)=\tau_o$ and $E(T_i^{c_i})=\alpha_i(\tau_o-\bar{\Delta}_i)$, where $T_i^{c_i}$ is the random variable expressed in the time scale of node i. Also, from [3] we have that

$$T_i^{c_i} \sim \mathcal{N}\left(\alpha_i(\tau_o-\bar{\Delta}_i), \frac{2\sigma^2(2m+1)}{m(m-1)}\right)$$

where m is the number of observations used to make the estimate. m will be further defined in section III-B. Note that there is a scaling factor of $1/N$ to keep the amplitude of the aggregate signal approximately constant. Without this scaling factor, as we study the limiting characteristics of the aggregate waveform with $N\to\infty$, the signal amplitude will also grow without bound.

The key insight into this setup is that if every node i, for $i=1,\ldots,N$, transmits $(A/N)p(t-T_i)$ then the aggregate waveform observed by the jth node in the network will be deterministic as $N\to\infty$. For simplicity, we choose $\tau >> \max_i \text{Var}(T_i^{c_i})$ so we can approximate the pulse shape as $p(t)$ defined as $p'(t)$ with $\tau=\infty$. We focus on the pulse transmission around $t=\tau_o$ and the aggregate signal received by node j in terms of c_1 thus becomes $r_j^{c_1}(t)=\sum_{n=1}^{N}\frac{A}{N}K_{j,n}p(t-T_n)$, where $K_{j,n}$ is a random variable that gives the fraction of the initial pulse amplitude, sent from any nth node, that is received at node j. This distribution tells node j how likely any possible received amplitude should be. This is important for our study of the aggregate waveform $r_j^{c_1}(t)$ because in order to accurately determine the amplitude of $r_j^{c_1}(t)$ we need to know the amplitudes of the received signals that make up the aggregate signal.

We choose to model path loss using an unknown distribution because we did not know of a model that accurately reflects signal attenuation over the very short distances of interest to our asymptotic analysis. Furthermore, this random model is very general and the distribution can be determined by specific path loss models, such as those proposed by Franceschetti et al. for small cells in urban environments [2]. Our random pathloss model is detailed in the journal version of [3].

In the limit as $N\to\infty$, the law of large numbers tells us that $r_{j,\infty}^{c_1}(t)=\lim_{N\to\infty}r_j^{c_1}(t)=E\big(K_{j,n}p(t-T_n)\big)$. We obtain the following key result in [3].

Lemma 3.1: Under the assumptions presented above, as $N\to\infty$, $r_{j,\infty}^{c_1}(t)$ is continuous with $r_{j,\infty}^{c_1}(t=\tau_o)\to 0$ and $r_{j,\infty}^{c_1}(t\neq\tau_o)\to G(t)$, where $G(t)>0$ for $t<\tau_o$ and $G(t)<0$ for $t>\tau_o$.

Note that from the conditions that $\tau << 1$ and $\tau >> \max_j \text{Var}(T_j^{c_1})$ we naturally get that the synchronization pulses are far apart. That is, the pulse duration is much smaller than the time between two pulses. This setup allows the network to carry out other operations, such as data distribution, between any two synchronization pulses.

B. Maintaining Synchronization

Lemma 3.1 is used in achieving and maintaining synchronization as outlined in [3]. Here, we briefly explain the process of maintaining synchronization before presenting details on dFSK.

We see that node 1's pulse transmission of $(A/N)p(t-\tau_o)$ has a zero-crossing at $t=\tau_o$. By Lemma 3.1, if every node i's estimate of τ_o, T_i, is Gaussian with $E(T_i)=\tau_o$, then the limiting aggregate waveform $r_{j,\infty}^{c_1}(t)$ has a zero-crossing at $t=\tau_o$. Since $r_{j,\infty}^{c_1}(t)$ is valid for all j, any node in the network will see an aggregate waveform that has a zero-crossing at exactly the same time node 1 increments its counter $s_1(t)$. This means that every node will always be able to see when node 1 increments its operational counter.

To maintain synchronization, node i will observe m consecutive zero-crossings and then make a minimum variance unbiased estimate of the next pulse transmission time. The m observations, $\mathbf{Y}=[y_1,\ldots,y_m]$, take on the form

$$\mathbf{Y} = \begin{bmatrix} \alpha_i(s_{n,1}^{c_1}-\bar{\Delta}_i)+\Psi_i(s_{n,1}^{c_1}) \\ \alpha_i(s_{n,1}^{c_1}-\bar{\Delta}_i)+\alpha_i+\Psi_i(s_{n,1}^{c_1}+1) \\ \alpha_i(s_{n,1}^{c_1}-\bar{\Delta}_i)+2\alpha_i+\Psi_i(s_{n,1}^{c_1}+2) \\ \vdots \\ \alpha_i(s_{n,1}^{c_1}-\bar{\Delta}_i)+(m-1)\alpha_i+\Psi_i(s_{n,1}^{c_1}+(m-1)) \end{bmatrix}$$

for some integer n. As a result, for any m consecutive observations, we can simplify notation by using the model $\mathbf{Y} = \mathbf{H}\theta + \mathbf{W}$, where

$$\theta = \left[\begin{array}{c} \theta_1 \\ \theta_2 \end{array} \right] = \left[\begin{array}{c} \alpha_i(s^{c_1}_{n'',1} - \bar{\Delta}_i) \\ \alpha_i \end{array} \right]$$

with

$$\mathbf{H} = \left[\begin{array}{ccccc} 1 & 1 & 1 & \dots & 1 \\ 0 & 1 & 2 & \dots & m-1 \end{array} \right]^T$$

and $\mathbf{W} = [w_1 \dots w_m]^T$. Since $\Psi_i(t)$ is a white Gaussian noise process, $\mathbf{W} \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = \sigma^2 \mathbf{I}$. Note that the observations are corrupted only by the jitter in node i's own clock since the limiting waveform seen by node i, $r^{c_1}_{i,\infty}(t)$, has a zero-crossing at the exact time node 1 increments $s_1(t)$. Finally, based on these observations, node i will compute an optimal estimate $\hat{y}_{m+1}$ of y_{m+1}, and transmit at that time. It is shown in [3] that for any node i, $\hat{y}_{m+1}$ is a Gaussian random variable with $E(\hat{y}_{m+1}) = \alpha_i(s^{c_1}_{n,1} - \bar{\Delta}_i) + m\alpha_i$. Since the mean, in terms of c_1, is the time $s_1(t)$ will next increment, when all the nodes transmit at their estimated time, the aggregate waveform seen at any node j will be $r^{c_1}_{j,\infty}(t)$. The nodes then take their most recent m observations and the cycle repeats.

IV. Distributed FSK Modulation

The time synchronization mechanism described in Section III forms the core on top of which dFSK is built. In this section, we first show how an aggregate waveform suitable now for both synchronization *and* communication is generated, to then show how bits are modulated onto this new waveform.

A. Waveform Generation

We observe that synchronization is achieved and maintained based solely on every node i's ability to observe a zero-crossing that occurs at the exact time $s_1(t)$ increments. It is possible to retain this property while generating an aggregate waveform that is suitable for reachback communication.

Consider a network that has already been synchronized and is simply maintaining synchronization as described in Section III-B. We know that the operational counter at every node i, $s_i(t)$, is counting at about the same time and has the same value. The nodes have agreed on a common stream of bits to transmit and they also know to start communication when their operational counters increment to a certain value, L. Each node i will change its transmission scheme when $s_i(t) = L$. Immediately after $s_i(t)$ increments to L, node i will send a signal of constant amplitude A/N. It will maintain this signal until its next estimated zero-crossing time where it will immediately start transmitting a signal of amplitude $-A/N$. To give a more specific example, we view this process in the time scale of c_1 for the zero-crossing times $t = \tau_o$ and $t = \tau_o + 1$. Node 1 starts transmitting a constant signal of A/N immediately after $t = \tau_o - 1$ and does so until $t = \tau_o$ at which it will start transmitting $-A/N$. Node i will be transmitting A/N until T_i where it will start transmitting $-A/N$. Recall that $E(T_i) = \tau_o$. At $t = \tau_o + 1$, node 1 will transition its transmission from an amplitude of $-A/N$ to A/N. For node i, it will start transmitting a signal of amplitude A/N at its next estimated transition time T_i where now $E(T_i) = \tau_o + 1$. If this is followed by every ith node in the network for $i = 1, \dots, N$, then we see that the result of Lemma 3.1 still holds. Furthermore, since the mechanism for maintaining synchronization relies solely on the zero-crossings, the new aggregate waveform generated by such a scheme will still allow synchronization to be maintained as before. Fig. 2 illustrates a simple example.

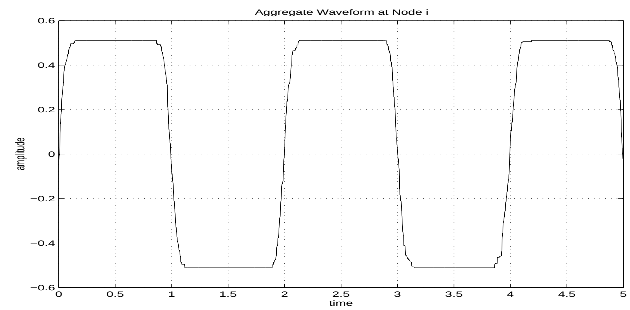

Fig. 2. To illustrate the aggregate waveform with $N = 200$ nodes seen at node i. In this example $A = 1$ and we assume that the jitter variance of every node is the same in the time scale of c_1 with standard deviation 0.05. For illustration purposes, we take $K_{i,n}$ to be exponentially distributed with $\lambda = 2$. Notice that even with only 200 nodes the zero-crossings occur almost exactly in the correct place.

From Fig. 2, we see an interval between two transitions where the signal is effectively flat with a magnitude of $E(K_{i,n})A$. This flat interval comes from the assumption explained in Section III-A that the time between two synchronization pulses, the *symbol time*, is long relative to the synchronization pulse duration. In the type of waveform displayed in Fig. 2, however, instead of the synchronization pulse duration we will consider the time it takes the signal to make a transition. We call the transitions that occur at the times where synchronization pulses would have occurred the *primary transitions*. From the figure we see that by putting extra transitions between the primary transitions, it is possible to modulate information onto the aggregate waveform. Thus, we need to characterize the time it takes for a transition since it will determine how many extra transitions can be placed between any two primary transitions.

We define a transition region around an integer value of $t = \tau_o$ as the interval $t \in [\tau_o - \tau^{\tau_o}_{trans}, \tau_o + \tau^{\tau_o}_{trans}]$ where $||r^{c_1}_{j,\infty}(t)| - A| > \epsilon$ with $A > \epsilon > 0$. The transition region is important since for two consecutive transitions, we will not let the transition regions overlap. The choice of ϵ thus allows for a trade-off between the number of extra transitions placed in a symbol and the amplitude of the resulting symbol. By choosing ϵ large, we can shrink the transition region thus allowing more transitions between two primary transitions. However, by declaring a smaller transition region means that it is possible for the aggregate waveform to begin the next transition before the current transition is completed. The result would be an aggregate waveform with a smaller magnitude. As well, given a jitter distribution that is not compactly supported, it is true that

mathematically $|r_{j,\infty}^{c_1}(t)|$ may only approach the value of A in the limit. However, for all practical purposes, ϵ can be chosen so that outside the transition region $|r_{j,\infty}^{c_1}(t)| \approx A$.

As expected, the number of extra zero-crossings we can put between $t = \tau_o$ and $t = \tau_o + 1$ depends on the size of the transition region around each zero-crossing. The size of a transition region depends on the variance of the transition time estimate of each node i. For example, the value of $\tau_{trans}^{\tau_o+1}$ will depend on Var(T_i) for $i = 2, \ldots, N$. For a given ϵ, the larger their variances are, the larger $\tau_{trans}^{\tau_o+1}$ will be. As a reminder, please recall that when we specify $t = \tau_o$ or $t = \tau_o + 1$, we are referring to any two consecutive integer values of t.

For a node i, the distribution of T_i for the zero-crossing at $t = \tau_o$ and $t = \tau_o + 1$ was found in [3], and a simple extension of that derivation yields the distribution of any estimate y_{m+s} with $0 < s \leq 1$, for a zero-crossing at $t = \tau_o + s$. These (tedious but relatively straightforward) calculations are omitted due to lack of space. What is important to mention though is that, not surprisingly, we find that the variance of any estimate y_{m+s} in $0 < s \leq 1$ increases monotonically with s. As a result, we can upper bound the number of zero-crossings that can be placed between $t = \tau_o$ and $t = \tau_o + 1$ by looking at the width of a transition at $\tau_{trans}^{\tau_o+1}$, or equivalently, $\tau_{trans}^{\tau_o}$ (since the variance of the estimate at each primary transition is the same). Specifically, the maximum number of zero-crossings, R', that can be placed in $t = \tau_o$ and $t = \tau_o + 1$ is $2(R'+1)\tau_{trans}^{\tau_o+1} \leq 1$. For implementation purposes however, we would like the waveform to be smooth at symbol boundaries. As a result, we choose to have the waveform always transition from $-A$ to A at the primary transitions. Because of this requirement, we must have the number of zero-crossings, R, between $t = \tau_o$ and $t = \tau_o + 1$ take on the form $R = 2q + 1$ where q is a non-negative integer. As a result, for our communication system, the maximum number of zero-crossings that can fit inside a symbol will be $R = 2q + 1 < R'$. This can be seen in the waveforms of Fig. 3.

B. Modulation Scheme

To study exactly how this modulation scheme would work, we focus on one interval between $t = \tau_o$ and $t = \tau_o + 1$. We consider the case where this reachback communication system will do M-ary signalling. This means that $q \in \{0, 1, \ldots, M-1\}$ and $R = 2q+1 \in \{1, 3, \ldots, 2M-1\}$. Symbol S_{q_o} is a symbol waveform with $2q_o + 1$ zero-crossings between $t = \tau_o$ and $t = \tau_o + 1$. Fig. 3 illustrates an example of a waveform modulated in this manner.

An important point to note is that each node i looks for a zero-crossing only in a small interval around its estimate of where the primary zero-crossing should be. As a result, when other zero-crossings are placed between $t = \tau_o + k$ and $t = \tau_o + k + 1$, node i still only observes the zero-crossings at $t = \tau_o + k$ and $t = \tau_o + k + 1$. Thus, the same synchronization properties are still maintained using the zero-crossings at $t = \tau_o + k$ for $\tau_o, k \in Z$ while the other zero-crossings are used for communicating with the far receiver.

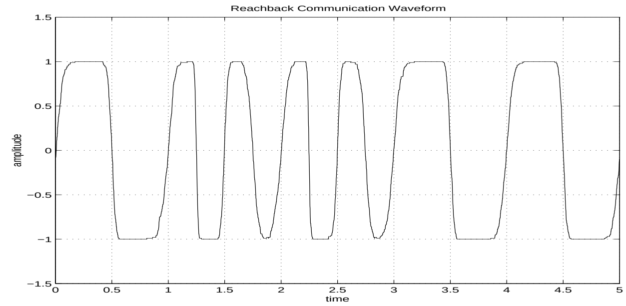

Fig. 3. The above figure illustrates 5 symbol periods with $R = 1, 3, 3, 1, 1$, respectively. Such a waveform could be used to send the bit stream 01100. We see that the signal is always increasing at integer values of t. This ensures smoothness in the limit waveform.

V. CONCLUSIONS

In this paper we presented dFSK, a distributed modulation method suitable for the sensor reachback problem. Transmission proceeds in two stages. In a first stage, sensor nodes cooperate to solve a broadcast problem, by which all nodes agree on a common stream of bits to upload. In a second stage, sensor nodes cooperate to generate an aggregate waveform providing an encoding of a stream of bits that in turn encodes the field of sensor readings. The distributed modulation technique employed closely resembles the mechanics of the classical FSK technique (if $A_\infty(t)$ was a pure sine wave, the rate at which its zero-crossings occur would uniquely determine its frequency), hence the title of this paper.

Current work is focusing on the study of performance of different receivers for this particular transmitter. Possible *coherent* receivers include a correlator against A_∞, and a binary hypothesis test for a set of observations consisting of the zeros of A_∞ immersed in noise. A possible *non-coherent* detector would be the standard energy detector for FSK. Some interesting preliminary results we have, both in terms of analysis and simulations, were not included in this paper due to space constraints. Therefore, this topic will be dealt with elsewhere.

Acknowledgement. We would like to thank Brian Sadler, for pointing out to us the connection between the idea of communicating using the roots of $A_\infty(t)$ and the classical FSK modulation method.

REFERENCES

[1] J. Barros and S. D. Servetto. The Sensor Reachback Problem. Submitted to the IEEE Transactions on Information Theory, November 2003. Available from `http://cn.ece.cornell.edu/`.

[2] M. Franceschetti, J. Bruck, and L. J. Schulman. A Random Walk Model for Wave Propagation. To appear, IEEE Trans. on Antennas and Propagation. Available from `http://www.paradise.caltech.edu/~massimo/`.

[3] A. Hu and S. D. Servetto. Asymptotically Optimal Time Synchronization in Dense Sensor Networks. In *Proc. 2nd ACM Workshop on Sensor Networks and Applications (WSNA)*, San Diego, CA, 2003. Held in conjunction with ACM MobiCom. Note: An extended *journal* version titled ”The Time Synchronization Problem in Large-Scale Sensor Networks” has been submitted for publication and is available at `http://cn.ece.cornell.edu/`.

[4] S. D. Servetto. Distributed Signal Processing Algorithms for the Sensor Broadcast Problem. In *Proc. 37th Annual Conf. Inform. Sciences Syst. (CISS)*, Baltimore, MD, 2003.