Wireless Pers Commun (2017) 97:4967–4980
https://doi.org/10.1007/s11277-017-4760-8

Effect of Clock Skew in Event Driven, Delay Constrained Heterogeneous WSN with Anycast

Debanjan Sadhukhan[1] · **Seela Veerabhadreswara Rao**[1]

Published online: 12 August 2017

Abstract In time critical event driven data gathering, anycasting technique is considered as one of the best methods that minimizes delay and maximizes lifetime. Moreover, asynchronous sleep/wake scheduling techniques like anycasting, is energy efficient compared to synchronous sleep/wake scheduling in rare event detection. However, asynchronous techniques incur additional delay during the event reporting as a result of the clock skew which may violate the delay constraint for a large number of packets. In this paper, we find the critical wake-up rate to constrain the increase in end-to-end delay, as a result of clock-skew, within given delay constraint ξ, by estimating the expected increase in end-to-end delay using a stochastic approach. We verify our mathematical analysis using Monte-carlo simulation. Further simulation results in network simulator 2.34 (ns2) confirm the effectiveness of our approach.

Keywords Ad-hoc network · Wireless communication · Energy efficiency · Sleep/wake scheduling · Anycasting · Time synchronization · Clock-skew

1 Introduction

In applications like, tsunami, forest fire and seismic event detection, the sensor nodes are deployed in remote areas to detect rare events. In these rare event detection applications, sensors remain idle until an event occurs and once detected the event information needs to be forwarded to base-station within strict delay constraint. Maintaining delay constraint and extending the lifetime are the key objectives for designing a WSN for such applications.

✉ Debanjan Sadhukhan
debanjan@iitg.ernet.in

Seela Veerabhadreswara Rao
svrao@iitg.ernet.in

[1] Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati 781039, India

In event driven data gathering, sleep/wake (s/w) scheduling technique is acknowledged as one of the best methods to save energy, where sensor nodes switch on the communication device only if an event is detected in their vicinity [7]. In synchronized sleep/wake scheduling, sensor nodes exchange synchronization messages periodically or aperiodically [11, 22, 23, 26]. Passing synchronization messages increases overhead and energy consumption, which may be a major dominating factor in a rare event detection application. Whereas in asynchronous s/w scheduling techniques, since sensor nodes wake-up independently, no clock-synchronization is required with the neighboring nodes, which in turn decreases energy consumption [8]. In these strategies, a node intends to send a packet needs to wait until the forwarding nodes wake up, which may increase overall end-to-end delay (e-delay). Anycasting forwarding strategy, each node maintains a set of forwarding nodes, is proposed in the literature to decrease e-delay.

Hardware oscillator is used for clock to implement the real time in the sensor nodes. The rate at which the clock runs is determined by the angular velocity of the hardware oscillator. As a result of imperfect oscillator or some external reasons, the oscillator frequency varies unpredictably, which is known as clock-skew [25]. The clock skew also depends on environmental factors like temperature, pressure, radiation, magnetic fields etc. Generally, the clock skew is measured in terms of parts-per-million (ppm), where 1 ppm is equivalent to drift of 1 µs (µs) in 1 s. The maximum clock skew is generally varies from up to 40 to 100 ppm [25]. As a result of clock-skew, the actual periodic wake-up interval may vary over time, and increases the overall e-delay. Hence, the e-delay constraint (Δ), may not be always satisfied throughout the lifetime.

Moreover, advancement in sensor technology allows the sensors to be heterogeneous, in terms of different maximum clock-skew associated with them. Also note that, when clock-skew is very low (sensors with high precision hardware clock), the effect of clock-skew in overall e-delay becomes negligible. Hence, we assume that a threshold ξ, is given for the amount of e-delay increases, as a result of clock-skew. Considering these limitations in the literature, we address the following problem *What is the critical wake-up rate to constrain the increase in e-delay, as a result of heterogeneous clock-skew present in WSN, within given threshold* ξ? We estimate the increase in e-delay as a result of heterogeneous clock-skew using stochastic analysis. We use this estimation to find the critical wake-up rate for all nodes to constrain the increase in overall e-delay within given threshold ξ. We validated the analysis using Monte-carlo simulation. We also shown the effectiveness of our approach by simulation using network simulator 2.34 (ns2).

The remainder of the paper is organized as follows. The next section reviews the significant contribution in the literature and motivation behind our work. In the Sect. 3 we estimate the expected increase in e-delay due to the clock skew in asynchronous forwarding strategies and use this estimation to devise a strategy that bounds the delay within given constraint. The fourth section validates our expected analysis using Monte Carlo simulation. Section 5 shows the effectiveness of our approach using ns2 simulation over conventional approaches. The last section concludes the paper with pointers to the future work direction.

2 Related Work

In this section we give a brief survey on various techniques for time synchronization protocols followed by asynchronous s/w scheduling protocols.

In order to mitigate the effect of clock-skew, several time synchronization protocols are proposed in the literature. We only discuss the significant contributions in the literature, the detailed survey of these protocols can be found in [9, 21]. In traditional synchronization schemes the sender periodically sends messages containing its current clock value to synchronize the time with the receiver. An extension of this approach is a two-way message passing to estimate the propagation delay in order to increase the accuracy of clock-synchronization [14]. Generally, synchronization error occurs as a result of nondeterminism present in send time, channel access time, propagation time, and receive time. In order to remove this nondeterminism, Elson et al. proposed Receiver-Broadcast Synchronization (RBS) scheme [3], by broadcasting physical layer beacons. The neighbors compare the arrival time of these beacons and calculate phase-offset and clock-skew using the least-square linear regression model. As a result of several rounds of message exchange RBS consumes a significant amount of energy. In Timing-sync Protocol for Sensor Networks (TPSN) [4], nodes exchange only two way time-stamped synchronization messages at medium access control (MAC) layer and calculate the phase offset. It successfully eliminates synchronization error caused by access time and propagation delay, but unable to estimate the clock-skew. In flooding time synchronization protocol [13], Maroti et al. combined TPSN and RBS to estimate the clock-skew using linear regression. In [5], Ganeriwal et al. proposed an energy efficient long term synchronization scheme by transmitting synchronization beacons at periodic intervals to maintain desired bound on synchronization error. The authors used the least square-based scheme to calculate the clock-skew from a set of received beacons. It also adapts to change in clock drift and environmental conditions to achieve application-specific precision with high probability [5]. Similar approach is followed in [3, 5, 20, 24] to calculate expected phase shift and clock-skew among the neighbors.

An asynchronous preamble-sampling based protocol which exploits Low-Power-Listening (LPL), is proposed in [16]. Nodes periodically listen for a short time to decide ongoing transmission. If an event is detected, the sender attaches long preamble before transmitting data. LPL minimizes energy consumption when no event is detected, whereas consumes more energy due to long preambles. In order to minimize the energy consumption for long preamble, short and strobed preambles based strategies are proposed [2], using additional low power radio, which increases sensor cost. In order to reduce the cost of additional low power radio, a similar strategy is proposed without the need of ultra low power radio [19]. The trade-off between energy efficiency and latency associated with waking up the nodes, is also shown in [19]. In these strategies, each node wakes up independently and waits for the next hop node to wake up before transmitting any packet. Hence, it increases e-delay.

In order to minimize this delay, anycasting forwarding schemes are proposed, where each node maintains a set of candidate nodes (closer to base-station) and forwards the packet to the first node wakes up within this forwarding set. Compared to the basic forwarding strategy, where a node maintains a single designated next-hop relaying node, anycasting strategy decreases expected one hop waiting time.

Several anycasting based packet forwarding schemes are discussed in the literature for wireless networks. In order to route a data-packet efficiently, the shortest path anycasting tree is proposed in [6]. Compared to general shortest path tree, in the shortest path anycasting tree, instead of forwarding the packet to a single parent, every node maintains a set of multiple parents, and forwards the data-packet to the first node wakes up within this set. Geographical distance to the sink is exploited in [10, 27], to minimize the e-delay. Hop-count information is also used to minimize the delay along the routing path [1, 17, 18].

Moreover, in [15] the authors used both hop-count and power consumption metrics to reduce the overall cost of forwarding a data-packet from source to the sink, which in turn increases the overall lifetime of the network.

Kim et al. proposed an anycasting forwarding technique, in which neighboring nodes are added to the forwarding set only when they collectively minimize overall expected e-delay [7]. But packets may follow a longer route to the base station. In order to minimize path length, the same authors developed a delay optimal anycasting scheme [8], where nodes do not immediately forward the packet, instead they wait for some time and then opportunistically forward only when expected delay involves for waiting is more.

Motivation None of the anycasting strategies considered the effect of clock-skew in e-delay to the best of our knowledge, that may lead to violate the delay-constraint in time-critical applications. This limitation in the literature motivates us to find the critical wake-up rate that constrains the overall increase in e-delay, as a result of heterogeneous clock-skew within given threshold ξ.

3 Minimizing Delay in Anycasting Forwarding as a Result of Clock-Skew

In this section, first we derive an expression for the expected increase in expected e-delay in delay-optimal anycasting forwarding scheme [8] and use this analysis to find the critical wake-up rate of node i, $r_i'' \forall i$, that constrain the increase in overall e-delay to within given threshold ξ. For the sake of completeness, we briefly discuss the delay-optimal anycasting forwarding policy. Whenever a node i needs to send a packet, it sends a beacon signal of duration t_B, followed by an ID signal of duration t_C, and listens for acknowledgment of duration t_A as shown in Fig. 1, where $t_I = t_A + t_B + t_C$ denotes the duration of an epoch. One can refer to Table 1 for the details of the notations used in this work. Every node j follows an optimal wake-up rate r_j and forwarding policy to satisfy the given delay constraint. If j wakes up and hears hth beacon and ID signal, then node j can either choose to receive the packet by sending its ID during t_A, or go back to sleep and wakes up at next interval. This interval is determined by the s/w scheduling policy chosen by the node j. Let $f_i(h) = j$ be the forwarding policy of node i at hth beacon signal. If $f_i(h) = j$ and $j \neq i$, then the node i decides to send the packet at any node with rank higher or equal to j, at beacon signal h. Whereas, if $i = j$, then the node i waits for the next beacon signal.

3.1 Expected Increase in End-to-End Delay

Let $F_i \subseteq N_i$ be the forwarding set of i, where N_i is the set of neighbors of node i. The conditional probability of any node $j \in F_i$ wakes up at hth beacon signal is

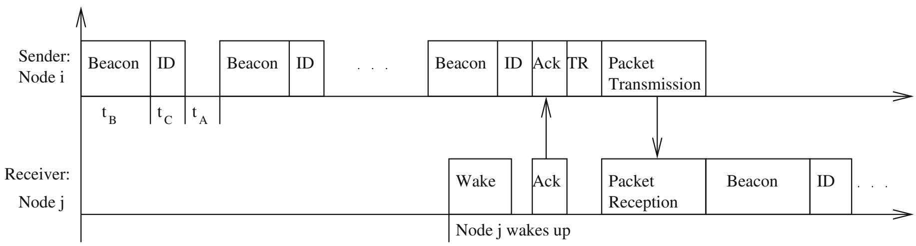


Fig. 1 Example of packet forwarding protocol

Table 1 Shows details of the notations

clk_j	Clock-skew rate of node j
Δ	e-Delay constraint
$d_i(f)$	Expected one-hop delay of node i
$d'_i(f)$	Expected one hop delay with clock-skew
$\delta d_i(f)$	Increase in expected one-hop delay as a result of clock-skew
$\delta D_i(f)$	Increase in expected e-delay as a result of clock-skew
f	Global forwarding vector
f_i	Forwarding policy of node i
$f_i(h)$	Forwarding policy of i at hth beacon
F_i	Forwarding set of node i
h_{max}	Minimum number of beacons for which the packet is surely forwarded
h'_{max}	Minimum number of beacons for which the packet is surely forwarded in presence of clock-skew
N	Set of all nodes in the given network
N_i	Neighbors of node i
$p_{j,h}$	Conditional probability of node $j \in F_i$ wakes up at hth beacon
$p'_{j,h}$	Conditional probability of any node j wakes up at hth beacon with clock-skew clk_j
$P_i(h \| f_i(h) = j)$	Probability of packet is forwarded at hth beacon
$P_i(h \| f_i(h) = j)$	Probability of packet is forwarded at hth beacon in presence of clock-skew
$q_{i,j}$	Probability of a packet is forwarded to node j
$q_{i,j}(h)$	Probability of sending a packet to node j at hth beacon
r_i	Optimal wake-up rate for node i
r'_i	Minimum wake-up rate due to clock-skew clk_i
r''_i	Critical wake-up rate to satisfy delay constraint
ξ	Threshold for increase in e-delay as a result of clock-skew
t_A	Duration of acknowledgment signal
t_B	Duration of beacon signal
t_C	Duration of ID signal
t_I	Duration of an epoch

$$p_{j,h} = \begin{cases} \dfrac{t_I}{1/r_j - (h-1)t_I} & \text{if } h \leq \left\lceil \dfrac{1/r_j}{t_I} \right\rceil, \\ 1 & \text{otherwise,} \end{cases} \tag{1}$$

where $t_I = t_A + t_B + t_C$.

In order to find an expression for expected increase in e-delay as a result of clock skew, we first need to find an expression for expected increase in one hop delay with given forwarding strategy. Let $f_i = \{f_i(1), f_i(2), \ldots, f_i(\infty)\}$ be the forwarding policy of node i. The packet is forwarded at the hth beacon signal, only if the packet is not forwarded during $h-1$ beacons and any node with equal or higher rank of j wakes up at hth beacon signal. The probability of any node with equal or higher rank of j, wakes up at the beacon signal h, is $1 - \prod_{k=1}^{j} (1 - p_{k,h})$. Hence, the probability of packet is forwarded at hth beacon to the next hop node, conditioned on $f_i(h) = j$ is

$$P_i(h|f_i(h)=j) = \left(\prod_{l=1}^{h-1}(1-P_i(l|f_i(l)=q))) \times (1-\prod_{k=1}^{j}(1-p_{k,h})\right). \tag{2}$$

Moreover, if $f_i(h)=j$ for some h such that $i \neq j$, $k \leq j$ and $h > \frac{1/r_k}{t_I}$, then the packet is surely forwarded at hth beacon. The minimum value of h for which the packet is surely forwarded is denoted by h_{max}. Therefore, $\sum_{h=1}^{h_{max}} P_i(h|f_i(h)=j) \geq 1$. The expected one hop delay of the node i, $d_i(\mathbf{f})$, for given global forwarding vector $\mathbf{f}=(f_1,f_2,\ldots,f_N)$, is the sum of the duration W_i, duration of the expected number of beacons it waits, and transmission time. Expected number of beacons the node i waits is given by $\sum_{h=1}^{h_{max}} P_i(h|f_i(h)=j)(h)t_I$. Hence, the expected one hop delay of the node i is given by Eq. 3. The value of h_{max} can be calculated by gradually increasing h and selecting the minimum h that satisfies $\sum_{h=1}^{h_{max}} P_i(h|f_i(h)=j) \geq 1$.

$$\begin{aligned} d_i(\mathbf{f}) &= \sum_{h=1}^{h_{max}} P_i(h|f_i(h)=j)(h)t_I + t_D, \\ &\text{where} \left(\sum_{h=1}^{h_{max}-1} P_i(h|f_i(h)=j)\right) \leq 1 \text{ and} \left(\sum_{h=1}^{h_{max}} P_i(h|f_i(h)=j)\right) > 1 \end{aligned} \tag{3}$$

$$\begin{aligned} d'_i(\mathbf{f}) &= \sum_{h=1}^{h'_{max}} P'_i(h|f_i(h)=j)(h)t_I + t_D, \\ &\text{where} \left(\sum_{h=1}^{h'_{max}-1} P'_i(h|f_i(h)=j)\right) \leq 1 \text{ and} \left(\sum_{h=1}^{h'_{max}} P'_i(h|f_i(h)=j)\right) > 1 \end{aligned} \tag{4}$$

In homogeneous clock-skew, every node is associated with equal maximum clock-skew rate in the network. Whereas, in case of heterogeneous clock-skew present in the network, node j is associated with clock-skew rate upto clk_j. Hence, change in the wake up interval as a result of clock-skew of node j is in $\left[-clk_j \times \frac{1}{r_j}, clk_j \times \frac{1}{r_j}\right]$ and wake-up rate is in range $\left[\frac{r_j}{1+clk_j}, \frac{r_j}{1-clk_j}\right]$. If wake-up rate is greater than or equal to r_j, there is no increase in e-delay. On the other hand, maximum increase in e-delay happen for the wake-up rate $r'_j = \frac{r_j}{1+clk_j}$. The equations corresponding to Eqs. 1 and 2 is given in Eqs. 5 and 6, respectively.

$$p'_{j,h} = \begin{cases} \frac{t_I}{1/r'_j} & \text{if } h \leq \left\lceil \frac{1/r'_j}{t_I} \right\rceil, \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

$$\begin{aligned} P'_i(h|f_i(h)=j) = &\left(\prod_{l=1}^{h-1}(1-P'_i(l|f_i(l)=q))\right) \\ &\times \left(1-\prod_{k=1}^{j}(1-p'_{k,h})\right). \end{aligned} \tag{6}$$

Accordingly we can find h'_{max} for given forwarding set.

Assume forwarding policy for every node remains same although the wake-up rate changes as a result of clock-skew. Hence, increase in expected one hop delay as a result of clock skew is upper bounded by,

$$\delta d_i(\mathbf{f}) = d_i(\mathbf{f}) - d'_i(\mathbf{f}). \tag{7}$$

In optimal anycasting forwarding technique, the packet is forwarded to node j at beacon signal h only if $f_i(h) = k$ and all nodes with higher priority than j are asleep, where $j \leq k$. Hence, the probability of sending the packet to node j at hth beacon is given by

$$q_{i,j}(h) = \begin{cases} \prod_{l=1}^{j-1} \left(1 - p'_{l,h}\right) p'_{j,h}, & \text{if } f_i(h) = k \text{ where } j \leq k, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

The probability of a packet is forwarded to node j within h'_{max} beacon signal is

$$q_{i,j} = \frac{\sum_{h=1}^{\left\lfloor \frac{h'_{max}}{t_I} \right\rfloor} q_{i,j}(h)}{\sum_{k \in F_i} \sum_{h=1}^{\left\lfloor \frac{h'_{max}}{t_I} \right\rfloor} q_{i,k}(h)}. \tag{9}$$

Expected increase in e-delay as a result of clock-skew is the sum of expected increase in one hop delay and expected increase in next hop e-delay. Expected increase in next hop e-delay is the sum of product of the probability of the packet is forwarded to the next hop node and expected increase in the next hop e-delay. Every node i calculates its own expected increase in e-delay as a result of clock skew as

$$\delta D_i(\mathbf{f}) = \left(\sum_{j \in F_i} q_{i,j} \times \delta D_j(\mathbf{f}) \right) + \delta d_i(\mathbf{f}). \tag{10}$$

3.2 Critical Wake-Up Rate

In this subsection, we are interested to calculate the expected increase in e-delay as a result of clock skew $\delta D_i(\mathbf{f})$ for all $i \in N$ and critical wake-up rate r''_i that satisfies given delay constraint. Moreover, we use the expected increase in e-delay as a result of clock skew $\delta D_i(\mathbf{f})$ to find the critical wake-up rate r''_i for each node i. The overall method is given in Algorithm 1, for node i, which calculates the expected increase in e-delay as a result of clock-skew, $\delta D_i(\mathbf{f})$ and if it is a peripheral node then this quantity is forwarded to the base-station. This method also sets new wake-up rate as r''_i for all i only if $\max_i(\delta D_i(\mathbf{f})) > \xi$. If the increase in e-delay as a result of clock-skew is less than the threshold ξ, then the wake-up rate remains unchanged. If the increase is more, then the critical wake-up rate is calculated for every node i to satisfy the given threshold ξ.

The nodes closer to the base station encounter less increase in e-delay compared to the nodes far from the base station. Let k be the node with maximum delay increase as a result of clock-skew. That is, $\delta D_k(\mathbf{f}) \geq \delta D_i(\mathbf{f})$ for all i in N, where N denotes the set of all nodes within the given network. If the packet emitting from node k satisfies ξ, then any other packet is expected to satisfy ξ as well. In order to constrain this delay within ξ, every node wakes up more frequently.

Peripheral nodes broadcast their own expected e-delay increase as a result of clock-skew to the base-station. The base-station waits for some time to receive the expected e-delay increase from the nodes at the periphery of the FoI. If the maximum e-delay increase is more than ξ, the base-station broadcasts a message to revise the wake-up rate of all the nodes in the network. A node is on the periphery of the FoI if no other node's forwarding set contains this node. Since, each node maintains a set of nodes to which it is in their forwarding set, it can determine whether it is a periphery node or not.

The nodes in direct communication range of the base-station set their expected e-delay increase for clock-skew as zero. Every other node i waits to receive the expected e-delay increase from the nodes in its forwarding set to calculate its own expected e-delay increase ($\delta D_i(\mathbf{f})$). Every node calculates its own expected e-delay only when the expected e-delay increase of all the nodes in its forwarding set is received.

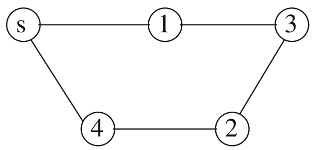
Example 1 We consider a simple network as shown in Fig. 2 [8] to demonstrate our analysis. Let $t_I = 1$ and $t_D = 2$ for all nodes, and $1/r_1 = 1/r_2 = 1/r_3 = 50t_I$ and $1/r_4 = 3t_I$. Node 1, 2, and 4 respectively contains {s,4} and {s} in their corresponding forwarding set. Since the sink is always awake, $f_1(1) = f_4(1) =$s. Moreover, f_1 and f_4 are defined for only first beacon. Whereas $f_2(1) = f_2(2) = \cdots = f_2(50) = 4$ as node 4 is the only forwarding node for 2. Node 3 contains node 1, 2 in its forwarding set and its forwarding strategy is $f_3(1) = f_3(2) = \cdots = f_3(41) = 2$, and $f_3(42) = \cdots = 1$. First the nodes are in direct communication range of base-station, that is 4 and 1 set $\delta D_4(\mathbf{f}) = \delta D_1(\mathbf{f}) = 0$ and forwards these values to node 2 and 3 respectively. When node 2 receives this information from node 4, it calculates $\delta D_2(\mathbf{f}) = 0 + \delta d_2(\mathbf{f}) = 0.000179$. Node 3 calculates $q_{3,2} = 0.44$, and $q_{3,1} = 0.56$, and subsequently $\delta D_3(\mathbf{f}) = 0.44 * 0 + 0.66 * 0.000179 + \delta d_3(\mathbf{f}) = 0.002118$. Let $\xi = 1\,ms$, hence we get $\max_i(\delta D_i(\mathbf{f})) > \xi$. New wake up interval $1/r'_1 = 1/r'_2 = 1/r'_3 = (50 - 50 * 0.0001)t_I = 49.0001\, t_I$ for node 1, 2, 3 and $1/r'_4 = 2.9997t_I$ for node 4.

Algorithm 1: Executed at node i

Output: $D_i(f)$: Increase in expected e-delay as a result of clock-skew and
r''_i: critical wake-up rate to satisfy given constraint

if *Base station* $\in F_i$ **then**
 $\delta D_i(\mathbf{f}) = 0$
else
 if $\delta D_j(\mathbf{f})$ *for all* $j|j \in F_i$ *received* **then**
  $\delta D_i(\mathbf{f}) = \left(\sum_{j \in F_i} q_{i,j} \times \delta D_j(\mathbf{f}) \right) + \delta d_i(\mathbf{f})$
if *there exist a node* j *such that* $i \in F_j$ **then**
 Forwards $\delta D_i(\mathbf{f})$ to all $\{j|i \in F_j\}$
else
 i sends $\delta D_i(\mathbf{f})$ to base station
if *Receives message from base-station to revise wake-up rate* **then**
 Set $r''_i = \frac{r_i}{1 - clk_i}$

Fig. 2 Simple example demonstrates our approach



3.3 Validation of the Analysis

In order to validate the expressions for expected increase in one hop delay (Eq. 7) and e-delay (Eq. 10) we evaluate the corresponding equations numerically, and compare with respective simulation results. For simplicity we consider a simple network shown in Fig. 4. For the simulation results, a Monte Carlo simulation is conducted which is used to estimate the probability from a large number of experiments. Each experiment consists of several tests where in each test a time within wake-up interval when a neighboring node wakes up is picked randomly from uniform distribution. Moreover, the simulation results with 95% confidence level with the error bars are given in Fig. 3.

Expected Increase in One-hop Delay For the Monte Carlo experiment, we select node 0 (see Fig. 4) and calculate maximum increase in one hop delay for different worst clock-skews. We set the wake-up interval 10 s and $m = 10$. We vary (homogeneous) clock-skew rate from 10 to 100 and compare the results obtained from simulation as shown in Fig. 3. Note that, the numerical and simulation results match with no visible error bars. Moreover, increasing clock-skew increases one-hop delay because as clock-skew increases wake-up rate may decrease which may increase expected increase in one-hop delay.

Expected Increase in e-delay In order to verify the expression for increase in expected e-delay, we evaluate Eq. 10 numerically and compare it with that obtained in a simulation. For the Monte Carlo simulation, we deploy 100 nodes uniformly at random in $1000 \times 1000\text{m}^2$ area with 200 m communication range. The sensor nodes follow optimal any-casting forwarding policy [8], with periodic wake-up interval 10 s. The farthest node is expected to have maximum increase in e-delay as a result of clock-skew. Hence, we select the farthest node and generate an event to calculate the maximum increase in e-delay. We repeat the experiment 100 times by generating multiple events and calculate average increase in e-delay for different clock-skew (see Fig. 3). Increasing clock-skew increases expected e-delay because as clock-skew increases expected one-hop delay increases which in turn increases expected e-delay. Although the size of error-bars increases as clock-skew

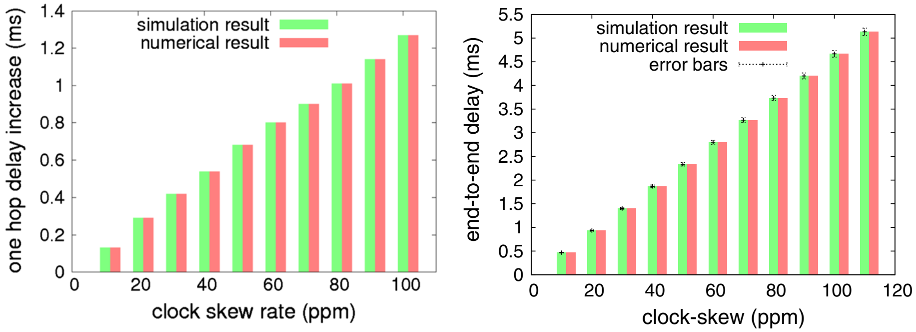


Fig. 3 Validating expected increase in one-hop delay and e-delay

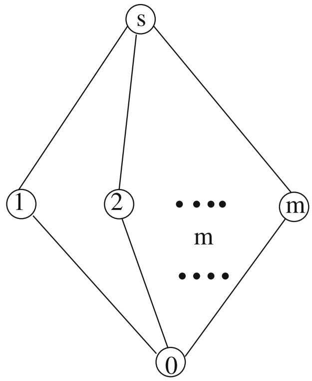


Fig. 4 Validating our approach

increases, the percentage of size of error-bars for e-delay remains same as we increase clock-skew.

4 Simulation Results

We now verify the effectiveness of optimal anycasting forwarding technique [8] with our approach (optimal anycasting [8] with revised wake-up rate) using network simulator 2.34 (ns2). For simulation we deploy 100 nodes in 1000×1000 m^2 area using a random uniform distribution. The sensor nodes calculate periodic wake-up rate and optimal anycasting forwarding policy during the configuration phase, for given delay constraint Δ [8].

We set data-packet size as 8 Byte and control packet size as 3 Byte [25]. We use Mica 2 motes parameters used in [12, 25]. We set receive and idle power to 13.0 mW [25]. Other parameters used in simulation are given in Table 2. We define lifetime of the network as the lifetime of the first node that depletes its energy completely.

In order to show effectiveness, we compare percentage of packets follows delay constraint in optimal s/w scheduling [8] and in our strategy. Moreover, we also compare lifetime of network, control packets overhead, and packet delivery ratio, in optimal s/w scheduling [8] and in our strategy.

Packets follow delay constraint We show the percentage of packets violate delay constraint $(\Delta + \xi)$, for different data generation interval in Fig. 5, with worst clock-skew 100 ppm. Whereas, in Fig. 5 we set wake-up interval 10 s and compare percentage of packets violate delay constraint. Percentage of packets do not follow delay constraint $\Delta + \xi$ increases, as we increase wake up interval or clock skew in conventional approach, after certain interval. Whereas using our approach the delay constraint violated by much lower percentage of packets. This is because, when ξ is fixed, increasing wake-up interval

Table 2 Shows details of the parameters used for ns2-based simulation

Communication range	250 m
Data rate	19.2 kbps
Transmission/receiving power	13.0 mW
Idle power	13.0 mW
Data packet length	8 bytes
Control packet length	2 bytes

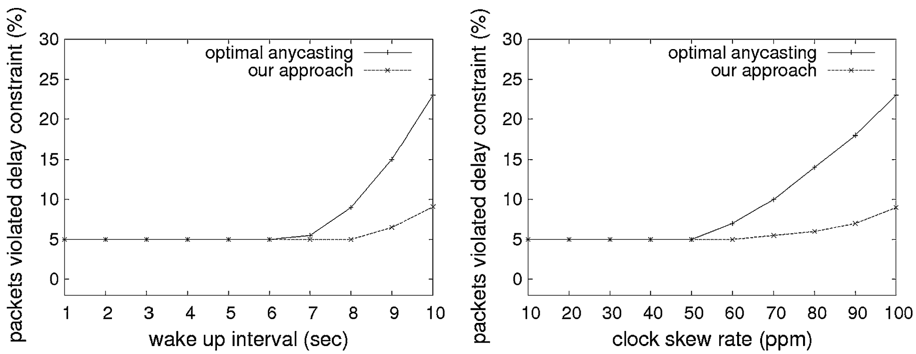


Fig. 5 Monitoring end-to-end delay while varying wake-up interval and clock-skew, and with fixed $\xi = 1$ ms

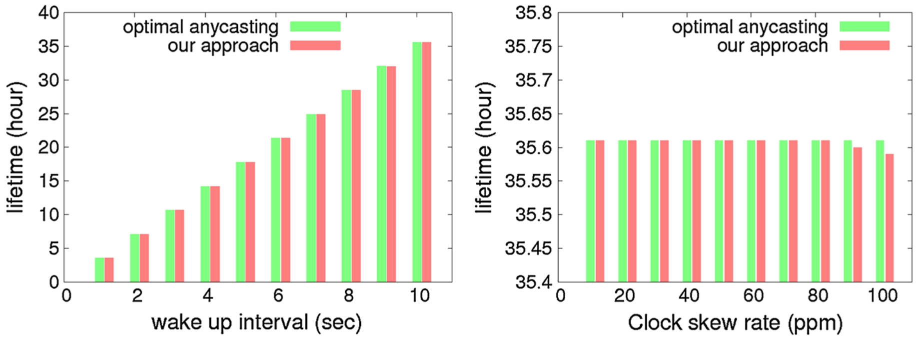


Fig. 6 Monitoring lifetime while varying wake-up interval and clock-skew, and with fixed $\xi = 1$ ms

or clock-skew, increases expected one-hop delay increase as a result of clock-skew, which in turn increases percentage of packets violated delay constraint $\Delta + \xi$.

Lifetime of network It can be observed from Fig. 6 that when wake-up interval increases the overall lifetime in our approach as well as conventional approach increases as well. This is because increasing wake-up interval, increases lifetime of every node, which increases the overall network lifetime. Moreover, when wake-up interval is set 10 s across the network, the lifetime decreases very marginally in our approach when clock-skew increases (Fig. 6). Although it is not clearly visible in the graph, with 100 ppm clock skew rate and 10 s wake-up interval, the lifetime decreases by approximately 0.1%. This is because, in our approach we increase the wake-up rate to satisfy the delay constraint $\Delta + \xi$, which decreases the overall lifetime of the network.

Packet delivery ratio Figure 7 shows percentage of data packets delivered when an event is detected, while varying clock-skew with wake-up interval 10 s. It can be noted that, the percentage of packet delivery ratio in both our approach and conventional approach remains same (≈100% when rare events occur).

Control packets overhead Amount of control packets exchanged for our approach depends only on number of nodes and network topology and forwarding policy. Wake-up interval, clock-skew does not affect the number of control packets. In Fig. 7, we show that percentage of control packets in our approach remains almost same when wake up interval increases.

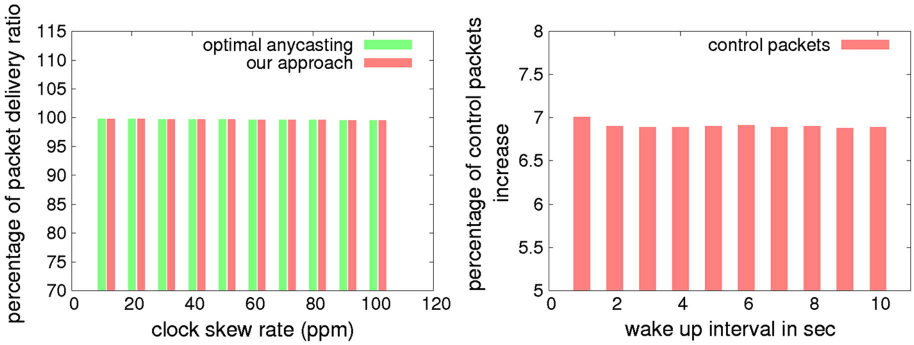


Fig. 7 Monitoring percentage of packet delivery ratio and control packets overhead

5 Conclusion and Future Work

Asynchronous sleep/wake scheduling techniques minimizes delay and maximizes lifetime in event-driven data-gathering. But in presence of clock-skew, these techniques incur additional delay during event reporting time. Our results show that for a larger wake-up rate and higher clock-skew rate almost 23% of the total packets do not follow the given delay constraint. In this paper we provided a solution to constrain this additional delay incurs as a result of clock skew within given bound ξ. Simulation results show that our method reduces the number of packets violate delay constraint to 7–8% in extreme conditions. Although this method constrains the additional delay but it is not an optimal strategy in terms of maximizing the lifetime. Future work can be done to provide an optimal solution that decreases this delay increase and maximizes the lifetime as well.

References

1. Biswas, S., & Morris, R. (2005). ExOR: Opportunistic multi-hop routing for wireless networks. In *ACM SIGCOMM Computer Communication Review* (Vol. 35, No. 4, pp. 133–144). ACM.
2. Buettner, M., Yee, G. V., Anderson, E., & Han, R. (2006). X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the 4th international conference on embedded networked sensor systems* (pp. 307–320). ACM.
3. Elson, J., Girod, L., & Estrin, D. (2002). Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Operating Systems Review*, *36*(SI), 147–163.
4. Ganeriwal, S., Kumar, R., & Srivastava, M. B. (2003). Timing-sync protocol for sensor networks. In *Proceedings of the 1st international conference on embedded networked sensor systems* (pp. 138–149). ACM.
5. Ganeriwal, S., Ganesan, D., Shim, H., Tsiatsis, V., & Srivastava, M. B. (2005). Estimating clock uncertainty for efficient duty-cycling in sensor networks. In *Proceedings of the 3rd international conference on embedded networked sensor systems* (pp. 130–141). ACM.
6. Hu, W., Bulusu, N., & Jha, S. (2005). A communication paradigm for hybrid sensor/actuator networks*. *International Journal of Wireless Information Networks*, *12*(1), 47–59.
7. Kim, J., Lin, X., Shroff, N. B., & Sinha, P. (2010). Minimizing delay and maximizing lifetime for wireless sensor networks with anycast. *IEEE/ACM Transactions on Networking (TON)*, *18*(2), 515–528.
8. Kim, J., Lin, X., & Shroff, N. B. (2011). Optimal anycast technique for delay-sensitive energy-constrained asynchronous sensor networks. *IEEE/ACM Transactions on Networking (TON)*, *19*(2), 484–497.
9. Lasassmeh, S. M., & Conrad, J. M. (2010). Time synchronization in wireless sensor networks: A survey. In *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)* (pp. 242–245). IEEE.

10. Liu, S., Fan, K. W., & Sinha, P. (2009). CMAC: An energy-efficient MAC layer protocol using convergent packet forwarding for wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, *5*(4), 29.
11. Lu, G., Krishnamachari, B., & Raghavendra, C. S. (2004). An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks. In *Proceedings of 18th international parallel and distributed processing symposium* (p. 224). IEEE.
12. Ma, J., Lou, W., Wu, Y., Li, X. Y., & Chen, G. (2009). Energy efficient TDMA sleep scheduling in wireless sensor networks. In *INFOCOM 2009, IEEE* (pp. 630–638). IEEE.
13. Marti, M., Kusy, B., Simon, G., & Ldeczi, A. (2004). The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 39–49). ACM.
14. Mills, D. L. (1991). Internet time synchronization: The network time protocol. *IEEE Transactions on Communications*, *39*(10), 1482–1493.
15. Mitton, N., Simplot-Ryl, D., & Stojmenovic, I. (2009). Guaranteed delivery for geographical anycasting in wireless multi-sink sensor and sensor-actor networks. In *INFOCOM 2009, IEEE*.
16. Polastre, J., Hill, J., & Culler, D. (2004). Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 95–107). ACM.
17. Rossi, M., & Zorzi, M. (2007). Integrated cost-based MAC and routing techniques for hop count forwarding in wireless sensor networks. *IEEE Transactions on Mobile Computing*, *6*(4), 434–448.
18. Rossi, M., Zorzi, M., & Rao, R. R. (2008). Statistically assisted routing algorithms (SARA) for hop count based forwarding in wireless sensor networks. *Wireless Networks*, *14*(1), 55–70.
19. Schurgers, C., Tsiatsis, V., & Srivastava, M. B. (2002). STEM: Topology management for energy efficient sensor networks. In *Aerospace conference proceedings* (Vol. 3, pp. 3–1099). IEEE.
20. Sichitiu, M. L., & Veerarittiphan, C. (2003). Simple, accurate time synchronization for wireless sensor networks. In *Wireless communications and networking, WCNC 2003* (Vol. 2, pp. 1266–1273). IEEE.
21. Sundararaman, B., Buy, U., & Kshemkalyani, A. D. (2005). Clock synchronization for wireless sensor networks: A survey. *Ad Hoc Networks*, *3*(3), 281–323.
22. Tseng, Y. C., Hsu, C. S., & Hsieh, T. Y. (2003). Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. *Computer Networks*, *43*(3), 317–337.
23. Van Dam, T., & Langendoen, K. (2003). An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems* (pp. 171–180). ACM.
24. Van Greunen, J., & Rabaey, J. (2003). Lightweight time synchronization for sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications* (pp. 11–19). ACM.
25. Wu, Y., Fahmy, S., & Shroff, N. B. (2009). Optimal sleep/wake scheduling for time-synchronized sensor networks with QoS guarantees. *IEEE/ACM Transactions on Networking*, *17*(5), 1508–1521.
26. Ye, W., Heidemann, J., & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Transactions on Networking*, *12*(3), 493–506.
27. Zorzi, M., & Rao, R. R. (2003). Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Energy and latency performance. *IEEE Transactions on Mobile Computing*, *2*(4), 349–365.

Debanjan Sadhukhan received the B.Tech. degree from West Bengal University of Technology, West Bengal, India. He is currently working toward the Ph.D. degree with Indian Institute of Technology Guwahati, India. His research interests include wireless sensor networks, wireless ad hoc networks, wireless communication, and distributed algorithms.

Seela Veerabhadreswara Rao received his Ph.D. in Computer Science and Engineering from the Indian Institute of Technology Kanpur, in 1999. Presently, he is a Professor in the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, India. His research interests include wireless sensor networks, wireless ad hoc networks, data center networks, cloud computing, software defined networks and computational geometry.