On Forwarding Protocols in Linear Topology Wake-up Wireless Sensor Networks

Jian Wang[1(✉)], Xiaolin Xu[1], Xiaoming Hu[1], and Wei Wayne Li[2]

[1] Shanghai Polytechnic University, Shanghai 201209, China
{wangjian,xlxu,xmhu}@sspu.edu.cn

[2] Texas Southern University, Houston, TX 77004, USA
wei.li@tsu.edu

Abstract. Wake-up radio (WuR) is a kind of ultra-low power transceiver that consumes energy at 1000 times lower in magnitude when compared to the main radio in traditional wireless sensors. When incorporated, traditional wireless sensor networks are possible to improve energy efficiency and packet delay simultaneously by mitigating idle listening and overhearing issues. In recent years, many works have designed and evaluated the performance of MAC protocols in WuR-enabled yet single-hop (i.e. star-shaped) wireless sensor networks. This paper moves to a multi-hop network and focuses on linear topology WuR-enabled WSNs. It makes practical sense as large-scale WSN topologies could be decomposed into multiple linear topologies. Based on WuR inherent characteristics and also signal interferences among adjacent sensors, we introduce some interesting design ideas and describe our proposed MAC protocol in detail. Analytical results on expected radio-on time of intermediate sensors when waken up are derived. Also numerical results based on normalized per-hop energy and delay ratios show the effectiveness of our protocol. It may serve as an interesting basis for potential researches into more realistically large-scale WuR-enabled WSNs.

Keywords: Wireless sensor network · Wake-up radio
Linear topology · MAC protocol · Energy efficiency

1 Introduction

Wireless sensor networks have found varied applications in environment monitoring, battlefield surveillance, industrial control, heath-care, and smart grid. They typically consist of many small low-cost wireless sensors. These sensors are in general battery-powered, bandwidth-constrained, and memory-limited. When deployed in the field, individual sensors continuously sense their surroundings.

This work is supported in part by Key Disciplines of Computer Science and Technology of Shanghai Polytechnic University under Grant No. XXKZD1604 and US National Science Foundation under Grant No. 1137732.

As sensors have limited wireless communication ranges and often no communication infrastructure is available in the field, the sensors have to forward the generated data packets to the sink in a hop-by-hop manner by themselves.

The wireless sensor networks are desirable to operate unattendedly in months or even years. Thus it is necessary to conserve the limited energy when forwarding data packets. Typically duty-cycling [2] is an effective approach yet causing a large end-to-end packet delay, whereas always-on incurs considerable idle listening and overhearing issues. In recent years, ultra-low power wake-up radio (WuR) transceivers [12] have been designed and subsequently manufactured that have an energy consumption rate of around 1000 times smaller in magnitude [11] when compared to traditional radio transceivers, i.e. μW versus mW [9]. After they are incorporated into traditional sensors, it is possible to keep main radio transceivers asleep as long as possible and wake up them whenever needed via wake-up beacon packets. As such, dilemma of energy efficiency and end-to-end packet delay is mitigated, if not completely resolved, although the price of building WuR-enabled sensors would rise.

This work focuses on a multi-hop linear (i.e. chain-based) topology [13], which serves as the most fundamental building blocks of large-scale topologies. The sink is located at one end of the linear topology. The constituting sensors can be thought of as cluster-heads. They forward data packets towards the sink on behalf of their upstreaming sensors, as well as their own cluster-members. Each sensor comprises one main radio transceiver and another ultra-low power radio receiver (WuRx). The main radio transceiver remains asleep as long as possible. In contrast, the WuRx is always on. WuRx receives only wake-up beacon packets and no data packets. Moreover, WuRx cannot transmit packets. Wake-up beacon packets are generated and transmitted by the main radio transceiver with dynamic physical technologies. The wake-up beacon packets as well as data/acknowledgement packets share the same frequency band. Nevertheless, the wake-up beacon packets are transmitted at a lower bit rate yet with much stronger signal strength when compared to data/acknowledgement packets. This accounts for WuRx's lower signal sensitivity and slower signal processing capability.

Beacon packets are addressable. Specifically, each sensor gets its identification number at the initialization phase. When WuRx receives a wake-up beacon packet, it matches the identification number of its hosting sensor against that in the beacon packet. Whenever matched, a wake-up signal is generated and sent to the main controller of its hosting sensor. Subsequently, the main radio transceiver is switched on for exchanging wake-up acknowledgement, data, and data acknowledgement packets.

Wake-up radios have witnessed potential advantages in increasing energy efficiency and sustaining system performance simultaneously in wireless sensor networks. In general, related researching works can be categorized into either receiver-initiated (RI) or transmitter-initiated (TI) paradigms.

The receiver-initiated paradigm is suitable for data collecting applications, where wake-up beacon packets are to wake up neighboring senders who may possess interesting data packets. [7] presents and models a receiver-initiated consecutive packet transmission WuR MAC protocol, where multiple packet transmissions are packed into a single access winning competition. As such, losing senders avoid unnecessarily medium competitions in the multiple separate packet transmissions. Yet [7] only studies a single-hop network setting.

The transmitter-initiated paradigm is suitable for data reporting applications, where wake-up beacon packets are to wake up the relevant receivers (often sinks) that should get urgent data packets quickly. For instance, [5] introduces a backoff procedure before transmitting wake-up beacons in order to avoid potential collisions among wake-up beacons, and correspondingly removes backoff requirements from the main radios. It assumes the same contention window in every cycle with analysis based on discrete time Markov chain models. [6] presents and models performances of CCA WuR, CSMA WuR, and ADP WuR. It attempts to extend light traffic WuR-based MACs into varied heavy traffic scenarios. In contrast to previous always-on wake-up radios, [10] presents and optimizes a duty-cycled WuR-based MAC protocol, where the wake-up radio is duty-cycled in order to reach further higher energy efficiency. However, all these aforementioned TI works are still in single-hop networks.

As for formal analysis frameworks [3], [1] present absorbed Markov chain models to analyze TI works, where the number of transmission failures is assumed to follow a geometric distribution. However, both frameworks are based on the single sensor level, not on the MAC level competition in single-hop network, let alone on the multiple-hop network.

Mobile sinks [8] are used to collect data packets from adjacent sensors via transmitting on-demand wake-up beacons. In contrast, LoRa [4] combines long range wireless communication technologies to collect data packets directly (not by hops) from individual remote clusters that are equipped with wake-up radios.

Our work is obviously different from the previous ones, since we are focusing on designing MAC protocols in multiple-hop linear topology WuR-enabled wireless sensor networks, and may ignite potential researching interests shifting away from single-hop networks. The rest of the work is organized as follows: We introduce network background, design ideas, and present our MAC protocol in Sect. 2. Theoretical results on total en-route non-sleep time and simulation results on per-hop delay as well as energy consumption ratios are shown in Sects. 3 and 4, respectively. Finally, Sect. 5 concludes the work.

2 Proposed MAC Protocol in Linear Topologies

We first present background for multi-hop linear topology networks. Then we introduce design ideas of MAC protocol by exploiting characteristics of wake-up radios as well as the linear topology. Finally, we give the protocol details from an event-driven perspective in order to facilitate simulation implementation.

Background. Since the linear topology is considered in this work, all of the n sensors are positioned in a straight line with the distance between consecutive sensors being exactly the wake-up range. In particular, the sink is also on such a line, with the wake-up range being away from the nth sensor. In general, the main radio has better sensitivity and could detect much longer communication signals than WuRx. Yet in this work, these main radios are supposed to adjust their transmitting power to efficiently conserve limited onboard energy and also to match the wake-up range. In this way, the wake-up beacon packets and data packets can only reach the most adjacent sensors. The whole packet exchange between two consecutive sensors is as below: wake-up beacon, wake-up acknowledgement, data, and data acknowledgement. Note that all four kinds of packets are transmitted by the main radio. In contrast, except that wake-up beacon is received by WuRx, the remaining three are still received by the main radio. Recall that signals of the four kinds of packets occupy the same frequency band, it is possible to form collisions among them. Usually, the radio interference range of data and acknowledgement packets is longer than their communication range. As wake-up beacon has much stronger signal strength, its radio interference range is even longer than that of other three kinds of packets.

Each sensor on the linear topology, as well as its cluster-member sensors, continuously monitor their surroundings. Application-specific interesting data packets are obtained from time to time. Among them, delay-tolerant packets can be collected by duty-cycling methods or mobile sinks. Yet urgent packets, for instance, exceeding temperature threshold in fire detection and violating material concentration in environment monitoring, should be forwarded to the sink as quickly as possible. In realistic environments, the urgent data emerges randomly and significantly infrequently. It would be affordable by the wake-up based packet forwarding, because the wake-up beacons are much more energy consuming. Note that only the sensors constituting the linear topology are equipped with WuRx. Other cluster-member sensors in the sensor network are still traditional sensors. Once they have obtained the urgent data, they may wake up its belonging cluster-head sensor on the linear topology and then forwards its urgent data packets. Then the cluster-head sensor on the linear topology could wake-up its downstreaming sensor in turn. In this way, our linear topology network still has realistic application potentials, and also refrains from huge deployment cost.

As the same to the existing literature, urgent data packets in this paper on individual sensors are supposed to follow independent Poisson processes in temporal dimension. Each cluster-head sensor on the linear topology is supposed to have an aggregate urgent data rate of λ based on composition characteristics of Poisson data processes of individual cluster-member sensors. Hereafter, we investigate only cluster-head sensors, and postpone additional implications of wake-ups and data transmissions from cluster-member sensors in the future work.

Design Ideas. On the linear topology, it is intuitive for individual sensors to route the urgent data packets unidirectionally towards the sink, without any necessary complex routing decisions made on-the-fly. Thus, wake-up beacon packets are always towards the downstreaming sensors. The protocol stack would be mainly issues at the MAC layer. Besides the traditional CSMA rule, design ideas for our MAC protocol in the linear topology are as follows: (1) when the sensor receives a wake-up beacon packet while it is involved with forwarding packets

Algorithm 1. On generating a data packet

```
1: append the data packet into the transmitting queue
2: if timer_data, timer_ack, timer_channel, timer_forward are all off then
3:     switch on the main radio if it is asleep
4:     set timer_channel on with timeout being interval_cca
5: end if
```

Algorithm 2. On receiving a wake-up beacon packet

```
1: switch on the main radio if it is asleep
2: set timer_channel off if it is on
3: transmit a wake-ack packet
4: the main radio switches into receiving state
5: set timer_data on with timeout being interval_data
```

to its downstreaming sensor, it should prioritize upstreaming packet forwarding activities immediately; (2) the sensor wakes up its downstreaming sensor to forward packets only after the upstreaming sensor has finished transmissions of all available data packets; (3) the sensor should refrain from waking up its downstreaming sensor if its last data forwarding activity has not gone enough long time.

Due to inherent characteristics of linear topology, it should be better to gracefully give transmission privileges to other sensors when detecting channel busy. If downstreaming sensors get the channel, the urgent data packets will be forwarded to the sink quickly. If the upstreaming sensors get the channel, sooner or later the current sensor will get the channel via receiving wake-up beacon packets. As such, concurrent forwarding activities on the linear topology will be separated by at least some physical hops in spatial dimension. This contributes to both interference mitigation and energy conservation simultaneously.

Proposed MAC Protocol. Now, we give the detail of our MAC protocol. Specifically, from an event-driven perspective, the sensor state remains unchanged unless (1) the sensor generates an urgent data packet; (2) the wake-up radio receives a wake-up beacon packet; (3) the main radio receives a wake-up or data acknowledgement packet; (4) the main radio receives a data packet; (5) timeout on data packet arrivals, timeout on wake-up or data acknowledgement packet arrivals, channel being continuously idle sufficiently long, and last downstreaming packet forwarding gone long time. We present how the sensor copes with each aforementioned triggering event in the following.

Algorithm 1 copes with the case of the sensor that generates an urgent data packet. Specifically, the sensor first appends the data packet in the transmitting queue. Then it checks whether it is waiting for data packet, data or wake-up acknowledgement packet, channel being idle sufficiently long, or last downstreaming packet forwarding gone enough long time. If any of these four timers is on, then nothing needs to be done. Otherwise, the main radio is switched on if it is still asleep. Then the main radio conducts channel CCA via setting a timer

Algorithm 3. On receiving a wake-up or data acknowledgement packet

```
1: set timer_ack off
2: remove corresponding data packet from the transmitting queue if data-ack received
3: if the transmitting queue is empty then
4:     the main radio switches into sleep state
5:     set timer_forward on to suspend downstreaming packet forwarding temporally
6: else
7:     transmit the data packet at the head of the transmitting queue
8:     set timer_ack on, and the main radio switches into receiving state
9: end if
```

Algorithm 4. On receiving a data packet

```
1: append the data packet into the receiving queue if not duplicated
2: respond with a data acknowledgement packet
3: the main radio switches into receiving state
4: set timer_data on with timeout being interval_data, for waiting for next data packet
```

(i.e. timer_channel) with the channel continuously idle for at least interval_cca time. Timer_channel could simply become timeouts every interval_cca seconds.

Algorithm 2 copes with the case of the sensor that receives a wake-up beacon packet. Specifically, if the main radio is asleep, then the sensor switches on it. The sensor sets timer_channel off with the purpose that any pending downstreaming packet forwarding, if existing, should be suspended immediately. Subsequently, the sensor responds with a wake-up acknowledgement packet to the upstreaming sensor. As such, the upstreaming packet reception is prioritized immediately. Afterwards, the main radio listens for the channel and waits for receiving forthcoming data packets in interval_data seconds.

Algorithm 3 copes with the case of the sensor that receives a wake-up acknowledgement packet or data acknowledgement packet. Specifically, the sensor first sets off the corresponding timer (i.e. timer_ack). If a data acknowledgement packet is received, then the corresponding data packet is removed from the transmitting queue. Subsequently, if there is no data packet in the transmitting queue, the sensor switches the main radio off and also sets timer_forward on with timeout being interval_exchange. As such, consecutive downstreaming packet forwardings can be separated by sufficient long time. In case data packets exist in the transmitting queue, the sensor transmits the data packet at the queue head, sets on a timer (i.e. timer_ack) with timeout being interval_ack seconds that is to wait for data acknowledgement packet, and switches the main radio into receiving state.

Algorithm 4 copes with the case of the sensor that receives a data packet. Specifically, the sensor appends the received data packet into the receiving queue if not duplicated. Then the sensor responds with a data acknowledgement packet and the main radio switches into receiving state. Finally, the sensor sets on a timer (i.e. timer_data) with timeout being interval_data in order to determine whether the upstreaming sensor has finished all data packet forwardings.

Algorithm 5. On four kinds of timeouts

```
 1: if timeout is from timer_data or timer_forward then
 2:    set the corresponding timer_data or timer_forward off
 3:    if timer_forward is off then
 4:       if the receiving queue is nonempty then
 5:          invoke Algorithm 1 with each packet extracted from the receiving queue
 6:       else if the transmitting queue is nonempty then
 7:          set timer_channel on, with continuously idle being at least interval_cca
 8:       end if
 9:    end if
10: else if timeout is from timer_ack then
11:    set timer_ack off
12:    set timer_channel on with random-value timeouts, being idle at least interval_cca
13: else if timeout is from timer_channel then
14:    if channel continuously idle time is less than interval_cca then
15:       reset timer_channel on again with random-value timeouts
16:    else
17:       set timer_channel off
18:       if an expected data acknowledgement packet does not arrive then
19:          transmit a data packet at the head of the transmitting queue
20:          set timer_ack on, waiting for a data acknowledgement packet
21:       else
22:          transmit a wake-up beacon packet
23:          set timer_ack on, waiting for a wake-up acknowledgement packet
24:       end if
25:    end if
26: end if
```

Algorithm 5 copes with different cases of timeouts. Specifically, if the timeout is from timer_data or timer_forward, the sensor sets the corresponding timer off. Then it checks whether the timer_forward is off (i.e. last downstreaming packet forwarding has gone long time). If so, all of the data packets within the receiving queue is in turn extracted and feeded to Algorithm 1. Otherwise, i.e. the receiving queue is empty, then the sensor starts a clear channel assessment process whenever the transmitting queue has pending data packets. If the timeout is from timer_ack, the sensor sets it off, and sets timer_channel on with the timeout being some randomly chosen large value (i.e. implementing backoff effects). In general, the range of the random timeout is multiple times of a single data packet forwarding. If the timeout is from timer_channel, the sensor checks whether the channel has been continuously idle for at least interval_cca. If not, the sensor resets the timer_channel and checks at its next timeout again. If so, the sensor sets timer_channel off. Depending on the current context, the sensor transmits a data or wake-up beacon packet and sets timer_ack on, waiting for corresponding acknowledgement packet.

3 Theoretical Results

The objective of this section is to derive average radio-on time interval for intermediate sensor when being waken up on the linear topology. In the following theoretical analysis, we assume there is no signal interference among concurrent packet forwardings as well as channel errors for simplified derivations (The simulation results in the next section will account for signal interferences, while channel errors will be considered in the future). Recall that each sensor independently generates urgent data at λ packets/second, and these packets should be forwarded towards the sink without en-route data aggregation.

Let τ denote time interval of one data packet transmission as well as its acknowledgement packet. Suppose the intermediate sensor in question has received k $(k = 1, 2, \cdots)$ data packets from the upstreaming sensor, and it needs to forward these k packets towards the sink. We have the following major result.

Theorem 1. *Let $P_{j|k}$ denote the probability that given k packets in buffer, the sensor forwards these k packets and then consecutively forwards j locally generated packets within one continuous forwarding activity, then*

$$P_{j|k} = \frac{\lambda^j}{j!} k\tau(k\tau + j\tau)^{j-1} e^{-\lambda(k\tau + j\tau)},\ k = 1, 2, 3, \cdots,\ j = 0, 1, 2, \cdots. \tag{1}$$

Proof. Let X_i $(i = 1, 2, \cdots)$, denote the time interval between $(i-1)$th and ith local data packets, we can easily know for any k $(k = 1, 2, 3, \cdots)$ that $P_{0|k} = P(X_1 \geq k\tau)$, and for any j $(j = 1, 2, \cdots)$ that

$$P_{j|k} = P\left(X_1 < k\tau, \cdots, \sum_{i=1}^{j} X_i < (k + j - 1)\tau, \sum_{i=1}^{j+1} X_i \geq (k + j)\tau\right).$$

Denote by $P_0(x) = P(X_1 \geq x)$, and for any j $(j = 1, 2, \cdots)$ that

$$P_j(x) = P\left(X_1 < x, \cdots, \sum_{i=1}^{j} X_i < x + (j - 1)\tau, \sum_{i=1}^{j+1} X_i \geq x + j\tau\right), \tag{2}$$

we use mathematical induction method to verify that for any $j \geq 0$ and $x > 0$

$$P_j(x) = \frac{\lambda^j}{j!} x(x + j\tau)^{j-1} e^{-\lambda(x + j\tau)}. \tag{3}$$

In fact, it is easy to verify that the result is true for the base $j = 0$ or $j = 1$. Suppose the Eq. (3) is true for $j = n$, we need to verify that it is also true for $j = n + 1$. Now, for any $x > 0$, we have

$$\begin{aligned} P_{n+1}(x) &= \int_0^x f(t) P_n(x + \tau - t) dt = \int_\tau^{x+\tau} f(x + \tau - y) P_n(y) dy \\ &= \int_\tau^{x+\tau} \lambda e^{-\lambda(x+\tau-y)} \cdot \frac{\lambda^n}{n!} y(y + n\tau)^{n-1} e^{-\lambda(y+n\tau)} dy \\ &= \frac{\lambda^{n+1} e^{-\lambda[x+(n+1)\tau]}}{(n+1)!} \cdot (n+1) \cdot \int_\tau^{x+\tau} y(y + n\tau)^{n-1} dy. \end{aligned} \tag{4}$$

It is obvious by a straightforward mathematical manipulation that

$$(n+1)\int_{\tau}^{x+\tau} y(y+n\tau)^{n-1}dy = x[x+(n+1)\tau]^n. \tag{5}$$

Substituting Eq. (5) into Eq. (4), we will have

$$P_{n+1}(x) = \frac{\lambda^{n+1}}{(n+1)!}e^{-\lambda[x+(n+1)\tau]}\cdot x[x+(n+1)\tau]^n, \quad x>0. \tag{6}$$

This means that result (3) is also true for $j = n+1$, and therefore the theorem is verified by noting $P_{j|k} = P_j(k\tau)$ through mathematical induction method. $\square$

Remark 1. *When $j = 0, 1$, we have $P_{0|k} = e^{-\lambda k\tau}$ and $P_{1|k} = k\lambda\tau e^{-\lambda(k+1)\tau}$. This respectively represents the probability that after the sensor forwards all of k packets, no or only one local data packet is generated and forwarded.*

Based on the above main Theorem, we may have the follow direct corollary.

Corollary 1. *Denote by K the number of packets at the epoch of starting forwarding packets at any sensor and J_K is the number of total new packets forwarded from the starting epoch to the epoch when no packet is available in the given sensor. If the distribution of K is given by q_k ($k = 1, 2, ...$), then the average number, denote it by Φ, of packets forwarded in a cycle from starting epoch to the ending epoch for the specified sensor is given by*

$$\Phi = \sum_{k=1}^{\infty}\sum_{j=0}^{\infty}\frac{q_k\lambda^j}{j!}k(k\tau+j\tau)^j e^{-\lambda(k\tau+j\tau)}$$

Proof. By using the major result in the Theorem, we will have

$$\begin{aligned}\Phi =& E[K+J_K] = \sum_{k=1}^{\infty}E[k+J_k]q_k = \sum_{k=1}^{\infty}\sum_{j=0}^{\infty}(k+j)P_{j|k}q_k \\ =& \sum_{k=1}^{\infty}\sum_{j=0}^{\infty}\frac{q_k\lambda^j}{j!}k(k\tau+j\tau)^j e^{-\lambda(k\tau+j\tau)}\end{aligned}$$

Considering the first sensor on the linear topology, its q_k when $k = 1$ is equal to 1. Then its $P_{j|1}$ is calculated. Considering the second sensor, its q_k, when $k = 1 + j$, is equal to $P_{j|1}$ multiplied by $q_1 = 1$ of the first sensor. Considering each other downstreaming sensor on the linear topology, its q_k is equal to the sum of $P_{k-\theta|\theta}$ multiplied by q_θ of its upstreaming sensor where $\theta = 1, 2, \cdots, k$.

Remark 2. *When accounting for time intervals of wake-up and wake-ack packet, let $\xi = T_{wakeup} + T_{wake\_ack}$ that is fixed and independent of τ, then the modified version of $P_{j|k}$, denote it by $P'_{j|k}$, is given by*

$$P'_{j|k} = P\left(X_1 < k\tau+\xi, \cdots, \sum_{i=1}^{j}X_i < (k+j-1)\tau+\xi, \sum_{i=1}^{j+1}X_i \ge (k+j)\tau+\xi\right).$$

With similar derivations, the modified version of Φ, denoted by Φ', can also be derived similarly. For each intermediate sensor on the linear topology, its radio-on time when waken up can thus be approximated by the τ multiplied by the sum of Φ' of the upstreaming sensor and current sensor. This expression is helpful for deployment where corresponding sensor density should be achieved for getting a balanced lifetime across different linear topology positions.

4 Simulation Results

We present simulation results of performance of our MAC protocol where packet forwarding undergoes signal interference that is neglected in the previous analysis results. We made a custom-built discrete-event simulator based on R programming language, strictly according to the aforementioned five algorithms. The relevant time instants and energy consumption of main radios are recorded until all data packets arrive at the sink. The default values of relevant parameters are shown in Table 1, which is referred to [5]. In addition, the interference range of data signals is twice of inter-sensor distance, while the interference range of wake-up beacon signals is four times of that distance. Each sensor produces data at a rate of 0.1 packet/second according to independent Poisson processes.

Table 1. Parameter configuration

Parameters	Data packet	ack packet	Data rate	cca interval	cca current	rx current
Value	35	5	20	128	20.28	18.8
Unit	bytes	bytes	kbps	μs	mA	mA
Parameters	Beacon packet	Beacon rate	Data tx current		Beacon tx current	
Value	6	8192	17.4		152	
Unit	bytes	bps	mA		mA	

In order to achieve comparable results among different scales of linear topology networks, we normalize performance results as follows. As data packets emerge randomly among individual sensors, we get the per-hop delay being total end-to-end delay of all data packets divided by their total hops. Then normalized per-hop delay ratio is obtained with the per-hop delay divided by the idealized one, which is equal to sum of time intervals for CCA, wake-up, wake-up acknowledgement, data and data acknowledgement packets. Similarly, normalized per-hop energy ratio is obtained with the per-hop energy consumption divided by the idealized one, which is equal to sum of energy consumption for conducting CCA, transmitting wake-up beacon packets, transmitting and receiving wake-up acknowledgement/data/data acknowledgement packets. Note that receiving wake-up packets is not accounted for, because WuRx is always on. The following plots are an average of 50 independent runs of simulations.

Figure 1 shows the per-hop delay ratio when varying number of sensors in the linear topology network, with total number of packets injected being 100. We

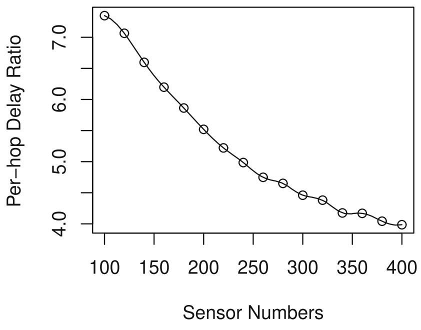

Fig. 1. Per-hop delay ratio vs sensors

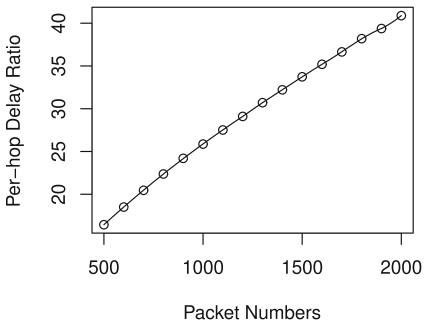

Fig. 2. Per-hop delay ratio vs packets

observe that as the number of sensors increases, the per-hop delay ratio decreases. This is because the average spatial distance between data packets grows, which mitigates signal interference among packets. Figure 2 shows the per-hop delay ratio when varying number of packets injected into the network, with the number of sensors being 200. We can see that more packets injected, much larger the ratio becomes. A deep investigation reveals that some spatial-and-temporal adjacent data packets would pack together into single packet propagation process. Multiple packets may be forwarded in a single wake-up activity. Thus, delay accumulates quickly because the previously received packets in the queue cannot be forwarded further by the downstreaming sensor until all packets have received from the adjacent upstreaming sensor.

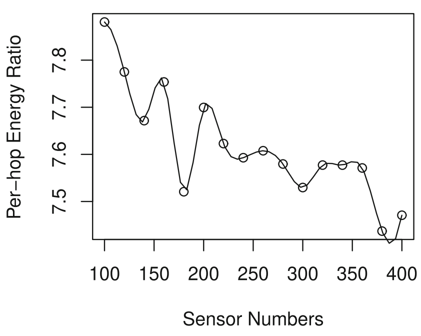

Fig. 3. Per-hop energy ratio vs sensors

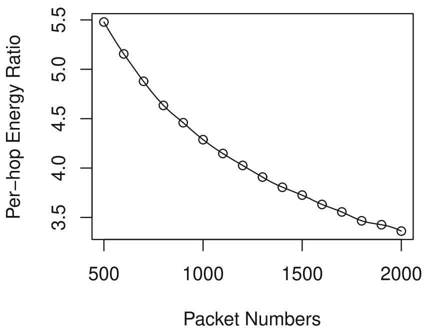

Fig. 4. Per-hop energy ratio vs packets

Figure 3 shows the per-hop energy ratio when varying number of sensors, with packets injected being 100. We observe that the energy ratio fluctuates within a small value range, indicating the average per-hop energy consumption remains stable. Figure 4 shows the per-hop energy ratio when varying number of packets injected, with sensors being 200. We find that the per-hop energy ratio obviously goes down when more packets become available. This is due to chances of multiple data packets packed into wake-up batch forwarding activities.

5 Conclusions

This paper investigates how urgent data packets propagate on the linear topology WuRx-enabled wireless sensor networks. It presents some interesting design ideas, proposes detailed MAC protocol in event-driven processes, presents theoretical results on expected radio-on time of intermediate sensor when waken up without accounting for signal interferences, and shows numerical results on normalized per-hop delay and energy ratios upon a custom-built simulator with interferences implemented. It opens a new direction on evaluating WuR applications towards large-scale, multi-hop instead of single-hop (i.e. star-shaped) wireless sensor networks, especially when taking channel noise and non-independent urgent data process into account.

References

1. Ait Aoudia, F., Gautier, M., Magno, M., Berder, O., Benini, L.: A generic framework for modeling MAC protocols in wireless sensor networks. IEEE/ACM Trans. Netw. **25**(3), 1489–1500 (2017)
2. Alfayez, F., Hammoudeh, M., Abuarqoub, A.: A survey on MAC protocols for duty-cycled wireless sensor networks. Procedia Comput. Sci. **73**, 482–489 (2015)
3. Aoudia, F.A., Magno, M., Gautier, M., Berder, O., Benini, L.: Analytical and experimental evaluation of wake-up receivers based protocols. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–7, December 2016
4. Aoudia, F.A., Gautier, M., Magno, M., Gentil, M.L., Berder, O., Benini, L.: Long-short range communication network leveraging LoRa and wake-up receiver. Microprocess. Microsyst. **56**, 184–192 (2018)
5. Ghose, D., Li, F.Y.: Enabling backoff for SCM wake-up radio: protocol and modeling. IEEE Commun. Lett. **21**(5), 1031–1034 (2017)
6. Ghose, D., Li, F.Y., Pla, V.: MAC protocols for wake-up radio: principles, modeling and performance analysis. IEEE Trans. Ind. Inform. **14**(5), 2294–2306 (2018)
7. Guntupalli, L., Ghose, D., Li, F.Y., Gidlund, M.: Energy efficient consecutive packet transmissions in receiver-initiated wake-up radio enabled wsns. IEEE Sens. J. **18**(11), 4733–4745 (2018)
8. Iwata, M., Tang, S., Obana, S.: Sink-based centralized transmission scheduling by using asymmetric communication and wake-up radio. In: 2017 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6, March 2017
9. Magno, M., Jelicic, V., Srbinovski, B., Bilas, V., Popovici, E., Benini, L.: Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-up radio receiver. IEEE Trans. Ind. Inform. **12**(2), 633–644 (2016)
10. Mazloum, N.S., Edfors, O.: Influence of duty-cycled wake-up receiver characteristics on energy consumption in single-hop networks. IEEE Trans. Wirel. Commun. **16**(6), 3870–3884 (2017)
11. Oller, J., Demirkol, I., Casademont, J., Paradells, J., Gamm, G.U., Reindl, L.: Has time come to switch from duty-cycled MAC protocols to wake-up radio for wireless sensor networks? IEEE/ACM Trans. Netw. **24**(2), 674–687 (2016)
12. Piyare, R., Murphy, A.L., Kiraly, C., Tosato, P., Brunelli, D.: Ultra low power wake-up radios: a hardware and networking survey. IEEE Commun. Surv. Tutor. **19**(4), 2117–2157 (2017). Fourthquarter
13. Tang, S., Li, W.: QoS supporting and optimal energy allocation for a cluster based wireless sensor network. Comput. Commun. **29**(13), 2569–2577 (2006)