Challenges: Communication through Silence in Wireless Sensor Networks *

Yujie Zhu and Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
{zhuyujie, siva}@ece.gatech.edu

ABSTRACT

Wireless sensor networks (WSNs) are typically characterized by a limited energy supply at sensor nodes. Hence, energy efficiency is an important issue in the system design and operation of WSNs. In this paper, we introduce a novel communication paradigm that enables energy-efficient information delivery in wireless sensor networks. Compared with traditional communication strategies, the proposed scheme explores a new dimension - *time*, to deliver information efficiently. We refer to the strategy as *Communication through Silence* (CtS). We identify a key drawback of CtS - *energy - throughput trade-off*, and explore optimization mechanisms that can alleviate the trade-off. We then present several challenges that need to be overcome, primarily at the medium access control layer of the network protocol stack, in order to realize CtS effectively.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Algorithms, Design, Performance

Keywords

Wireless sensor networks, Power conservation, MAC, Communication model, Communication through Silence, Communication Strategy

1. INTRODUCTION

Most, if not all, forms of communication in wireless sensor networks (WSNs) are typically assumed to use a common strategy for communication that we refer to as *Energy based Transmissions* (EbT). Essentially, the information transfer between any two sensors happens solely using energy based transmissions. For example, when a sensor s_1 wants to communicate a *value* 97 to a neighboring sensor s_2, it sends the sequence of bits (1, 1, 0, 0, 0, 0, 1) in succession using energy for every bit transmitted. Thus, if the energy consumed per bit transmitted is e_b, the total energy consumption is $7 * e_b$.

In this paper, we explore a new communication strategy that is based on conveying information using silent periods as opposed to energy based transmissions. In the above example, s_1 would send a *start* signal to s_2, which would then start *counting* up from zero. s_1 knowing the rate at which s_2 is counting, would send a *stop* signal when it knows s_2 would have counted up to the value 97. When s_2 receives the stop signal, it stops counting and treats the value in the counter as the information transmitted by s_1. If the start and stop signals can be sent with the same or lesser energy than e_b, the total energy consumed is at most $2 * e_b$, which is better by a factor of over 3 when compared to the EbT scheme. We refer to such an approach that uses *silence* to communicate as *Communication through Silence* (CtS).

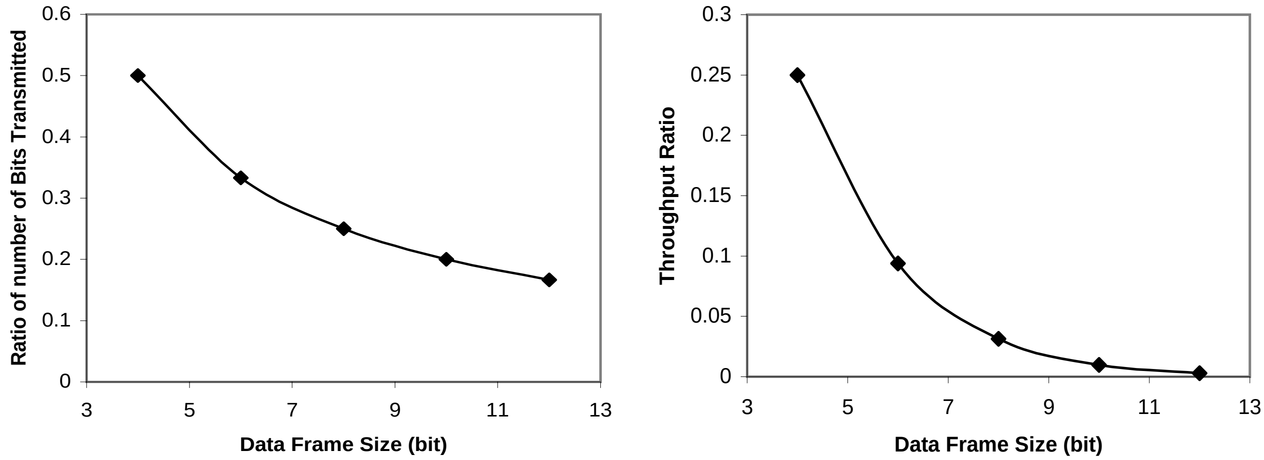
While it is evident that the CtS Strategy can deliver considerable amounts of energy improvement, there is a key trade-off involved in the form of throughput reduction. Considering the same example as before, while the value 97 was transmitted in 7 bit slots when using EbT, the same value will take 97 clock cycles when using CtS. If the clock rate of the sensor is of the same order as the data rate of its radio, this translates into an exponential decrease in the throughput enjoyed by the sensor. Secondly, even if the throughput problem is overlooked (say, for delay insensitive applications) or addressed through appropriate optimization mechanisms, there still remain several challenges related to how traditional medium access control (MAC) functionalities such as framing, addressing, sequencing, error-control, and contention resolution can be performed when using a strategy such as CtS.

In this context, we make three contributions in this paper:

- We introduce the new paradigm of *Communication through Silence* for wireless sensor networks, define the concept of CtS, and identify its basic trade-offs.
- We present unique optimization strategies that can be employed when using CtS that either alleviate the serious throughput trade-off, or in some cases even improve upon the throughput performance when compared to the traditional EbT scheme.
- We identify and discuss several research challenges that exist in realizing both the optimization strategies and traditional MAC layer functionalities.

*This work was funded in part by the National Science Foundation under grants ECS-0225497, ECS-0428329, ANI-0117840, and CCR-0313005.

MobiCom'05, August 28–September 2, 2005, Cologne, Germany.



(a) Ratio of bits transmitted

(b) Throughput Ratio

Figure 1: **The Energy-Throughput Tradeoff in Basic CtS (CtS:EbT)**

The rest of the paper is organized as follows: Section 2 introduces CtS, and identifies the key challenges to be addressed to practically realize CtS. Section 3 describes several optimization strategies that can be employed with CtS in order to improve its throughput performance while retaining its energy benefits. Section 4 identifies several challenges in realizing a MAC protocol for the CtS approach. Section 5 discusses related work, and finally Section 6 summarizes the paper.

2. OVERVIEW AND CHALLENGES

The conventional communication strategy - EbT - involves the use of energy driven bit transmissions for sending information from a sender to a receiver. Generally, if the information to be sent by the sender at given instance spans k bits, the energy consumption is $k * e_b$. Note that we ignore issues of headers, synchronization, etc., for clarity. In this context, we introduce a new communication strategy that uses *silent periods* to convey information from the sender to the receiver. We refer to such a strategy as *communication through silence* (CtS). The start and stop signals in the example presented in Section 1 can be assumed to be one-bit transmissions The energy consumption for CtS is then always $2 * e_b$, irrespective of the value being sent. Note that the energy consumption for counting per clock cycle is considerably smaller than the energy consumption for transmission, and more importantly the counting clock (or the system clock), which has to be active even for the conventional communication, can be tapped into, thus not incurring any additional overheads. Not surprisingly, the above performance improvements will grow as the magnitude of the value being transmitted increases. For example, if the value being conveyed spans 20 bits, the savings grow to a factor of 10 or about 900%. Essentially, with increase in the magnitude of information being conveyed, the energy consumption of CtS stays at $2 * e_b$, while that of EbT increases.

The trade-off, however, lies in the delay taken to convey any piece of information. Assuming that the clock rate is the same as the data rate of the underlying communication channel, the CtS strategy, as described earlier, incurs exponential delay increase when compared to EbT. Thus, *the throughput of the basic CtS strategy is substantially lower than that of EbT.* Figure 1(a) and (b) shows the energy (number of bits transmitted) and throughput ratio between CtS and EbT scheme. Since each frame takes only two signals (start and stop) to transmit, the energy consumed for transmit each bit of data decreases inverse linearly with frame size. However, the throughput of transmitting each frame decreases exponentially with the number of bits in the frame, as oppose to EbT, which always take 1 bit slot to transmit 1 bit. Essentially, the throughput of CtS decreases as $\frac{s}{2^s}$, where s is the frame size.

While we discuss later in the paper how the performance of the basic CtS strategy can be augmented to not only alleviate the decrease in throughput, but in many cases go beyond that of EbT while preserving the same benefits in energy consumption, we now provide a formal definition of the *basic* CtS strategy:

To deliver a binary packet of size k and form $n_{k-1} \ldots n_1 n_0$ (where $n_i = 0,1$ for all $i = 1,2 \ldots k-1$), sender interprets the bit stream as a value N as follows:

$$N = 2^{k-1} \times n_{k-1} + 2^{k-2} \times n_{k-2} + \ldots + 2 \times n_1 + n_0 \qquad (1)$$

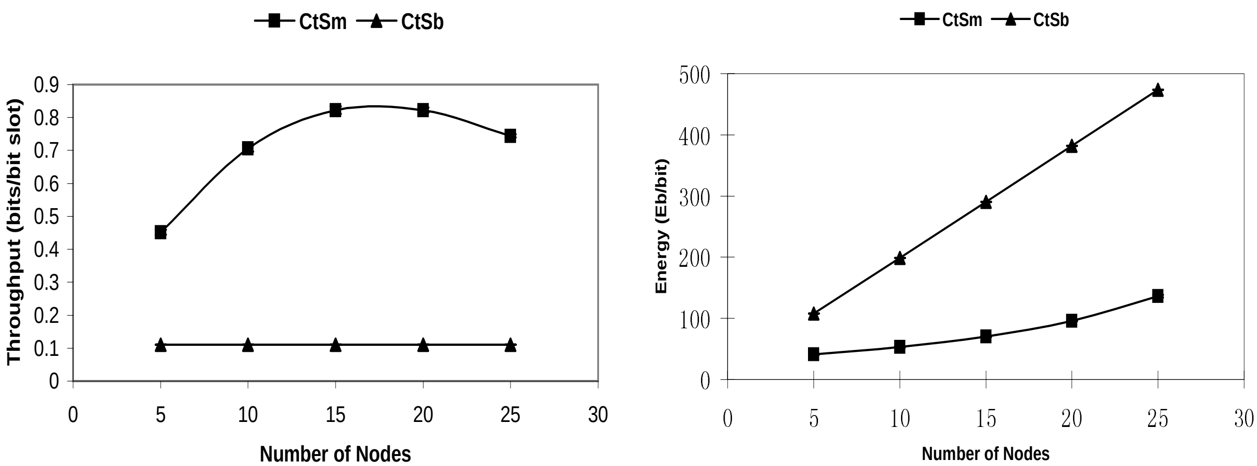
and transmits N using only two signals: a start signal and a stop signal, with the time between the two signals being the time taken by the receiver to count up to the value N from zero. The receiver, knowing N and k (which is the standard frame length in bits), infers the bit stream $n_{k-1} \ldots n_1 n_0$.

2.1 Goals

The contributions of this paper are around two key high level challenges that need to be addressed in order to be able to use CtS as a practical strategy in WSNs:

- *Energy-throughput trade-off:* In the basic CtS strategy described earlier, for an improvement of the order of α in energy consumption, the throughput reduction incurred is by a factor of 2^{α}. While it is true that many WSN applications might trade-off throughput for energy performance improvement, the scope of applicability of CtS in WSNs is likely to increase tremendously if the energy-throughput trade-off can be alleviated or even improved. Hence, we explore the following question: *Can the throughput performance of CtS be improved using additional techniques such that the throughput reduction is lesser or absent for the same amount of energy improvement?*

- *Protocol Realization :* The energy-throughput trade-off is a fundamental theoretical and potentially an algorithmic problem to solve. However, another important challenge to be



(a) Throughput Performance

(b) Energy Performance

Figure 2: **Impact of Multiplexing**

addressed is how CtS is practically realized in a WSN setting. In this context, we explore the question: *What are the challenges associated with the realization of the CtS strategy, specifically with respect to functionalities such as framing, addressing, sequencing, error control, and contention resolution?*

3. OPTIMIZATION STRATEGIES

In this section, we explore communication strategies that are possible due to the unique characteristics of CtS, and can improve upon the considerably low throughput performance of basic CtS. For the rest of the discussion in this section, we make the following assumptions: (i) the start and stop signals are uniquely addressable and occupy a limited number of bits, but for simplicity we will assume one bit transmission slot each for the time being, (ii) the communication channel is lossless, (iii) the sender and the receiver clocks are perfectly synchronized, and there are no counting errors, (iv) the route from a sensor to the sink is made available by an independent routing protocol, and (v) the sequence of CtS frames (with values) to be transmitted are available in the buffer (i.e. the framing is already performed). We make the above assumptions to simplify the discussion of the strategies. We refer to the segment of information CtS sends using a pair of start-stop signals as a *CtS frame*, and its content as a *value*.

All results presented are derived from a simple custom built event-driven simulator that allows for packet transmissions and receptions in pre-configured sensor network topologies. The simulator does not incorporate elements such as channel losses, jitter, etc. However, collisions and retransmissions are recorded, and appropriately accounted for in the performance evaluations. We discuss the specific network topologies used, and any other assumptions made as we present the results. The energy results are based on a model that incorporates transmit, receive, idle, and sleep energy values of 1100mw, 900mw, 100mw, and 20mw respectively based on [1]. The measure for energy consumed is energy/bit, and the measure for throughput is defined as bits/bit slot, which indicates how many information bits are actually transmitted in one transmission clock cycle. 95% confidence intervals are presented for all data points.

3.1 Multiplexing

Consider two contending links L_{12} and L_{34} between sensors s_1 and s_2, and s_3 and s_4 respectively. In EbT, when L_{12} has an ongoing transmission, L_{34} has to remain idle. For example, if s_1 has to send information, say the value 8, to s_2, while s_3 also has to send information, say the value 13, to s_4, L_{12} will have to first complete its transmission of the value 8, before L_{34} can perform its transmission of the value 13. Hence, information transmission has to be inherently sequentialized as long as the links in question are contending links.

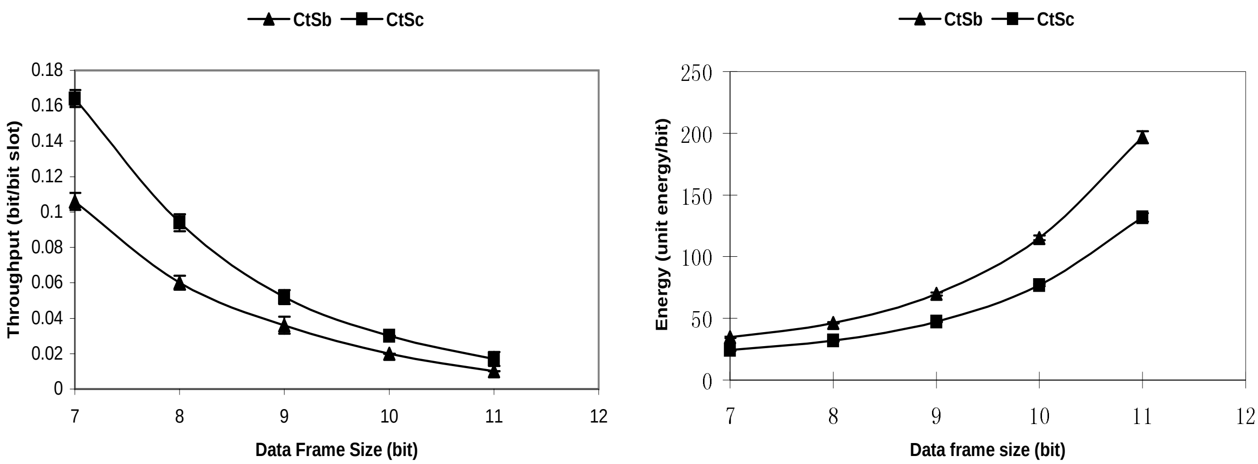
However, when using CtS, even if the links are contending, such information transfer can be performed in *parallel*, by multiplexing the start and stop signals appropriately. Using the same example as above, if s_1 sends the start signal in bit time slot t_i, the stop signal has to be sent in bit time slot t_{i+8}. However, s_3 can send its start signal in bit time slot t_{i+1}, which in turn will require it to send its stop signal in slot t_{i+14}. Thus, while in the basic CtS approach, the two transmissions would have taken a total of 21 slots, the multiplexing has allowed the two transmissions to be completed in 14 time slots.

We refer to the ability to send multiple overlapping CtS frames simultaneously as *multiplexing*, and CtS with multiplexing as CtS_m in the rest of the paper. Note that such multiplexing can be done irrespective of whether the contending links in question share vertices or not. Multiplexing in CtS can thus be formally defined as follows:

Multiplexing: *If a link L has a scheduled transmission of a CtS frame with start and stop signals in bit time slots t_i and t_{i+k}, any other CtS frame on a contending link can be scheduled as long as the start and stop signals of the new frame are not transmitted in slots t_i and t_{i+k}.*

How the multiplexing is controlled by the sensors in a distributed fashion is an important problem. However, we defer discussion about the problem till Section 4, and assume a simple ALOHA like scheme for the rest of the section. Essentially, a sensor transmits its signals independent of the other sensors, and multiplexing occurs naturally. Figures 2(a) and (b) show the simulation results of the throughput and energy consumption performance of the basic CtS approach (CtS_b) and CtS with multiplexing (CtS_m) using the simple ALOHA like scheme respectively. The results are shown for varying number of contending sensors in a *one-hop region*. In this configuration, the transmission of each sensor is capable of reaching all the other sensors in the region. Note that multiplexing is essentially a strategy for communication over a single link.

Figure 2(a) shows the throughput performance, and it can be ob-



(a) Throughput Performance (b) Energy Performance

Figure 3: Impact of Cascading

served that CtS_m achieves about a 4*x* improvement in throughput when compared to CtS_b. The primary reason for the improvement is the ability to multiplex multiple CtS frames over a given time window. The throughput performance of CtS starts to decrease when the number of nodes in the contention region is more than 15. This is because of the increase in the collisions and retransmissions of start/stop signals. In practice, this problem can be alleviated by adapting the CtS data frame size to reduce contention. However, in simulations we keep the same frame size for consistency.

The energy performance in Figure 2(b) is presented in terms of the average unit energy consumed per information bit transferred. It can be observed that the energy performance of CtS_m is better when compared to CtS_b, and the difference increases as the total number of nodes in the contention region increases. This is due to the less time nodes spend in the idle state: when multiplexing of CtS frames is possible, the network throughput increases, hence the total amount of time used to transmit all the CtS frames is less, and the proportion of time nodes spend in the idle state is less. Although in CtS_m, more collisions and retransmissions are possible, which can cause each node to transmit more start/stop signals than in the case of CtS_b, the resulting extra energy consumption is compensated by the previous factor. This shows that the multiplexing strategy can not only improve the throughput performance of CtS_b, but also save on energy.

3.2 Cascading

Consider sensor s_1 having to send CtS frames with values of (7, 13, 19, 28, 10, 6, 8, 21) to a neighboring sensor s_2. In CtS_b, the above information can be sent in a total of 112 bit time slots (sum of the values). Assuming 5 bit EbT data frames, this translates into a throughput utilization of 35.71% ($\frac{40}{112}$). However, assume that CtS is augmented with an *intermediate* signal in addition to the basic start and stop signals. When a receiver receives an intermediate signal, it records the value it has currently counted up to, but *continues* counting for the next value instead of resetting the counter to zero. In the above scenario, s_1 can then send a start signal at time slot t_i, intermediate signals at time slots t_{i+7}, t_{i+13}, and t_{i+19}, and a stop signal at time slot t_{i+28}. Note that it has to "stop" at 28 since the next value to be transferred - 10 - is less than 28. Thus, four values can be transferred in just 28 bit time slots for an effective throughput of 71.43% (the overall throughput for the entire sequence using the same methodology is 67.8%). Furthermore, the number of signals transmitted is now reduced to 5 from the original 8 (4 start/stop signals) required in CtS_b.

We refer to this ability to build on previous transmissions to send subsequent transmissions as *cascading*. We refer to CtS with cascading as CtS_c. Again, if the start/stop/intermediate signals are uniquely addressable, the cascading can be done not just between the same pair of sensors, but also between different pairs of sensors with the intermediate signals carrying the address of the specific sensor that value is meant for. Also, despite the fact that the type of signals has now increased to three, a single bit time slot is still sufficient to transfer any of the three signals when using a pulse based signaling approach as described in Section 4.

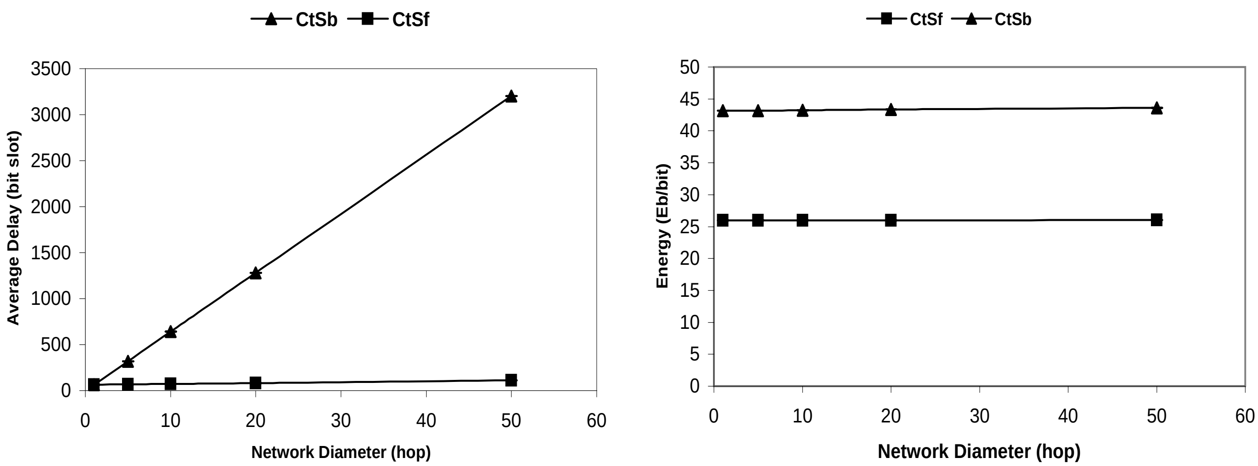
Cascading can thus be defined as:

Cascading: *If a sensor has values* $v_1, v_2, ..., v_k$ *to send to a neighboring sensor, it can send a single start signal, multiple intermediate signals corresponding to the values* $v_1, v_2, ..., v_i$*, and a stop signal corresponding to value* v_{i+1}*, where i is the minimum subscript such that* $v_{i+2} < v_{i+1}$.

Figures 3(a) and (b) present results for the throughput and energy consumption results for CtS_b and CtS_c respectively in a simple two node topology where a transmitter is sending information to a receiver. The values of the frames that the transmitter has to send to the receiver is generated randomly using uniform distribution. It can be seen that the throughput results for CtS_c increase by about 50% when compared to CtS_b. This increase is due to the cascading performed. Interestingly, the energy results are also better for CtS_c. This is because all but one value conveyed in a "cascade" is conveyed using effectively only one signal. Hence, the average number of signals per value decreases to a value less than 2. Also notice that the throughput performance of both CtS_b and CtS_c degrade with data frame size, while energy performance of both CtS_b and CtS_c improve with data frame size. This is due to the longer CtS delay frames when the data frame size increases.

3.3 Fast-forwarding

Consider a sensor s_m sending a value 17 back to the sink along a path that includes other sensors s_{j1}, s_{j2}, and s_{j3} respectively. In both EbT and CtS_b, the value is conveyed first to s_{j1} completely, which then completely transfers the value to s_{j2}, and so on. Thus, considering CtS_b, and assuming that there are no other transmissions contending for the channel, the total time taken to transfer the value back to the sink is $4 * 17$, which is 68 bit time slots (when us-



(a) Throughput Performance

(b) Energy Performance

Figure 4: **Impact of Fast Forwarding**

ing EbT, it would have taken 20 bit time slots assuming 5-bit data frames).

However, note that the sensor s_{j1} does not really need to wait for receiving the complete value when using CtS. Essentially, when s_{j1} receives the start signal at time slot t_i, it can go ahead and send its own start signal to sensor s_{j2} at time t_{i+1}. Similarly, when sensors s_{j2} and s_{j3} receive the start signals at time t_{i+1} and t_{i+2} respectively, they can go ahead and send their start signals in the next bit slot. Then, when sensor s_m eventually transmits the stop signal at t_{i+17}, sensor s_{j1} can immediately forward the stop signal at t_{i+18}. The sink will thus receive the stop signal at time slot t_{i+20}. Thus the same amount of information is now transferred in 20 bit time slots as opposed to 68 bit time slots.

More generally, for a h hop path from a sensor to the sink, and a value v that will take time $h*v$ to deliver back to the sink, such a forwarding technique will take time $v+h-1$ to deliver the same information. We refer to such a forwarding technique as *fast-forwarding*, and CtS with fast forwarding as CtS_f. Also, it is straightforward to observe that such physical overlapping of information transmission on subsequent contending hops is not feasible when EbT is used. Fast-forwarding can be formally defined as:

Fast-forwarding: *When a sensor sends information v back to the sink, sensors on the the intermediate hops can relay their respective start (and stop) signals on the slot next to the one they receive it on from a downstream sensor.*

Figures 4(a) and (b) show the performance results for CtS_f when compared to CtS_b in a scenario consisting of a linear chain of sensors where information is being sent from one end to the other (say the sink). Packets are transmitted hop-by-hop from the sender to the receiver. Using the CtS_b strategy, a relay node only transmits a packet to the next hop after receiving the entire packet (both start and stop signals). Using the CtS_f strategy, a node doesn't have to wait until receiving the entire packet to transmit the start signal, and hence, the end-to-end delay of delivering a data packet is much lower for CtS_f. It can be observed that the average end-to-end delay of transmitting one frame of CtS_b increases linearly with path length, which is caused by the "wait and forward" strategy CtS_b uses. While the delay of CtS_f increases much slower with path length, since the end-to-end delay is dominated by the delay between start and stop signals, not the delay of relaying signals through multi-hops. The energy consumption of CtS_b is also worse since the total amount of time transmitting all the frames end-to-end is higher, hence each node spends more time being idle. Finally, the average energy per node for transmitting each bit remains the same with different path lengths as the same number of signals are transmitted on every hop.

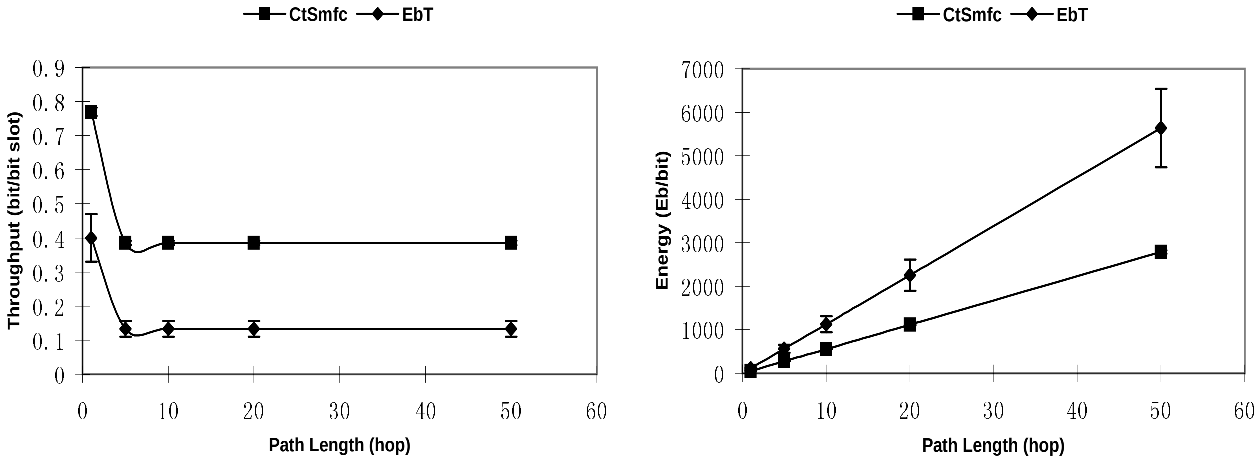
3.4 Integrated Operations

The three strategies outlined thus far are orthogonal in nature in terms of the decision process, and hence can be easily overlayed on each other. The only drawback of using the multiple techniques together is the potential increasing of the collision probability due to higher throughput. However, this can be handled just like an increased load scenario through appropriate adaptation of the frame length. To illustrate the benefit of combining all the proposed strategies, we provide simulation results that compare the energy and throughput performance of CtS with combined optimization strategies (referred to as CtS_{mfc}) and EbT in a multi-hop network environment.

For the simulation results presented here, we assume a sensor network with radius h hops with the sink located at the center of the network. Each node in the network has approximately the same number of neighbors (average degree = 10). For each simulation scenario, a sensor node is randomly chosen in the network as a source which sends data through a path of hop length h using CtS_{mfc} strategy. We calculate the end-to-end throughput and energy consumption of CtS_{mfc} with varying network radii, and compare those results with those of a EbT.

When the path length is relatively low, the CtS_{mfc} throughput in figure 5 is about twice that of EbT, due to the reasons discussed earlier in the section. And as the network radius increases, this ratio remains the same. This is because for both schemes, the end-to-end throughput decreases linearly with the number of hops in a path. For EbT, the reason is obvious, while for CtS_{mfc}, since per-hop throughput is close to 1 bit/slot, and fast forwarding is used to deliver each signal without any delay at each hop, the throughput is inversely proportional to the number of hops of the end-to-end path. Therefore, the improved throughput performance of CtS in a single hop scenario is extended to the multihop case due to the employment of fast forwarding.

As to the energy performance, for both CtS_{mfc} and EbT, energy consumption increases linearly with number of hops. This is expected because each hop requires around the same amount of transmission, receiving and idle energy for the given network scenario.



(a) Throughput Performance

(b) Energy Performance

Figure 5: Energy and Throughput Performance Comparison beteen EbT and CtS_{mfc} scheme, Average Degree = 10

However, the energy consumption of EbT scheme increases faster than CtS_{mfc} as hop count increases, since EbT consumes higher energy for delivering the same amount of data at each hop.

While the results presented in the section thus far are merely illustrative of the performance improvements achievable using CtS, they do motivate that CtS is a promising communication strategy worth further study.

4. CHALLENGES

In the previous sections, we have introduced the CtS paradigm for communication, while establishing that CtS can have considerable benefits both in throughput and energy consumption. However, the CtS mode of communication is significantly different from conventional energy based transmissions, and hence will require appropriate protocols tailored to the paradigm from the physical layer to several of the higher layers. In the rest of the section, we present the challenges that need to be addressed in perhaps the most important layer of the protocol stack with respect to CtS - the medium access control (MAC) layer. We present the challenges in terms of distinct functionalities that need to be supported by the MAC layer. We also briefly outline what we believe will be the PHY layer requirements to facilitate the MAC protocol, and provide a short discussion on the impact of the CtS strategy on other higher layers of the protocol stack. The goal of this section is thus to expose the research challenges that need to be investigated in order to realize CtS. In the rest of the section, we refer to a MAC protocol supporting the CtS strategy as simply CtS-MAC.

4.1 Radio Requirement Basics

Recall that a typical CtS frame consists of up to three types of signals - the start, stop, and intermediate signals. The radio available at the sender thus should be suitable for transmitting the above signals. In conventional communication systems that use EbT, a digital stream of bits is typically modulated to a higher frequency. To demodulate the bit stream correctly, the receiver has to first synchronize to the carrier frequency or phase of the data stream, and then interpret each frequency shift or phase shift as a *zero* or a *one*. However, this synchronization overhead is usually a few tens of bits, if not longer. Given the typical frame sizes usable with CtS - a few tens of bits, and we discuss this issue in detail in the section on framing - such overhead clearly cannot be accommodated.

Thus, we assume a baseband modulation scheme for the CtS transmissions. The advantage of such a scheme is that there is no carrier present in the data stream, and hence no carrier synchronization overhead. More specifically, we assume a pulse position modulation scheme similar to the one used in ultra-wideband (UWB). Practical technologies for generating and receiving pulses in time durations of nanoseconds are available today [2]. Given the significantly low durations for the pulse-widths, a pulse train rather than a single pulse can be employed while still adhering to a ”signal duration” of a single bit slot. A pulse train consists of monocycle pulses spaced T_f seconds apart in time. The frame time or pulse repetition time T_f typically may be a hundred to a thousand times the pulse width. Note that synchronization preambles are still required to interpret the pulse train, however the synchronization can now be achieved using a relatively shorter number of symbols [3].

We now proceed to elaborate on the specific design challenges associated with the realization of the CtS-MAC.

4.2 Challenge 1: Framing

The problem of framing has to do with determining the length of the messages the transmitter will send as a single unit to the receiver. This is the most basic design decision that needs to be addressed in any MAC layer protocol.

In section 2 we discussed the fundamental energy-throughput tradeoff of the CtS strategy. While the optimization strategies presented in Section 3 do improve the overall network throughput performance, note that *for a single sensor attempting to transmit specific information, the length of the CtS frame will still determine the amount of delay taken for the transfer of that information*.

Hence, the decision on framing centers on how big the CtS frame size should be in terms of bit slots. In a traditional communication strategy using EbT, the frame size ranges from several hundred bytes to several thousand bytes. This is obviously not a feasible frame size for CtS: if the packet size is 100 bytes, the delay between start and stop signals could be as high as 2^{800} bit slots! On the other hand, it is also not desirable to have a very small CtS frame size, since in this case, the energy savings brought by CtS scheme may not be significant enough, and may be even offset by the synchronization overhead required.

Our preliminary empirical evaluations have shown the practical CtS frame size to be 256-65536 bit slots, which translates to a raw data frame size of 8-16 bits. The above frame size range, without any of the optimization strategies presented in Section 3, translates into 10Kbps and 100bps respectively for a 10Mbps raw data rate network. However, it remains to be determined how other environment factors will impact the choice of the frame size.

Note that in the frame size range obtained through the empirical evaluations, there is an obvious trade-off between the energy and throughput enjoyed by the sensors, depending upon the specific frame size chosen. However, there also exists another impact of the frame size, which is related to the contention resolution mechanism used. Briefly, when the frame sizes are large, the *silent periods* in CtS tend to be larger, and this in turn reduces the chances of collisions when simple contention resolution mechanisms are employed. We revisit this issue later in this section. Specifically, with respect to this impact, sensors might have to adapt the frame size depending upon the load conditions within their vicinity.

Another critical issue that needs to be addressed with respect to the framing strategy in CtS is what abstraction is provided to the higher layers. Higher layer protocols ideally should not be burdened with the task of segmenting messages into the small sized frames required by CtS. One strategy is to still provide a "regular" link layer abstraction (with conventional frame sizes) to the higher layer, and hence higher layer protocols can provide CtS with "regular" sized frames. This "regular" frame can be split by CtS into frames of the appropriate size, sent to the receiver back to back, and reassembled back into the corresponding "regular" link layer frame by the receiver, before being processed again.

4.3 Challenge 2: Addressing

While framing determines the length of the units of messages being transferred between a transmitter and its receiver, addressing is the critical functionality through which the transmitter indicates which of its neighbors a message is destined for, and hence should act as a receiver.

Recall from section 3 the assumption about the addressability of the three CtS signals. Specifically, both the sender and the receiver need to be identifiable when a CtS frame is transmitted. While the conventional approach to embed the sender and receiver identifiers in the CtS frame is a possibility, such an approach would impose an additional overhead on the already limited sized CtS frame.

A simplistic solution would be to record the original global source and destination addresses only in the "regular" link layer frame handled by CtS. However, for the CtS frames, locally computed addresses that distinguish only between sensors in a neighborhood will be encoded, thereby reducing the number of bits required for the CtS addressing. A local coloring algorithm can be used to allocate the local addresses. Once the local addresses are determined, the signals can then be modulated with the address information. To modulate the signal with address information, each user is assigned a distinct code, and the code may be translated to a pulse shift pattern or a phase shift pattern, which causes the pulses in the pulse train to shift away from the original sequence in pulse locations or phases. At the receiver, a correlator can be used to detect and recover the modulated signals.

While the above strategy is a high level overview of how the addressing issue can be handled, several open issues still remain to be addressed. How the coloring can be performed in a localized fashion, and how often it needs to be performed has to be addressed. More importantly, the specifics of how much of an overhead the addressing mechanism turns out to be needs evaluation, and the details of the mechanisms used need to be derived.

4.4 Challenge 3: Sequencing

Sequencing is the ability of a receiver to reconstruct information received from a transmitter in the same order that the transmitter had originally sent it, in the presence of reordering either due to losses or other reasons.

In a traditional communication scheme, a packet sequence number is required to cope with the problem of out of order packet delivery. The same situation may also occur in CtS. However, again due to the limited length of the CtS frame, adding sequence number to frames will result in significant reduction in throughput.

One possible solution is to not use a sequence number to any frame that is transmitted using the CtS strategy. In such a set-up, the receiver might rely on the presence of sequence numbers in the "regular" link layer frame as in the traditional EbT scheme.

While this is a simple strategy and does away with the overhead issue, not having sequence numbers in the smaller CtS frames requires that all frames be delivered in order. Also, any completeness check can be performed only by waiting for all CtS frames to arrive at the point where the check is being performed.

4.5 Challenge 4: Error Control

Error control is the problem of detecting errors in the received information, and possibly being able to recover from the errors.

Since CtS uses intervals between signals to infer information, the correctness of the scheme relies heavily on the precise delivery of the start/stop/intermediate signals, and perfect synchronization of sender-receiver counting clocks during a single transmission [1]. Any discrepancy in the arrival time of the signals at the receiver, either due to channel characteristics or processing characteristics, will result in an error in the information delivered.

Due to the specific form of communication used in CtS, well-established error control techniques traditionally used for normal EbT based communication schemes may not be applicable. Hence, the challenge is to properly incorporate error control measures into the system design, so as to achieve robust and efficient data delivery simultaneously.

A straw-man solution is for CtS-MAC to again rely solely on error control only at the granularity of the "regular" link layer frames, where traditional FEC (or ARQ) techniques such as Reed-Solomon code or maximum-erasure-burst-correcting code can be used to encode the raw data before transmission. The receiver, upon assembling back to back CtS frames from the same sender into a "regular" link layer frame performs error control at the level of that frame. However, the performance of such a simple strategy can be expected to be sub-optimal, and needs to be explored.

Another form of error control that can be explored is the use of *delay* as an error control mechanism. For example, consider a coding scheme where the value i is encoded as $2i-1$ before being transmitted. Thus, if the transmitter has to transmit the value 5, the value 9 is actually transmitted. Note that under such a coding scheme, only *odd values* are valid under reception. Thus, if the value at the receiver is interpreted as 10 due to an error, the receiver immediately can detect the occurrence of an error. The trade-off in such a solution is the additional delay spent for the CtS communication.

4.6 Challenge 5: Contention Resolution

Contention resolution is the problem of determining which of the sensors sharing a common channel in a neighborhood gets to transmit at any given point in time.

Note that approaches such as *carrier-sensing* will have no application in the CtS strategy as the only signals transmitted are the start, stop, and intermediate signals, and such signals last merely for a bit slot duration. While sophisticated contention resolution al-

[1]Note that use of CtS does not require perennial synchronization between two sensors.

gorithms that depend upon the *values* each sensor wants to transmit can be devised (note that unlike in a traditional EbT based scheme where sensors have to look out for overlapping packet durations, in CtS sensors have to look out only for overlapping start-stop-intermediate signals, the positions of which are directly determined by the values), WSNs being resource constrained environments, such algorithms can turn out to be highly resource intensive.

On the other hand, a unique characteristic of CtS might allow for a relatively simple contention resolution scheme to be used very effectively. Recall that the empirically determined CtS data frame size is in the range of 8-16 bits. In other words, CtS on an average conveys a data frame size of 12 bits using 2 signals. In other words, for k signals transmitted, CtS in effect transfers $6 * k$ bits of raw information. Now, consider the CtS (delay) frame size of 256-65536 slots with an average size of 4096 slots (2^{12}). If the average number of signals transmitted during the 4096 slots is 800 signals, the total amount of information transferred can still be 2400 bits for an effective utilization of more than 50%. Now, note that the 800 signals translates to a average per bit slot access probability of only about 20%.

This characteristic of CtS can potentially be leveraged to require a relatively low access probability to generate reasonable throughput, and use a simple ALOHA medium access control strategy. Since ALOHA's performance reasonably scales well at low loads of less than 20%, the performance of CtS does not suffer inordinately. At the same time, effective throughput utilization is maintained as justified above. Thus, a sensor, when it wants to transmit, merely goes ahead and transmits by picking a random bit slot within the frame slot window without regarding to what other sensors in the vicinity are performing. If collisions occur, they will be detected by the error control strategy, which will then trigger retransmissions at the "regular" link layer frame level.

4.7 Other Challenges

We have thus far discussed specific challenges pertaining to the medium access control layer when using CtS. While it is true that the most impact of using the CtS strategy will be on the MAC layer mechanisms, CtS will have an impact on other higher layer protocols as well. Due to space constraints, we do not delve into such challenges in this paper, and instead refer readers to [4].

5. RELATED WORK

The possibility of using intervals between data transmissions to convey covert messages has been studied earlier in the context of *timing channels* [5]. In this approach, the durations of intervals between consecutive packets are translated into certain information in an alphabet at the sender. This scheme is similar to CtS in that it uses the time interval to transmit information. However, the timing channel approach is primarily proposed for secure communication, and hence throughput and energy consumption are not major concerns. Furthermore, the granularity of the timing channel solution is at the packet level, while for CtS, the granularity is at a bit level. Hence, it is difficult to apply similar optimization strategies as those in CtS to the timing channel approach, since packet sizes are usually comparable to the interval lengthes, or even bigger.

In the context of modulation schemes, DPIM (digital pulse interval modulation) [6] has been proposed to convert bits into the number of time slots between two consecutive pulses. For example, assume each interval between two pulses in DPIM represents 3 bits, then an interval of length 6 time slots represents the bit sequence ”110”. This principle of using the length of time intervals to convey information is similar to that of CtS. However, as a modulation scheme, the durations between consecutive pulses in DPIM is usually short, and hence DPIM cannot achieve significant energy savings as CtS does. From the network stack's perspective, DPIM is a pure PHY layer modulation solution. Furthermore, it's not possible to apply CtS optimization strategies such as cascading, multiplexing and fast-forwarding to DPIM due to the short durations between pulses. Hence, DPIM can still be considered to be a conventional energy based communication strategy.

6. CONCLUSIONS

In this paper we propose a fundamentally different communication paradigm called *Communication through Silence* (CtS) to achieve energy-efficient communication without significant degradation on overall throughput in WSNs. The proposed scheme primarily uses silence, along with a minimal amount of energy to deliver information between sensors. We analyze the primary energy-delay tradeoff inherent in this approach as well as other challenges related to the realization of the proposed communication strategy. We present several optimization strategies that can be used along with CtS, and can help in improving its throughput performance while retaining its energy benefits. Our performance evaluations show that CtS can deliver considerable improvement in energy performance in WSNs.

7. ACKNOWLEDGEMENT

We gratefully acknowledge the very important feedback given to us by our shepherd Per Gunningberg during the preparation of the camera ready version of this paper.

8. REFERENCES

[1] "Pc4500/pc4800 developers reference manual.," in *Document number: 710-004247*, Aironet Wireless Communications.

[2] Moe Z. Win and Robert A. Scholtz, "Impulse radio: How it works," in *IEEE Communication Letters*, 1998, pp. vol. 2, No.2.

[3] Cao Rui Zheng Yuanjin and Lian Yong, "A new synchronization algorithm for uwb impulse radio communication systems," in *IEEE International Conference on Communication Systems*, Singapore, 2004, pp. pp. 25–29.

[4] Yujie Zhu and Raghupathy sivakumar, "Communication through silence in wireless sensor networks," in *Technical Report*, 2005.

[5] R.G. Gallager, "Basic limits on protocol information in data communication networks," in *IEEE Transaction on Information Theory*, July 1976, pp. 385–398.

[6] Z. Ghassemlooy E.D Kaluarachi and B. Wilson, "Digital pulse interval modulation for optical free space communication links," in *IEE Colloq. On Optical Free Space Communication Links*, Feb. 1996.