Design of transmission manager in heterogeneous WSNs

Jinseok Yang, Alper Sinan Akyurek, *Student member, IEEE,* Sameer Tilak, and Tajana Simunic Rosing, *Senior member, IEEE*

Abstract

The current generation of sensor networks are designed to be application-specific, thus are exposed only to a limited set of users. The emerging concept of IoT is expected to house multiple applications with diverse delay requirements. A transmission manager provides an optimal transmission time for transmitting the buffered measurements. In the literature, solutions have been proposed optimizing mainly for single sensing infrastructures. In this work, we first propose an optimal transmission manager that supports multiple applications in a single-hop wireless sensor networks. Then, we extend our solution into a distributed transmission manager to operate in multi-hop WSNs. Both transmission managers work in tandem, and determine the transmission time for every buffered measurement. We implement both solutions in ns3 and compare with other state of the art solutions. Our case studies show that our proposed solution reduces energy consumption by 75% compared to the state of the art approaches while having on average 12% less expired measurements.

Index Terms

Wireless sensor network, IoT, Transmission manager, Application layer delay

- *Jinseok Yang was with University of California, San Diego, CA, 92093.*
  E-mail: jinsuksfa@gmail.com / jiy011@ucsd.edu
- *Alper Sinan Akyurekis is with University of California, San Diego, CA, 92093.*
  E-mail:aakyurek@eng.ucsd.edu
- *Sameer Tilak was with CalIT2, University of California, San Diego, CA, 92093.*
- *Tajana Simunic Rosing is with University of California, San Diego, CA, 92093.*

Manuscript received Dec 8, 2015; revised xxx xx, xxx.

Design of transmission manager in heterogeneous WSNs

1 Introduction

The new generation of Wireless Sensor Networks (WSNs) envisions commodity sensing & actuation infrastructure to provide services to the Internet of Things (IoT). These WSNs form a critical and general interface between the physical and cyber worlds. They convert physical quantities into measurements which can be used for a wide spectrum of applications. The trend of WSNs are shifting from the traditional specialized network solutions that run a single application and support a limited set of users, into a multiple application network, deployed and shared by possibly multiple organizations. This results in the running applications to have diverse delay and accuracy requirements.

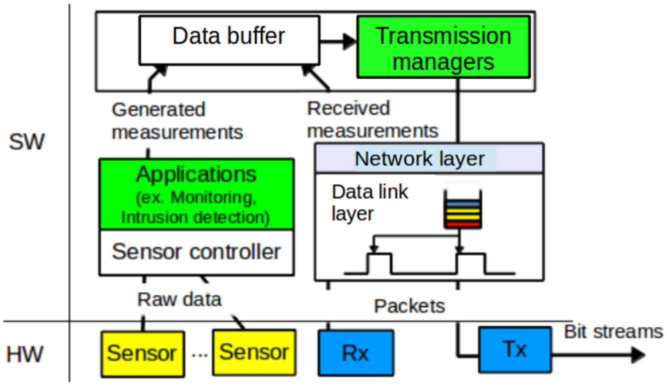


Fig. 1. Applications generate measurements which have application-specific delay requirements and send them to the data buffer in the transmission manager. Transmission manager determines the transmission time for the buffered measurements.

Fig. 1 describes the operation of a sensing platform which collects data from heterogeneous measurement sources in WSNs. Each sensing platform runs multiple applications that collect raw data from the corresponding sensors. Applications process the raw data to generate measurements, and send these measurements along with application-specific delay requirements to the data buffer in the transmission manager. In addition to generating its own data, the sensing platform also receives packets to be forwarded from its neighboring nodes as WSNs may use multi-hop wireless routes to forward the data to the sink node. Thus, the data buffer in the transmission manager has both the measurements generated by the node itself as well as the ones received from its neighbors. The job of the transmission manager is to determine the optimal single hop transmission instance of buffered measurements, based on the end-to-end delay requirements and distance to the sink along a routing path. Recent publications that look at delay guarantees in multi-hop WSNs [1] [2] decompose the end-to-end delay problem into a set of single-hop delay subproblems. However, they don't determine a specific solution for how each node in WSNs will obtain a single-hop delay requirement. They show that the end-to-end delay guarantee problem requires information about the buffer, the channel, and the system conditions of ancestor nodes in the routing paths such that the problem is NP-hard.

The key challenge of the transmission manager is to minimize the number of messages that expire prior to reaching their destination while minimizing the energy consumption. In this work, we first propose an optimal transmission manager using the optimal stopping theorem based on Markov Decision Process model (MDP) in order to obtain the optimal transmission instance between a pair of nodes. Then, we propose a distributed transmission manager that modifies the optimal transmission manager to operate in multi-hop WSNs. The optimal and distributed transmission managers work in tandem (ref. Fig. 5), and run on each node as shown in Fig. 1. In this figure, our approaches for the transmission managers are implemented in the green box. They determine the transmission time for every buffered measurement. Measurements expire if they cannot arrive to a sink node before the application specific time constraint.

We implement our solutions in Network Simulator 3 (ns3) [3] and compare with the other state of the art transmission managers [4] [5] [6]. We evaluate the energy consumption and the number of expired measurements for three different network topologies: i) single hop, ii) multi hop linear topology and iii) multi hop grid topology. The single hop case is typical for small scale wireless sensor networks such as Body Area Networks (BANs) [7]. The linear topology is commonly used in buoy deployments in order to monitor lake conditions such as temperature, dissolved oxygen, conductivity and pH/ORP. The grid topology has been shown to be effective for city-wide deployments such as air quality monitoring in downtown San Diego for projects, such as CitiSense [8]. In all three cases, our optimal and distributed solutions work in tandem, and consume on average 75% less energy than the state of the art approaches, while having on average 12% less measurement that expire.

This work is structured as follows: first, we summarize the related work on transmission managers in Section 2. Section 3.1 formulates the optimal transmission instance problem using Markov Decision Process (MDP) model and proves the existence of an optimal transmission time. Details of our proposed optimal transmission manager are explained in Section 3.2. Section 3.3 describes how our proposed transmission manager determines the optimal transmission time for buffered measurements in a single hop WSN case. In Section 3.4, we describe how our proposed transmission manager operates in multi-hop WSNs. Section 4 discusses the experimental setup. Finally, network simulation results with single hop, linear and grid topologies are summarized in Section 5.

2 Related Work

There is a small body of publications focusing on designing a transmission manager that minimizes the energy consumption, while ensuring a low number of expired measurements. These works are based on decreasing energy consumption by decreasing the number of communication instances using an embedded buffer. Then, the transmission time is estimated based on different factors such as local application delay constraints, distance to the sink node or node's available energy.

The periodic per hop approach [4] waits periodically for a predefined time interval before forwarding the received data to its parent node. The cascade time-out protocol, presented in the same publication, buffers all generated and received measurements in its buffer and transmits at a calculated transmission time. The calculation is based on a function of the hop distance to the sink node (h), average single hop delay ($D_s = 0.03$ sec in [4]) and sampling interval (SI) as Transmission Time $= 2\,(\text{SI} - (D_s \times h))$.

Transmission managers that consider application based delay constraints are proposed by [5] [6] [9]. These solutions determine the optimal transmission time of buffered measurements based on the characteristics of a running application. The selective-forwarding approaches [5] [9] maximize the reward of all measurements that are generated in the wireless network to achieve energy reduction while minimizing the number of expired measurements. They calculate the importance of a measurement as: $\alpha E[E_{avail} - E_{req}]$, where E_{avail} is the available energy of the node and E_{req} is the necessary energy for transmission and α is the delay-sensitivity factor. Expectation is taken over the probability of arrival to the sink node. The importance value is compared with a threshold, upon which the measurement is discarded or transmitted. Threshold is the total expected importance at a time instant. Selective-forwarding approaches assume that all nodes in the wireless network run a single delay sensitive application. In contrast, the delayed forwarding approach [6] uses α to represent the trade-off between energy gain and the expiration rate of measurements. For example, a large α value requires a node to schedule a transmission within a short amount of time. While this decreases the probability of measurements timing out, it also increases energy consumption. The choice of α is arbitrarily specified by the authors and can be inverse function of measurement lifetime. In experimental section, we evaluate the effect of different α to both energy consumption and measurement expiration rate. At initial state, the delayed forwarding approach [6] transmits all buffered measurement when the buffer size is over a predefined threshold. Meanwhile, this approach updates the threshold with the arrival rates (λ), average sampling rate (μ) over a time interval and inverse function of lifetime of currently buffered measurements (α) as: $\left\lceil \frac{\lambda \cdot \mu}{\alpha(\alpha+\mu)} + 1 \right\rceil$.

Although single application based WSNs have been studied extensively, prior works do not consider different delay requirements of heterogeneous applications. This is the limitation that our proposed solutions seek to resolve in this work.

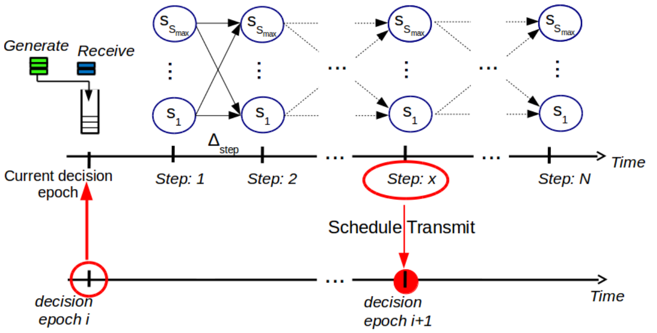


Fig. 2. At every decision epoch, transmission manager uses Markov Decision Process model and find out optimal transmission instance of buffered measurements. Black arrows represent the transitions between the states

3 Optimal Transmission Manager

The system model for our proposed solution is shown in Fig. 1. The distributed and optimal transmission managers operate between the running applications and the network layer on each sensing platform. All generated or received measurements are stored in the data buffer. They contain four types of information: i) raw data obtained from embedded sensors, ii) end-to-end time constraints determined by the applications [1] [2], iii) destination address and iv) application id. Once the optimal transmission manager decides to transmit, it flushes its measurements in the data buffer into the network stack which manages the end-to-end communication. If the underlying communication protocol is based on Internet Protocol (IP), all fields can be implemented into IP headers and tunneled through. However, practical considerations are outside the scope of this work.

Our optimal transmission manager (OptTM) determines the optimal transmission time of the measurements based on the time constraints of buffered measurements in order to minimize the number of measurements that expire while keeping the energy consumption low. OptTM makes the transmission time decisions at regular time intervals called *decision epochs*, as shown in Fig. 2.

We divide time into equal length discrete intervals that we designate as *steps*. The time interval between each step is defined as Δt. Our solution checks each future step as a possible transmission instance candidate and decides whether to schedule a transmission at that step. This decision is determined using a reward based MDP model. In this section, we first prove that an optimal transmission time exists and then describe how to find that optimal solution. The following notations are used throughout this section:

- Steps are indexed using k, where $k = 1$ is the first future step that is being investigated. The maximum MDP horizon is designated as N, such that $k \in [1, N]$.
- The set of *States*, $S_k = \{s_1, s_2, \ldots\}$, where each state value represents the buffer characteristics at step k. Since our solution makes a binary decision, the only state values are $S_k : \{s_{best}, s_{not}\}$, designating whether the maximum reward was achieved at step k or not.

- The set of *Actions*, $A_k = \{a_1, a_2, \ldots\}$, where each action determines the next state of the buffer. For our solution, the actions are $A_k : \{\text{transmit}, \text{wait}\}$.
- Transition probability matrix, $P_k(S_{k+1}|S_k, A_k)$. It defines the likelihood of the next state based on the current state and the current action. For example, if the chosen action at step k is to transmit all measurements in the buffer ($A_k = \text{transmit}$), the state of next step $k+1$ is an empty buffer.
- Reward, $R_k(S_k, A_k)$, is a functional mapping from a state and action into a quantitative reward that we wish to maximize as our optimization goal.

Based on these definitions, OptTM selects the action of transmit at step k, ($A_k = A^* = \text{transmit}$), only if that action returns the maximum expected future reward, ($R_k = R^*$), as described in Eq. (1) and Eq. (2).

$$\begin{aligned} A_k^* &= arg \max_{A_m \in A} \{R_k(S_k, A_m) \\ &\quad + \sum_{S_{k+1} \in S} P_k(S_{k+1}|S_k, A_m) R_{k+1}^*(S_{k+1})\} \end{aligned} \tag{1}$$

$$\begin{aligned} R_k^*(S_k) &= \max_{A_m \in A} \{R_k(S_k, A_m) \\ &\quad + \sum_{S_{k+1} \in S} P_k(S_{k+1}|S_k, A_m) R_{k+1}^*(S_k)\} \end{aligned} \tag{2}$$

3.1 Criteria for optimality

Based on our system definitions, we derive the criteria for optimality. We use the following definition of *super-additive functions* and theorem for the existence of an optimal policy:

Definition 1. *Let S and A be partially ordered sets and $f(s, a)$ a real-valued function on $S \times A$. The function $f(s, a)$ is super-additive when $f(s', a') - f(s, a') \geq f(s', a) - f(s, a)$ with $s' > s$ in S and $a' > a$ in A.*

This definition requires an ordering of the states. Since we have a binary decision, where the s_{best} has the maximum reward, we assign the following arbitrary values: $s_{not} = 0, s_{best} = 1, \text{wait} = 0, \text{transmit} = 1$. Note that any value assignment would work here, as long as the ordering agrees with the resulting reward value order to satisfy the following theorem.

Theorem 1. *There exists an optimal structured transmission rule if the formulated problem satisfies the following conditions [10]:*

1) *$R_k(S_k, A_k)$ is nondecreasing in $S_k, \forall A_k \in A, \forall k \in [1, N-1]$.*
2) *Cumulative transition probability, $q_k(S'|S_k, A_k) = \sum_{S_{k+1} \in [S', \ldots]} P_k(S_{k+1}|S_k, A_k)$, is nondecreasing in $S_k, \forall k \in [1, N-1], \forall A_k \in A, \forall S' \in S$.*
3) *$R_k(S_k, A_k)$ is super-additive function on $S \times A$.*
4) *$q_k(S'|S_k, A_k)$ is super-additive function on $S \times A$.*
5) *$R_N(S_N, A_N)$ is nondecreasing in S_N.*

Then, the optimal value is nondecreasing in state $S \leq S'$: $R_k^*(S) \leq R_k^*(S')$.

Theorem 1 states that, if the reward function selection is super-additive and nondecreasing in all states for all actions at all steps, then an optimal transmission instance exists for that reward function. By definition, the reward function becomes super additive when it satisfies the following relationship with $S' \geq S$ and $A' \geq A$: $R_k(S', A') - R_k(S, A') \geq R_k(S', A) - R_k(S, A)$. We first derive a sufficient condition to satisfy the above two relationships with an assumption that the reward function is separable into the multiplication of two nondecreasing functions: $R_k(S, A) = f(S) \cdot r_k(A)$. With this assumption, we can derive a sufficient condition for super-additivity of $R_k(S, A)$ as described in Eq. (3).

$$\begin{aligned} &R_k(S', A') - R_k(S, A') \geq R_k(S', A) - R_k(S, A) \\ &= r_k(A')\{f(S') - f(S)\} \geq r_k(A)\{f(S') - f(S)\} \\ &= r_k(A') \geq r_k(A) \end{aligned} \tag{3}$$

The sufficient condition is satisfied as r_k was assumed to be nondecreasing. The cumulative transition probability is also nondecreasing as proven in Eq. (4).

$$\begin{aligned} &q_k(S'|S'_k, A_k) - q_k(S'|S_k, A_k) \\ &= \sum_{S_{k+1}=S'}^{\infty} P_k(S_{k+1}|S'_k, A_k) - P_k(S_{k+1}|S_k, A_k) \geq 0 \end{aligned} \tag{4}$$

We check the value for three possible cases: When $S' \geq S'_k$, the difference becomes $\sum_{S'}^{S'_k - S_k} P_k(S'|S_k, A_k)$. For $S_k \leq S' < S'_k$, it becomes $\sum_{S'=S_k}^{S'_k} P_k(S'|S_k, A_k)$. And for $S' < S_k$, the difference is 0. In all three cases, q_k is always nondecreasing.

With these properties, our model guarantees that an optimal transmission time exists. In the next section we discuss how we define all of the parameters of our model in order to guarantee the optimality.

3.2 Optimal transmission manager implementation

At every decision epoch, OptTM checks the buffer state for each time step and determines when to transmit the currently buffered measurements. We define the reward at step k, R_k, as the sum of all buffered measurements' remaining lifetime and the expected lifetimes of the future arrivals. Previous works [5] and [6] use an exponential function as their reward to emphasize the effect of decreasing message lifetime with delay. However, through case studies we have observed that the exponential function decreases the delay too fast under realistic applications resulting in high energy consumption. Thus, we use a linear function of remaining lifetime as shown in Eq. (5) and Eq. (6).

We use the index i to represent the i^{th} application and define the number of applications as N_{app}. We define $T_{m,k}^{life}$ as the remaining lifetime of the m^{th} measurement in the buffer at the k^{th} future step as shown in Eq. (6). It is calculated by taking the difference between the deadline of the measurement, D_i, and the time at step k, $k\Delta t$. Future measurements are generated by applications and may arrive at different times. Thus, we calculate the expected future reward in terms of the number of running applications, delay and the number of arrivals. The expected delay at the k^{th} step is β_k and γ_k represents the expected number of measurement arrivals. $P_i(\gamma_k)$ is the probability of γ_k arrivals from application i. We model this probability using

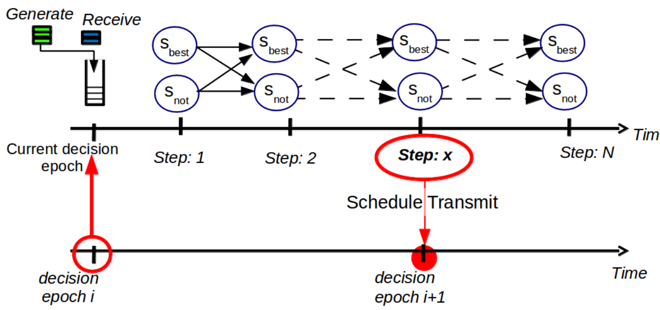


Fig. 3. Optimal transmission manager uses only two buffer states based on optimal stopping theorem

a Poisson distribution as described in Eq. (7), where λ_i is the arrival rate parameter. $E\{D\}$ is the expected deadline of future measurements.

$$R_k = \sum_m T^{life}_{m,k} + \sum_{i=1}^{N_{app}} \sum_{\beta=1}^{\Delta t(k-1)} \sum_{\gamma_k=0}^{\infty} \gamma_k \cdot \{E\{D\} - \beta_k\} P_i(\gamma_k) \tag{5}$$

$$T^{life}_{m,k} = \max\{D_i - k \cdot \Delta t, 0\} \tag{6}$$

$$P_i(\gamma) = e^{-\lambda_i}(\lambda_i)^{\gamma}/\gamma! \tag{7}$$

The solution uses backwards iteration [10] and calculates the reward at each step. When the reward at the current step is the greatest so far, the system is in state s_{best}, otherwise, it is in state s_{not}.

Using the Poisson distribution, we can rewrite the reward at step k, $R(s_k)$, as in Eq. (8). The arrival rate of any application is determined by counting the measurements between two consecutive decision epochs.

$$R_k(s_{best}, A_k) = \sum_m T^{life}_{m,k} + \sum_{i=1}^{N_{app}} \sum_{\beta_k=1}^{\Delta t(k-1)} \{E\{D_i\} - \beta_k\} \cdot \lambda_i \tag{8}$$

Since transition probability only exists when OptTM continues buffering ($A_k =$ wait), the optimal reward at step k, $k < N$, can be simplified as in Eq. (9). We define the reward at the end point as $R^*_N(s_{best}) = 1$ and $R^*_N(s_{not}) = 0$ due to the nondecreasing property of the reward function.

$$R^*_k(S_k) = \max\{R_k(S_k, A_k = \text{transmit}), E[R^*_{k+1}(S_{k+1})]\} \tag{9}$$

The $R_k(S_k, A_k = \text{transmit})$, $k \leq N$ is the reward of buffered measurements at step k when OptTM transmits all buffered measurements. $R^*_{k+1}(S_{k+1})$ is the optimal reward at the next step, $k+1$. The expected reward for the next state, $E[R^*_{k+1}(s_{k+1})]$ is calculated as in Eq. (10).

$$E[R^*_{k+1}(S_{k+1})] = \sum_{S' \in \{s_{best}, s_{not}\}} R^*_{k+1}(S')P(S') \tag{10}$$

The expression can be further expanded into:

$$\begin{aligned} R^*_k(s_{best}) &= \max\{R_k(s_{best}), E\{R^*_{k+1}(s_{k+1})\}\} && (11) \\ &= \max\{R_k(s_{best}), P(s_{best}) \cdot R^*_{k+1}(s_{best}) \\ &+ P(s_{not}) \cdot R^*_{k+1}(s_{not})\} \\ R^*_k(s_{not}) &= \max\{0, P(s_{best}) \cdot R^*_{k+1}(s_{best}) && (12) \\ &+ P(s_{not}) \cdot R^*_{k+1}(s_{not})\} \end{aligned}$$

Since the expected reward is nonnegative, we can simplify further:

$$R^*_k(s_{best}) = \max\{R_k(s_{best}), E[R^*_{k+1}(s_{k+1})] = R^*_k(s_{not})\} \tag{13}$$

This result defines the optimality criteria. In state s_{best}, when $R_k(s_{best}) \geq E\{R^*_{k+1}(S_{k+1})\}$ the optimal action is to transmit all the currently buffered measurements. When $R_k(s_{best}) < E\{R^*_{k+1}(S_{k+1})\}$, the optimal action is to continue buffering without transmitting measurements. Simply put, the optimal transmission instance is which maximizes the reward while minimizing the number of measurements that expire as shown in Eq. (14).

$$N^* = \arg\min_k\{R_k(s_{best}) \geq E\{R^*_{k+1}(s_{k+1})\}\} \tag{14}$$

We can select any probability distribution that satisfies Eq. (4). We select $P_k(s_{best}) = \frac{1}{k+1}$, which states that the reward at next step $k+1$ is the largest seen so far if no information about the system is present. As s_{best} and s_{not} are mutually exclusive events, $P(s_{not}) = 1 - P(s_{best})$. This results in the optimal transmission instance being:

$$N^* = \arg\min_k\{R_k \geq R^*_{k+1}\} \tag{15}$$

ALGORITHM 1: Optimal single hop transmission manager (OptTM)

```
Input 1: time constraints set $\boldsymbol{T}$
$\boldsymbol{A} = zeros(1, N)$, this is action vector
$\boldsymbol{R} = zeros(1, N)$, this is reward vector
for $k$=1 to $N$ do
    Calculate $\boldsymbol{R}(k)$ with Eq. (8)
end
// Determine optimal action
for $k$=$N$-1 to 1 do
    if $\max \boldsymbol{R}(1:k) \neq \boldsymbol{R}(k)$ then
        //Current state is $s_{not}$
        $\boldsymbol{A}(k) \leftarrow$ wait // Continue ;
    else
        //Current state is $s_{best}$
        Calculate $E[R^*_{k+1}]$ with Eq. (10) and
         apply transmission probabilities
        if $\boldsymbol{R}(k) \geq R^*_{k+1}$ then
            $\boldsymbol{A}(k) \leftarrow$ transmit // transmit all currently
             buffered measurements;
        else
            $\boldsymbol{A}(k) \leftarrow$ wait ;
        end
    end
end
if $\exists k, A(k) = transmit$ then
    Output: optimal transmission instance =
     $\min\{k \geq 1 : A(k) =$ transmit$\}$
end
```

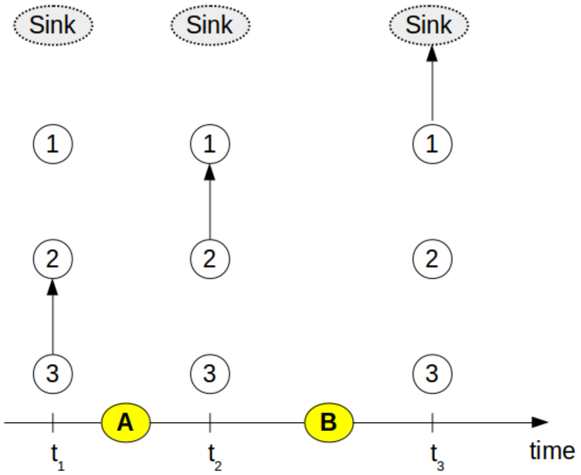


Fig. 4. Additional delays at relay nodes (A+B)

3.3 OptTM algorithm for a single hop WSNs

We use the theoretical result in the previous section to devise our algorithm, as described in pseudo-code Algorithm 1 for a single hop WSN. Inputs to the algorithm are the current buffer with time constraints of all measurements, $\boldsymbol{D}$, and maximum decision horizon, N. OptTM calculates the reward (Eq. (8)) for each potential transmission step in lines 4:6. When the current reward is not the greatest one calculated at previous steps (line 9), OptTM continues buffering without transmitting measurements, Eq. (12). Otherwise, OptTM compares the current reward with the expected reward (line 15). It transmits all the measurements if the current reward is greater than or equal to the expected reward. Finally, OptTM selects the earliest step for which Eq. (14) is satisfied. This is then defined as the optimal transmit time (line 23). The computation and space complexity of this algorithm is $O(N)$.

3.4 Distributed transmission manager

We extend our optimal single hop solution into the multi hop case. We assume that the applications assign end-to-end time constraints based on their own delay requirements for the generated measurements. Measurements coming from different applications may have different time constraints. Our solution, designated as Distributed transmission manager (DistTM), calculates the maximum decision horizon, N, and uses the single hop solution, OptTM, at each decision epoch. Instead of calculating the rewards using only generated measurements, DisTM also uses the received measurements to be forwarded to decide when to transmit. This process is described in pseudo-code Algorithm 2.

In lines 6:7, distributed transmission manager calculates N. Algorithm 1 uses N as an input and calculates optimal transmission instance between 1 and N in line 9. After distributed transmission manager gets the transmission instance, it updated end-to-end time constraints of all original measurements in $\boldsymbol{B_{in}}$ in lines 11:16. When updated end-to-end time constraints have zero or negative values, DistTM removes corresponding measurements from $\boldsymbol{B_{in}}$. Lastly, DistTM schedules transmission of all measurements in $\boldsymbol{B_{in}}$ at the optimal transmission instance in line 17.

ALGORITHM 2: Distributed multi-hop transmission manager (DistTM)

```
1  Input 1: buffer $\boldsymbol{B_{in}}$
2  Input 2: $h_{sink}$: distance in number of hops
    from current node to sink
3  $length(\boldsymbol{B_{in}})$: the number of measurements in
    $\boldsymbol{B_{in}}$
4  $D_i$: end-to-end time constraint of $i^{th}$
    measurement in $\boldsymbol{B_{in}}$
5  // Calculate upper bound of transmission
    delay
6  Calculate $N^*$ with Eq. (17)
7  Calculate $N$ with Eq. (16)
8  // Calculate optimal transmission instance
9  $Optimal\ tx_{instance}$ ← Algorithm 1 with input $N$
10 // Check measurements that expire
11 for $m$=1 to $length(\boldsymbol{B_{in}})$ do
12     $\boldsymbol{B_{in}}(m) = \boldsymbol{B_{in}}(m) - tx_{instance}$ if $\boldsymbol{B_{in}}(m) \leq 0$ then
13         delete $\boldsymbol{B_{in}}(m)$ from $\boldsymbol{B_{in}}$
14     end
15 end
16 Output: Schedule transmission of
    measurements in $\boldsymbol{B_{in}}$ at $tx_{instance}$
```

The computation and space complexity of this algorithm is $O(length(\boldsymbol{B_{in}}) + N)$.

In contrast to other state of the art approaches, DisTM is a non-synchronized approach. This means that there is no time synchronization assumed between the nodes. Naturally, this yields additional delays at the relay nodes as depicted in Figure 4. Node 3 transmits a measurement to Node 2 at t_1. Node 2 relays the measurement to Node 1 at t_2. Node 1, then, relays the received measurements to the sink at t_3. A measurement generated from Node 3 can expire if the additional delays at relay nodes (A+B) is greater than the generated measurement's time constraint. A measurement of Node 2 has additional delay at Node 1, but a measurement of Node 1 does not suffer additional delay because Node 1 can directly connect to the sink. This implies that even though all nodes have same measurements with the same time constraints, Node 3 must have the shortest transmission interval in order to avoid measurement expirations at relay nodes. This means that DisTM must consider the distance to the sink when it calculates the maximum decision horizon, N.

We first consider the worst case delay at relay nodes in multi-hop WSNs. Suppose that all nodes have the same measurements with the same characteristics in their buffer, and they have the same transmission interval N^* sec. Since each node transmits after N^* sec, the worst case delay is a function of the number of hops to the sink and the transmission interval, $h_{sink} \times N^*$. This means that when transmission interval x satisfies the following condition, $h_{sink} \times N^* \leq D \rightarrow N^* \leq D/h_{sink}$, all measurements arrive at the sink without expiring. The best case delay is N^* in case of no additional delay at the relay nodes. The average delay at the relay nodes in multi-hop WSNs is a function of the worst and best case delays as described in Eq. 16. The right hand side of the Eq. (16) is bounded average delay at relay nodes with x sec transmission interval.

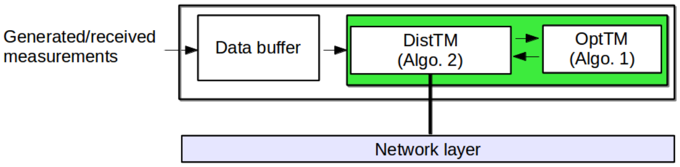


Fig. 5. Three components of multi-hop TM

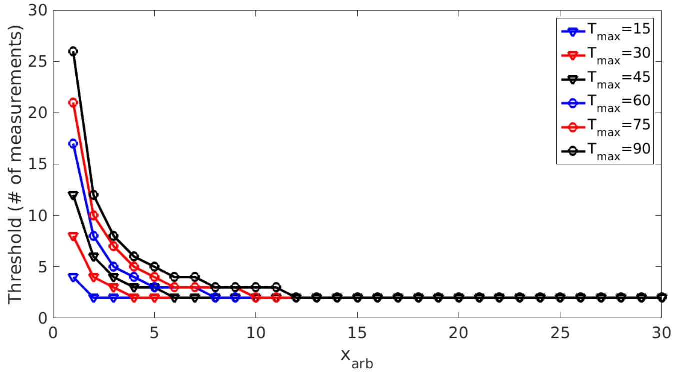


Fig. 6. Buffer threshold of CL for different x_{arb} and maximum time constraints (T_{max}) with $\lambda = 0.6$

$$\frac{D/h_{sink} + N^*}{2} \geq \frac{(h_{sink}+1) \times N^*}{2h_{sink}} \tag{16}$$

The goal of DistTM is to find the maximum horizon, N, by estimating the expected delay at relay nodes with hop distance of h_{sink}. Thus, DistTM uses expected time constraints of measurements, $E(D)$, and calculates N^* as described in Eq. (17). We use a similar method as in [4], because they have the best performance when all nodes are synchronized. Single hop delay is the sum of maximum values of propagation delays in the wireless channel and staggering delay in the network layer [4]. The propagation delay depends on deterministic factors (ex. distance and packet size) and non-deterministic factors (ex. channel condition, weather and moving objects) [11]. Thus, it is impossible to exactly obtain the maximum propagation delay. In [4], authors set the maximum staggering delay to 0.03 sec and estimate the maximum single hop delay, $T_{maxDelay}$, as 0.3 sec for CSMA. We use the same value in our experiments.

$$x = \{E(D) - T_{maxDelay} \times h_{sink}\} \tag{17}$$

4 Experimental Setup

We evaluate the performance of our proposed OptTM in terms of percentage of expired measurements and energy consumption (mJ) compared to the following state of the art approaches:

- $Fixed$ [4]: All nodes have a fixed buffering time limit and periodically transmit all buffered measurements at every instance.
- Cas [4]: The **cascade time-out protocol** considers the distance to the sink node to evaluate the buffering time limit. Farther nodes have shorter buffering time limit. For single-hop analysis, this method converges to the shortest time interval between transmission instances.

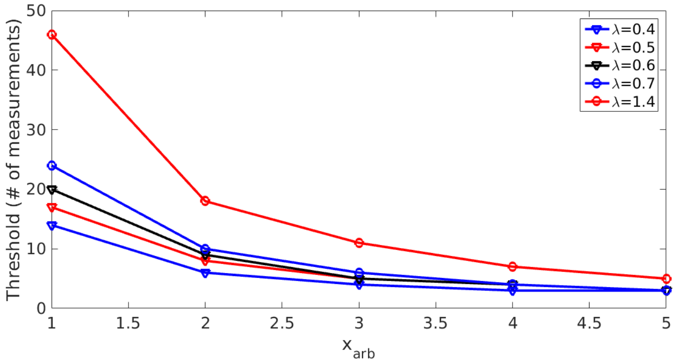


Fig. 7. Buffer threshold of CL for different x_{arb} and λ

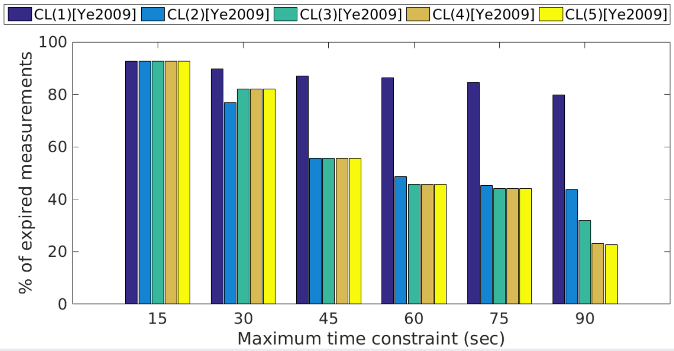


Fig. 8. Percentage of expired measurements

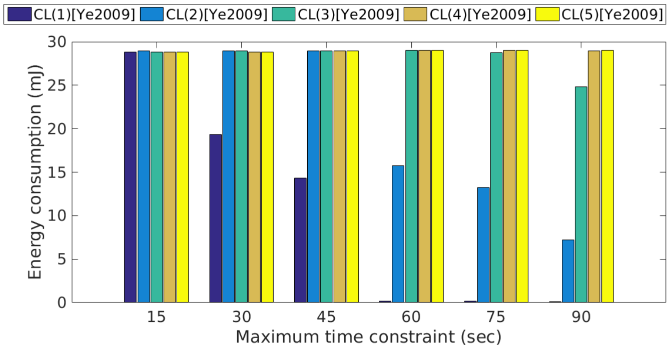


Fig. 9. Energy consumption (mJ)

- CL [6]: The **control-limit transmission manager** transmits all buffered measurements when the buffer size is over calculated threshold. Simultaneously, this approach updates the threshold with a function of arrival rates (λ) and average sampling rate (μ) over a time interval as: $\left\lceil \frac{\lambda \cdot \mu}{\alpha(\alpha+\mu)} + 1 \right\rceil$. The α is a inverse function of average time constraints as: $\alpha = \frac{x_{arb}}{\text{E(time constraints)}}$ where x_{arb} is arbitrary specified by application users. Then, the CL keeps buffering measurements until the buffer size is over the updated threshold. In, Fig. 6, we vary x_{arb} and evaluate the performance of CL when 10 nodes construct linear topology. All generated measurements have uniformly distributed time constraints between 1 and maximum time constraint (T_{max}). We can observe that threshold is converging while we increasing x_{arb} for all T_{max}. More specifically, larger T_{max} makes CL to have larger threshold when it has smaller x_{arb}. For example, when T_{max} is 90 sec, the threshold of CL is 26 when $x_{arb} = 1$. when $T_{max} = 15$ sec, the threshold is 4 which means transmitting buffered measurement more frequently. However, when we evaluate energy consumption of CL with different x_{arb} in Fig. 9, when x_{arb} is less thn 2, CL consumes less energy while we increasing maximum time constraints. However, CL consumes the same amount of energy whenb x_{arb} is larger than 3. This is because the threshod is affected by λ as described in Fig. 7. We can observe that the threshold converges more fastly with smaller arrival rate (λ). In our experiment, λ is varying between 0.4 and 1.4, but average arrival rate is 0.6. In this paper, we select $x_{arb} = 2$ because it decrease both energy consumption and the number of expired measurements while we increaseing maximum time constraints as we can see in Fig. 8 and Fig. 9.

- SF [5]: The **selective forwarding transmission manager** calculates a threshold based on the consumed energy and importance of measurements. Importance is an inverse function of time constraints. It sends a measurement if the measurement's importance is larger than the calculated threshold. Otherwise, SF discards the measurements. As same as CL [6], we use maximum time constraints as α.
- DistTM: The **distributed transmission manager** employs OptTM which determines the optimal transmission instance based on Eq. (8) and Eq. (15) at every decision epoch. At each decision epoch, DistTM calculates upper bound of transmission instance, and OptTM uses it to get the optimal transmission instance. Then, DistTM schedules transmission of all buffered measurements at the optimal transmission instance. The details of OptTM and DistTM are described in Algorithm 1 and Algorithm 2.

We implement all algorithms using ns3 network simulator [3]. Our approach operates on top of the network layer, so it is agnostic to the underlying protocols. We use DSDV [12] as the routing protocol. The physical distance between two nodes is at least 100m.

We consider three different network topologies: i) single hop, ii) linear and iii) grid. The single hop case considers a situation with a single source which is directly communicating with a sink. Such direct communication between a sink and sensors is common in small scale WSNs such as body area networks (BANs) as shown in Fig. 10 [7]. In this case, we use CC2630 at 2.4 GHz [13] as the transmission device. The radio module runs at 3V with transmit current at 6.1mA, receive current at 5.9mA and sleep current of 100 nA. Linear topology establishes a routing path as depicted in Fig. 11. A good example for this case is lake monitoring application that deploys multiple buoys on a lake to monitor temperature, dissolved oxygen, conductivity, pH/ORP, fluorescence (Chlorophyll and Blue-green Algae) [14] to understand important issues such as spatial distribution of algal bloom and invasive species. Lastly, we imagine sensor nodes deployed in San Diego downtown forming a grid topology as shown in Fig. 12. Such topology has also been assumed for air quality monitoring in CitiSense project [8]. Blue node is the sink node while red nodes generate and relay measurements. Nodes in both linear and grid topologies have a non-QoS 802.11b radio which uses ad-hoc mode with 1 Mb/s maximum data rate. The WiFi module runs at 3V with transmit current at 380mA, receive current at 313mA, idle listening current at 273 mA and sleep current at 33mA [3]. Constant speed propagation delay model characterizes the channel conditions.

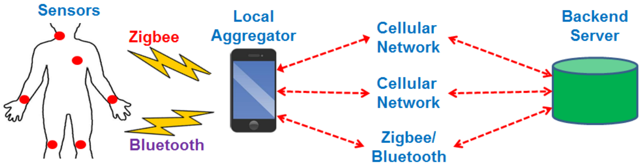


Fig. 10. Wireless healthcare system architecture [7]

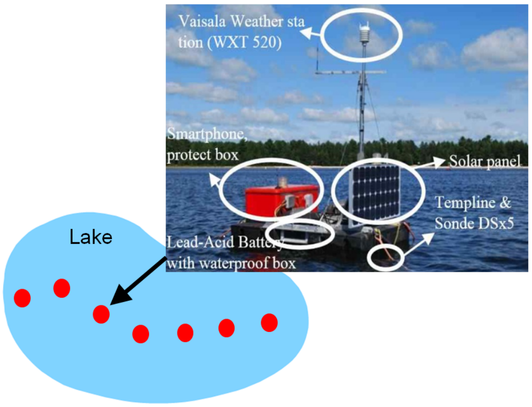

Vaisala Weather sta tion (WXT 520)
Smartphone, protect box
Solar panel
Templine & Sonde DSx5
Lead-Acid Batte with waterproof box
Lake

Fig. 11. Example of linearly deployed buoys in an single lake [14]

We use synchronized and unsynchronized sampling methods. In synchronized scenario, all nodes periodically generate measurements at a predefined time interval. This is typical in monitoring systems which use WSNs to measure fine grained environmental factors [8]. For example, environmental monitoring application such as CIMIS [15] use temperature, humidity and pressure sensors to have a 10 sec sampling granularity. In case of multiplea systems (applications) running on a single WSN, each application has own fixed sampling interval. In our experiment, each application randomly selected sampling time which is uniformly distributed between 1 to 10 sec at initial phase. Then, each application periodically generates own measurements at selected time interval. In unsynchronized scenario, all nodes generate measurements with randomly selected sampling time which is uniformly distributed between 1 to 10 sec. This is typical when different applications require different sampling intervals at different situation. Time constraints of measurements are uniformly distributed between a minimum of 1 sec and a varying value for the maximum value ranging from 15 to 90 sec. The varying maximum value enables us to study the effect of heterogeneous time constraints and effect of randomness. Similar values have been used in deployed WSNs. For example, in the air quality

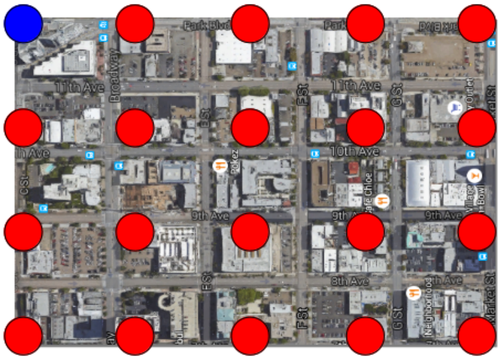


Fig. 12. Considered topology for the realistic network simulation. Node 0 (Blue node) is the sink node. Other nodes generate and relay measurements to the sink node

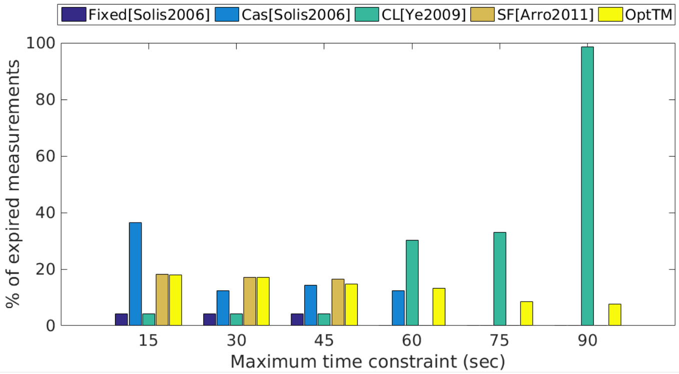


Fig. 13. Percentage of expired measurements

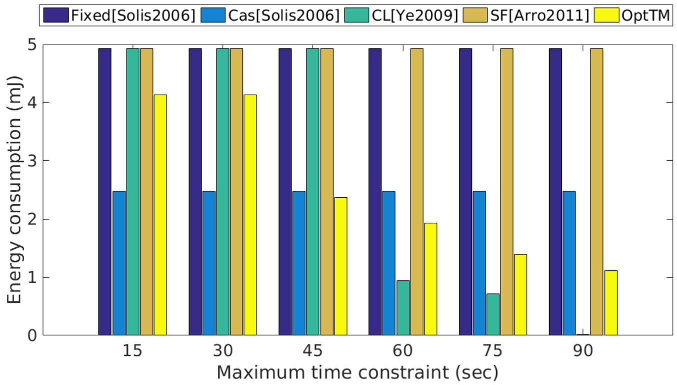


Fig. 14. Energy consumption (mJ)

Body Area Network (BAN)

monitoring system [8] deployed sensors provide data to the patients every 30 sec. Vaisals weather stations [16] measure weather conditions such as temperature and humidity every 1 to 120 sec. The initial time interval between decision epochs is set to be 10 sec. The total simulation time is set to 24 hours to represent a single day. We run 10 different applications on each node. The energy consumption is assumed to be heavily due to communication, rather than processing. There are two major reasongs: 1) we have an $O(N)$ linear solution, 2) the processing power compared to communication power is relatively low [13]. Aaron et al. [17] analyse of power consumption in Smartphone, and they showed that because network activities consumes at least 5 times more power than CPU and RAM .

5 Experimental results

5.1 Network simulation of BAN

In this section, we compare the percentage of expired measurements and wireless energy consumption of transmission managers in a small scale, single hop WSN. We evaluate the performance of the state of the art transmission managers for both synchronized and unsynchronized sampling. $Fixed$ transmits all the buffered measurements at every decision epoch (10sec). CL [6], SF [5] and DistTM check data buffer at every decision epoch . The initial decision epoch is 10sec, but next decision epoch is transmission instance when to transmit buffered measurements. Cas [4] calculates the time interval between transmissions based on the node distance to a sink. Since there is only one hop, it transmits measurements every 19.334sec.

Results with synchronized sampling: In this scenario, 10 different applications in a source node has uniformly distributed sampling interval between 1 and 10 sec, and keep the interval to generate measurement. Fig. 13 and Fig. 14 show the percentage of expired measurements and energy consumption with different maximum time constraints. While we increase the maximum time constraints to 90 sec, $Fixed$ and Cas consume same amount of energy because of fixed transmission intervals. Cas consumes less energy than $Fixed$ because Cas has longer time interval (19.3334 sec) between communications than $Fixed$ (10sec). CL adjusts buffer limit from 2 to 26 for different maximum time constraints, and transmits measurements if buffer length reaches the limit. When the maximum time constraint is larger than or equal to 60 sec, CL consumes the least energy, but expire at least 30% of measurements that expire because the buffer length does not reach the transmission threshold. This shows a tradeoff between expiration rate and energy, and the CL algorithm chooses to favor the latter metric. SF uses time constraints of a measurement, transmission and reception power to calculate the threshold and the importance of the measurement. When the importance is larger than the threshold, SF transmits the measurements. Otherwise, it discards them. However, given varying maximum time constraint ranging from 15 to 90sec, all of measurements have higher importance than the thresholds. Thus, SF expires between 0% and 18.13% of measurements for different maximum time constraints. SF achieves energy saving by discarding some portion of measurements, but remaining messages have to be transmitted at every decision epoch, so it consumes the same amount of energy as $Fixed$. DistTM dynamically adjusts the transmission instance based on measurements' time constraints. As the maximum time constraint is increased to 90sec, DistTM consumes on average 1.6 times less energy than other algorithm, on top of which DistTM has on average 5.4% more measurements that expire (except for CL which does not deliver messages in timely fashion).

Results with unsynchronized sampling: Unlike synchronized sampling, each node generates measurements based on a varying sampling interval which is randomly selected between 1 to 10 sec. Fig. 15 and Fig. 16 describe the percentage of expired measurements and energy consumption.

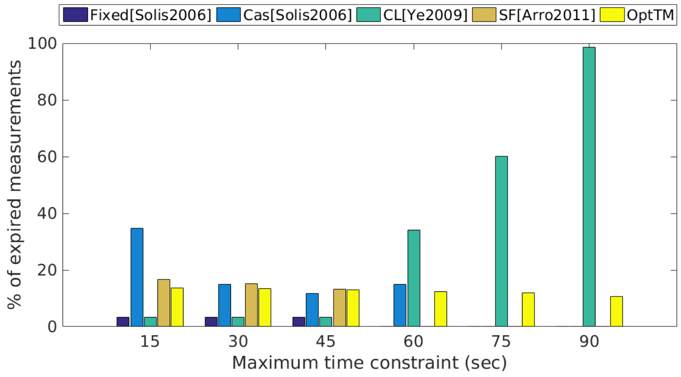


Fig. 15. Percentage of expired measurements

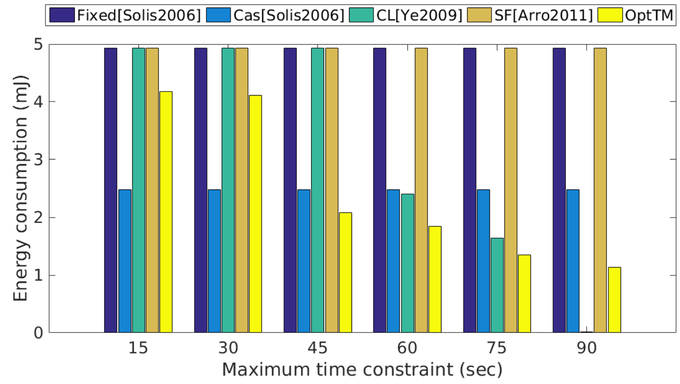


Fig. 16. Energy consumption (mJ)

BAN with random sampling interval

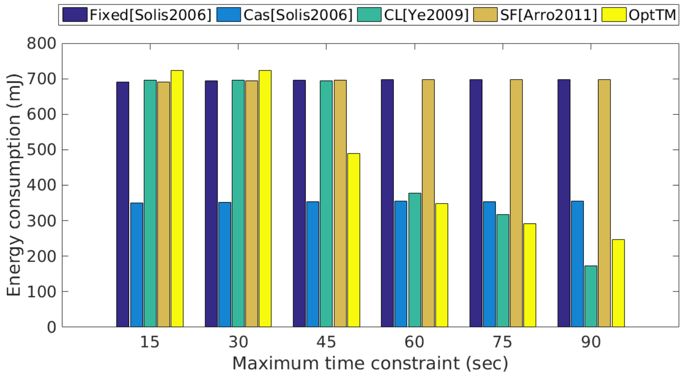


Fig. 17. Percentage of expired measurements

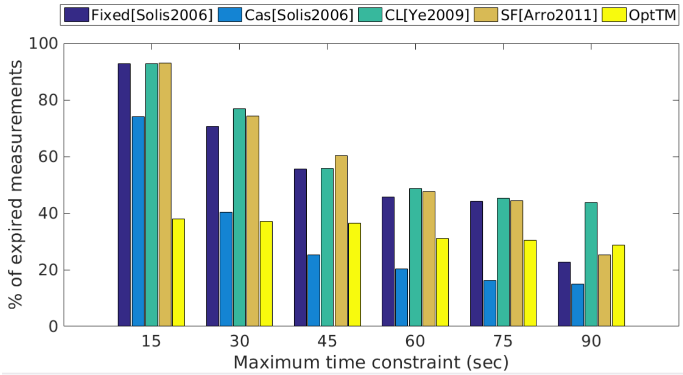


Fig. 18. Energy consumption (mJ)

Linear WSN

$Fixed$ has at most 3.3% of expired measurements. Cas and SF expire at most 34.8% and 16.68% of measurements, but they consume on average 1.9 time more energy than DistTM. While we increase the heterogeneity of time constraints, other algorithms consume on average 100% more energy compared to DistTM, except for CL which does not meet the QoS requirements. DisTM has on average 5.1% more measurements that expire except for CL which does not deliver messages in timely fashion.

5.2 Network simulation of a linear WSN

We compare the performance of transmission managers in terms of energy consumption and the percentage of expired measurements when a set of 10 linearly connected nodes sample data and communicate findings to a single sink on the shore. Each node constructs a routing path to the sink with table driven routing protocol, DSDV. Cas determines transmission instance based on a node distance from the sink. More specifically, farther nodes have shorter time intervals between communications than nodes closer to the sink.

Results with synchronized sampling method: In this scenario, applications in a source node has sampling interval beween 1 and 10 sec, and keep the interval to generates measurements. Fig. 17 and Fig. 18 show the percentage of measurements that expire and the energy consumption. DistTM adjusts transmission instance based on time constraints of buffered measurements. Thus, the state of the art algorithms consume on average 36% more energy than DistTM, while we expire on average 18% less measurements than than all the approaches. $Fixed$, Cas, SF periodically transmit buffered measurements without considering maximum time constraints. These strict approaches result in higher energy consumption as compared to DistTM. While we increase the heterogeneity of time constraints, $Fixed$, Cas and SF decreases the number of measurements that expire because they can deliver more measurements in timely fashion when we extend the maximum time constraints to 90 sec. However, they also consume more energy than DistTM in order to transmit measurements. As we increase the the maximum time constraint to 90sec, Cas expires on average 0.9% more measurements while consuming on average 10% more energy than DistTM.

Results with unsynchronized sampling method: We further investigate the performance of transmission managers in a linearly constructed WSNs when all nodes use a uniformly distributed sampling interval between 1 and 10 sec. Fig. 19 and Fig. 20 show the percentage of measurements that expire and the normalized energy consumption for different maximum time constraints. In linear WSN, approaches using static transmission instance such as $Fixed$ doesn't adjust the transmission instance based on maximum time constraints, so it consumes more energy and have higher percentage of expired measurements compared to DistTM. Cas consumes on average 8% less energy than DistTM while expiring 6% more measurements. When we increase the maximum time constraints to 90 sec, DistTM dynamically adjusts transmission time between communications, so it decreases the energy consumption by decreasing the number of transmission events. DistTM is surpassed in

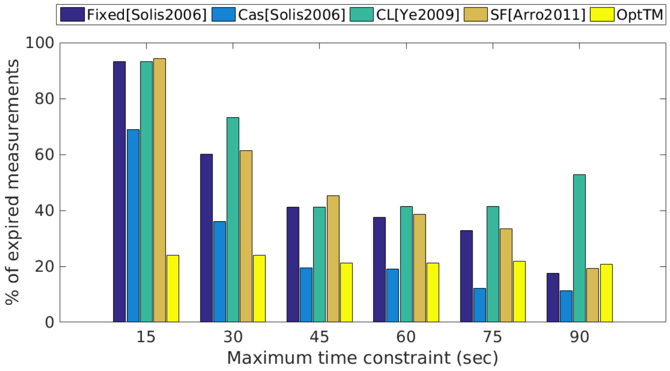


Fig. 19. Percentage of expired measurements

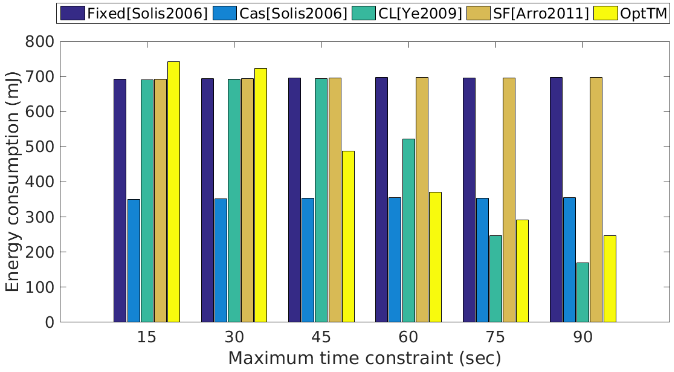


Fig. 20. Energy consumption (mJ)

Linear WSN with random sampling interval

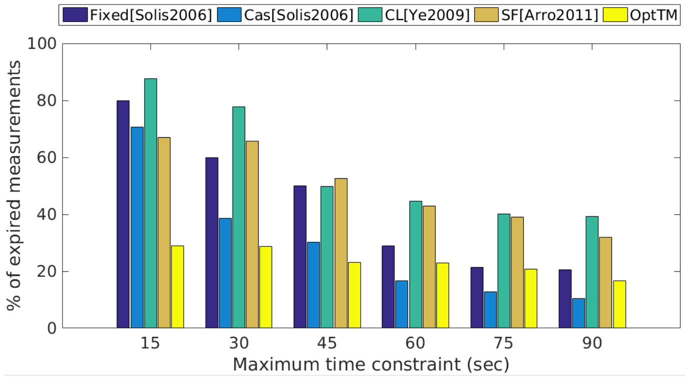


Fig. 21. Percentage of expired measurements

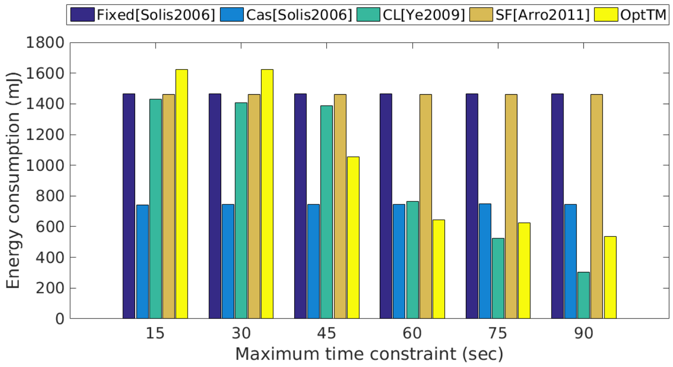


Fig. 22. Energy consumption (mJ)

Grid WSN (20 nodes)

terms of energy saving by an average of 71% while having on average 23% less expired measurements compared to other approaches.

5.3 Network simulation of a grid WSN

In this scenario, we evaluate the performance of transmission managers in a grid topology based WSN. We evaluate the performance of the state of the art transmission managers for both synchronized and unsynchronized samplings. As described in Fig. 12, WSN is constructed with 19 nodes and 1 sink. All nodes except for the sink construct routing paths with table-driven routing protocol, DSDV, and forward buffered measurements to the next hop nodes following the established routing path. We study the effects of network size in Sec. 5.4.

Results with synchronized sampling method: Applications in each node determine sampling interval which is uniformly distributed between 1 and 10 sec, then they periodically generate measurements with the interval. Fig. 21 and Fig. 22 show the percentage of expired measurements and the energy consumption. $Fixed$, Cas and SF periodically transmit measurements without considering maximum time constraints at fixed time instances, so they have less measurements that expire while we increase the maximum time constraint to 90 sec. DistTM determines time interval between transmissions based on measurements' time constraints, so the other approaches consume on average 34% more energy with an average of 22% more expired measurements.

Results with unsynchronized sampling method: In this section, we evaluate the performance of transmission managers when all nodes have sampling interval that is uniformly selected from 1 to 10 sec. Fig. 23 and Fig. 24 show the percentage of expired measurements and the normalized energy consumption for different maximum time constraints. $Fixed$ and SF do not adjust their time intervals between transmissions, based on time constraints of measurements. Unlike their approaches, DistTM dynamically finds the transmission time based on measurements' time constraints as described in Algorithm 1 and Algorithm 2 at each decision epoch. DistTM decreases the energy consumption when we increase the maximum time constraints to 90 sec. $Fixed$, Cas and SF have higher percentage of expired measurements under the same condition, but on average they consume 71% more energy than DistTM. The percentage of measurements that expire under these protocols is on average 19% more than DistTM.

5.4 Results for different sizes of WSNs

In this section, the change the network size ranging from 2 to 20 nodes with synchronized sampling method: single-hop WSN=2, linear WSN=10, and grid WSNs=20.

Fig. 25 and Fig. 26 show the log scaled energy consumption for different network sizes when maximum time constraints are 15 sec and 60 sec. The number of nodes in WSN is shown on x-axis ranging from 2 to 20. Bigger networks generate more measurements and traffic. Thus, all approaches consume more energy with larger network sizes. DistTM dynamically adjusts the transmission time

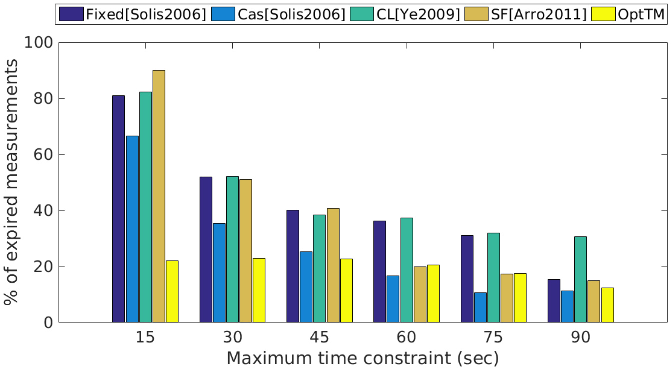


Fig. 23. Percentage of expired measurements

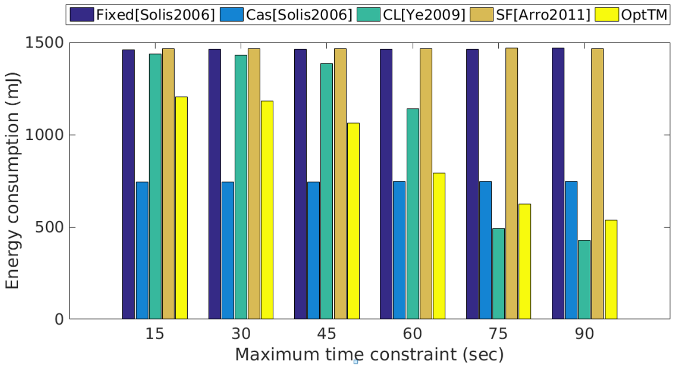


Fig. 24. Energy consumption (mJ)

Grid WSN with random sampling interval (20 nodes)

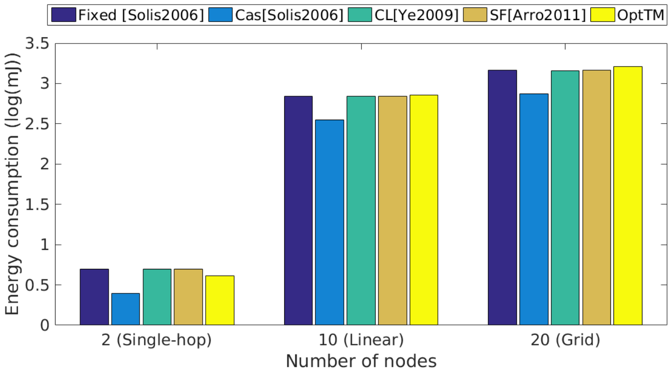


Fig. 25. Maximum time constraint: 15 sec

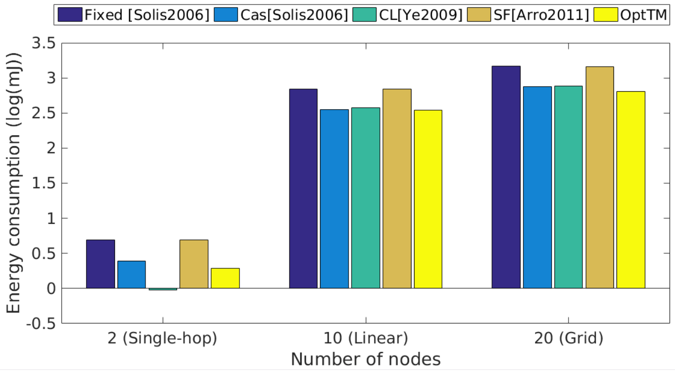


Fig. 26. Maximum time constraint: 60 sec

Energy consumption (log(mJ))

instance based on the measurements' time constraints, so it decreases energy consumption with greater maximum time constraints (ref. Fig. 26). When maximum time constraint is 15 sec, The other approaches consume on average 4.5% more energy compared to DistTM and expire on average 2.2% less measurements. (ref. Fig. 25). When maximum time constraint is 60 sec, the average energy consumption of the other solutions is 73% more than DistTM, with on average 2.6% less expired measurements for different network sizes (ref. Fig. 26 and Fig. 28).

5.5 Comparison between Cas and OptTM

As we can observe from previous results, DistTM consumes less energy than the state of the art approaches except for Cas when maximum time constraint is less than 45sec for both synchronized and unsynchronized sampling methods. When maximum time constraint decrease to 15sec, Cas consumes more energy than DistTM. Thus, in this section, we study their relationships for different maximum time constraints, network sizes and sampling methods. Fig. 29 and Fig. 30 describe the trade-off between the percentage of expired measurements and energy consumption that has been normalized to Cas. Numbers in parenthesis denote network sizes: single-hop WSN=2, linear WSN=10, grid WSNs=20. Numbers on data points denote maximum time constraints. While we increase the maximum time constraints to 90 sec, Cas consumes the same amount of energy because Cas does not adjust its transmission instance based on the time constraints of buffered measurements. However, DistTM decreases the energy consumption for higher maximum time constraints. In BAN, Cas expires on average 0.6% less measurements, and consumes on average 1.3 times more energy than DistTM for different time constraints and sampling methods. In linear WSN, DistTM consumes on average 0.89 times less energy than Cas while having on average1.7

5.6 Overhead

In this section, we evaluate the computational overhead of transmission managers. We set 90 sec as the maximum time constraint of measurements because it causes the worst case computation and space complexity of DistTM. We measure elapsed time of DistTM on a low-power and small-scale embedded device, Raspberry PI2 [18] (1GHz CPU and 1 GB main memory), and compare the elapsed time with the simplest approach, $Fixed$. 100 measurements are placed into the data buffer. DistTM determines the optimal transmission instance based on time constraints. We repeat experiments 100 times. DistTM spends on average 0.4msec more time than $Fixed$ resulting in 8% computation overhead. This shows that DistTM can operate on low-power embedded devices with minimum overhead while it significantly decreases both energy consumption and the number of expired measurements. This result is a natural reflection of the $O(N)$ complexity of our algorithm.

The computation time is also directly related to the decision epoch selection, as a smaller epoch would result in more executions of the algorithm. There is no need for the decision epoch to be smaller than the smallest time constraint of the applications, as it would result in waste

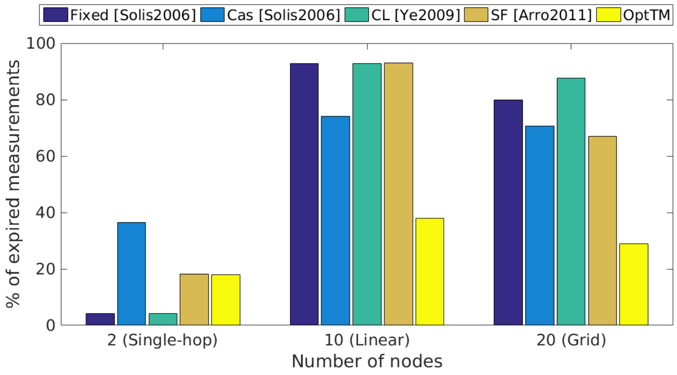


Fig. 27. Maximum time constraint: 15 sec

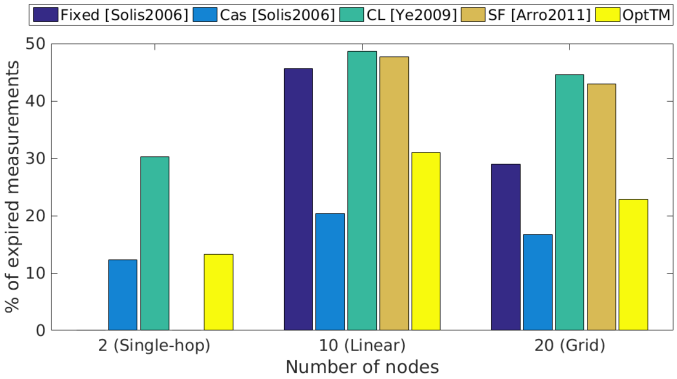


Fig. 28. Maximum time constraint: 60 sec

Percentage of expired measurements

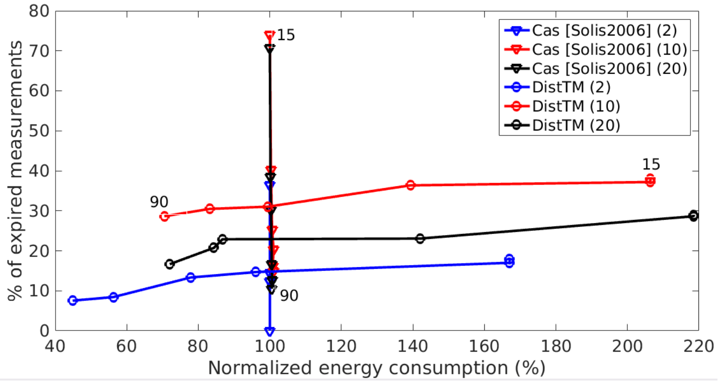


Fig. 29. Synchronized.

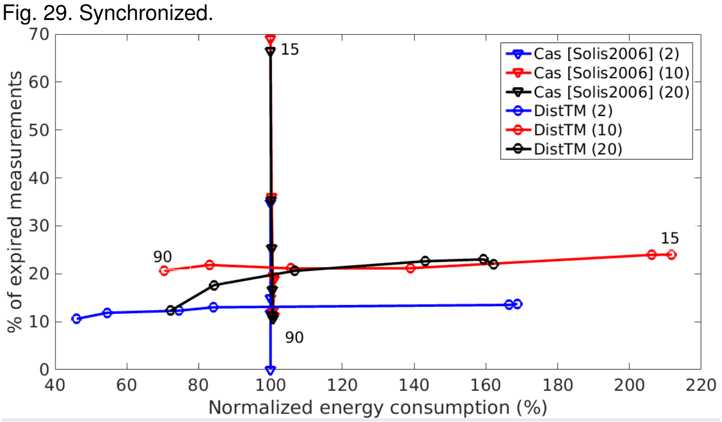


Fig. 30. Unsynchronized.

Trade-off between normalized energy consumption and the percentage of expired measurements

of computation. Thus, the computation is upper bounded by $\max(E(D)) \times N$.

6 Conclusion

In this paper, we propose an optimal transmission manager solution (OptTM) in WSNs where multiple applications operate with different time constraints. OptTM determines the optimal transmission instance based on time constraints of both generated and received measurements in a single hop. We mathematically prove the existence of the optimal transmission instance. Distributed transmission manager (DistTM) evolves OptTM to operate in multi-hop WSNs. DistTM explicitly considers the relationship between end-to-end time constraints and the distance to sink unlike other approaches. We implement both DistTM and OptTM in ns3 simulator, and compare their performances with other state of the art approaches in terms of energy consumption and the number of expired measurements. We consider three different network topologies, single hop, linear and grid, and vary maximum time constraints from 15 sec to 90 sec. For all configurations, the state of the art approaches consume average 75% more energy than our solution, with our algorithm expiring on average 12% less measurements than the other solutions.

use section* for acknowledgment

Acknowledgments

This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA; and in part by NSF grants #1344153 and CNS-1446912.

References

[1] X. Wang, X. Wang, L. Liu, and G. Xing, "Dutycon: A dynamic duty-cycle control approach to end-to-end delay guarantees in wireless sensor networks," vol. 9, no. 4. New York, NY, USA: ACM, Jul. 2013, pp. 42:1–42:33.

[2] Q. Wang, P. Fan, D. Wu, and K. Ben Letaief, "End-to-end delay constrained routing and scheduling for wireless sensor networks," in *IEEE Communications (ICC)*, June 2011, pp. 1–5.

[3] ns3. (2015) Discrete-event network simulator. http://www.nsnam.org/.

[4] I. Solis and K. Obraczka, "In network aggregation trade offs for data collection in wireless sensor networks," in *International Journal of Sensor Networks*, vol. 1, no. 3/4, Inderscience Publishers, Geneva, SWITZERLAND, Jan. 2006, pp. 200–212.

[5] R. Arroyo-Valles, A. Marques, and J. Cid-Sueiro, "Optimal selective forwarding for energy saving in wireless sensor networks," vol. 10, no. 1, January 2011, pp. 164–175.

[6] Z. Ye, A. Abouzeid, and J. Ai, "Optimal stochastic policies for distributed data aggregation in wireless sensor networks," vol. 17, no. 5, Oct 2009, pp. 1494–1507.

[7] P. Aghera, J. Yang, P. Zappi, D. Krishnaswamy, A. K. Coskun, and T. S. Rosing, "Energy management in wireless mobile systems using dynamic task assignment," in *Journal of Low Power Electronics*, vol. 9, no. 2, Aug 2013, pp. 198–217.

[8] N. Nikzad, J. Yang, P. Zappi, T. Rosing, and D. Krishnaswamy, "Model-driven adaptive wireless sensing for environmental healthcare feedback systems," in *ICC, IEEE International Conference on*, June 2012, pp. 3439–3444.

[9] R. Arroyo-Valles, A. Marques, and J. Cid-Sueiro, "Optimal selective transmission under energy constraints in sensor networks," vol. 8, no. 11, Nov 2009, pp. 1524–1538.

[10] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[11] T. Muetze, P. Stuedi, F. Kuhn, and G. Alonso, "Understanding radio irregularity in wireless networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, SECON, 5th Annual IEEE Communications Society Conference on*, June 2008, pp. 82–90.

[12] G. He. (May 06, 2002) Destination-sequenced distance vector (dsdv) protocol.

[13] CC2630. Zigbee wireless mcu. http://www.ti.com/product/CC2630.

[14] J. Yang, R. T.S, and S. Tilak, "Leveraging application context for efficient sensing," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, April 2014, pp. 1–6.

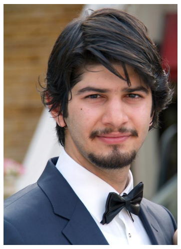
[15] CCIMIS-WSN. (2013) California irrigation management information system. www.cimis.water.ca.gov.

[16] WXT520. Weather station. http://www.vaisala.com/en/products/multiweathersensors.

[17] A. Carroll and G. Heiser, "An analysis of power consumption in a smartphone," in *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, ser. USENIXATC'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 21–21. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855840.1855861

[18] (2015) Raspberry pi2. https://www.raspberrypi.org.

Jinseok Yang received Ph.D degree from University of California, San Diego in 2016. His dissertation focused on energy efficienty data aggregation in wireless sensor networks. He obtained his M.Sc. degrees in Electrical and Computer Engineering from University of California, San Diego in 2012. Dr. Yang is currently working as a Research Engineer at LG Advanced Research lab. His research interestes includes machine learning and context-aware smartphone optimizations.



Alper Sinan Akyurek is a PhD Candidate in Electrical and Computer Engineering at UCSD. His current work is on control and optimization of energy efficiency in the smart grid. He obtained his M.Sc. and B.Sc. degrees in Electrical and Electronics Engineering from Middle East Technical University in 2011 and 2008, respectively. Prior to PhD, he worked as a Senior Design Engineer on Wireless Networks at Aselsan, Turkey.

Sameer Tilak is a Sr. Data Scientist with the Medical Informatics group at Kaiser Permanente where he leads the big data initiative. Before joining Kaiser Permanente he was an Associate Research Scientist at the University of California at San Diego (UCSD) where he was PI and Co-PI on several projects funded by NSF and private foundations. He has extensive experience in the areas of sensor networks, big data technologies and their applications in building real-world large-scale applications. He received his Ph.D. in computer science from SUNY Binghamton and M.S. in computer science from the University of Rochester and SUNY Binghamton.

Tajana Simunic Rosing is a Professor, a holder of the Fratamico Endowed Chair, and a director of System Energy Efficiency Lab at UCSD. She is currently heading the effort in SmartCities as a part of DARPA and industry funded TerraSwarm center. During 2009-2012 she led the energy efficient datacenters theme as a part of the MuSyC center. Her research interests are energy efficient computing, embedded and distributed systems. Prior to this she was a full time researcher at HP Labs while being leading research part-time at Stanford University. She finished her PhD in 2001 at Stanford University, concurrently with finishing her Masters in Engineering Management. Her PhD topic was Dynamic Management of Power Consumption. Prior to pursuing PhD, she worked as a Senior Design Engineer at Altera Corporation.