Latency Aware IPv6 Packet Delivery Scheme over IEEE 802.15.4 based Battery-Free Wireless Sensor Networks

Yi-Hua Zhu¹, Senior Member, IEEE, Shu-Wei Qiu¹,², Kai-Kai Chi¹, Member, IEEE, and Yuguang “Michael” Fang³, Fellow, IEEE

¹School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, Zhejiang 310023, China
²Department of Computer Science, Shantou Polytechnic, Shantou, Guangdong 515071, China
³Department of Electrical and Computer Engineering, University of Florida, USA

Abstract—Battery-Free Wireless Sensor Networks (BF-WSNs) have become increasingly useful for many applications and how to ensure timely information exchange between nodes in IP networks and those in BF-WSNs is indispensable. The 6LoWPAN protocol is usually used to deliver IPv6 packets over IEEE 802.15.4 based WSNs, and has resolved the size mismatching problem between IPv6 packets and 802.15.4 Medium Access Control (MAC) frames by using packet fragmentation scheme to break an IPv6 packet into multiple small pieces with each fitted into a single 802.15.4 MAC frame. Unfortunately, IPv6 packets in BF-WSNs may suffer from intolerable delay for timely reassembling back to IPv6 packets. In this paper, we present a Latency Aware IPv6 Packet Delivery (LAID) scheme to reduce such IPv6 packet latency while maintaining high packet delivery ratio. Our LAID considers charging time, data rate, and the Maximum Number of Transmission Trials (MNTT) used in the IEEE 802.15.4 MAC layer so that the minimum latency can be achieved by optimizing the pairing of data rate and MNTT. In addition, we apply network coding to improve packet delivery reliability. Our analysis shows that the proposed LAID significantly outperforms existing schemes with fixed data rates in terms of IPv6 packet latency.

Index Terms—Latency, MAC, 6LoWPAN, battery-free wireless sensor networks, energy harvesting wireless sensor networks, Internet of Things (IoT).

I. INTRODUCTION

NODES in traditional Wireless Sensor Networks (WSNs) are usually powered by batteries. In many applications, it is not convenient and even very difficult to replace batteries once a WSN is deployed. Therefore, Battery-Free WSNs (BF-WSNs) emerge. This paper focuses on BF-WSNs, in which sensor nodes do not have to have batteries but use various capacitors to harvest and store energy from ambience.

Normal energy sources in ambience include sunlight [1] [2], radio transmissions [3] [4], tremors, winds, piezoelectricity, and others [5]. Unlike traditional WSNs, BF-WSNs or Energy Harvesting WSNs (EH-WSNs) exhibit new characteristics [5] [6]. One of them is the intermittent and changeable energy source, which makes energy harvesting rate hard to accurately predict and residual energies of nodes fluctuate with time. Thus, data delivery in BF-WSNs proceeds intermittently because nodes have to switch to sleeping mode in order to harvest sufficient energies when their energies diminish below a threshold. Therefore, BF-WSNs face new challenges in data delivery [7].

Being major components of the Internet of Things (IoT), BF-WSNs are required to exchange information between various smart objects (such as sensors and actuators) with nodes in the Internet [8] [9]. Fortunately, the IPv6 over Low-Rate Wireless Personal Area Network (6LoWPAN) protocol [10] standardized by the Internet Engineering Task Force (IETF) can be used to address this issue. This protocol supports the nodes running IEEE 802.15.4 in BF-WSNs to deliver IPv6 packets. One of the main tasks in the 6LoWPAN protocol is to resolve the packet size mismatching problem, i.e., an entire IPv6 packet cannot be carried within a single 802.15.4 Medium Access Control (MAC) frame, because the Maximum Transmission Unit (MTU) of IPv6 packets generated in the global Internet is at least 1280 bytes whereas the size of 802.15.4 MAC frame is up to 127 bytes. With the fragmentation scheme introduced in the 6LoWPAN protocol, a gateway node, which is located at the boundary of an IP network and a BF-WSN, breaks an IPv6 packet into multiple small pieces fitted within a single 802.15.4 MAC frame. The IPv6 packet is reassembled at the destination when all pieces are received.

In IoT, numerous smart objects are deployed within a BF-WSN or an EH-WSN. The Internet users control them via commands encapsulated in an IPv6 packet. Unfortunately, the fragmentation scheme in the 6LoWPAN protocol may bring in intolerable packet delay and result in the serious problem that the commands may not be timely delivered to the destination. That is, the IPv6 packet cannot be reassembled because one or more of its pieces do not reach the destination on time, causing the command invalid after the IPv6 packet is reassembled at the destination or making smart objects malfunctioned. The main reasons causing this intolerable delay in an IPv6 packet are as follows: 1) at least one piece of the IPv6 packet is delayed because some nodes on the path connecting a source and a destination do not have sufficient energy for transmission and 2) at least one piece suffers from frequent retransmissions. Even worse, retransmission encounters shortage of power, which brings in greater delay. Furthermore, nodes also face the problem of making decision on choosing a proper data rate for transmission. The reason is that a higher data rate shortens the packet transmission time, but it may bring with a higher Bit Error Rate (BER), followed by frame retransmissions; whereas...
a lower data rate may lead to longer transmission time, which contributes packet delay, but it reduces the probability of retransmissions. In a word, it is important to design a packet delivery scheme that takes node’s energy charging time, data rate, and retransmissions into account so that the end-to-end packet latency is minimized.

Although (re)transmission/acknowledgement (ACK) mechanism, i.e., ARQ mechanism, is applied in the IEEE 802.15.4 MAC layer, which uses the parameter called the maximum number of retransmissions in order to improve packet delivery reliability, it is still possible that one piece of the IPv6 packet fails to be delivered to the recipient over a lossy wireless link after the maximum number of retransmissions is performed. If this occurs, the entire IPv6 packet cannot be reassembled at the recipient, which will cause all the pieces, including those that have been correctly received, to be retransmitted. To overcome this problem, we apply network coding to encode all pieces of the IPv6 packet so that loss of some encoded packets does not prevent the recipient from reassembling the original IPv6 packet. Considering only around 81 bytes are left in the IEEE 802.15.4 MAC layer for carrying the upper layer data when the security header is present [11], we apply the network coding scheme as those in our previous study [12] [13], in which the coding vectors only consume a few bytes.

The main contributions of this paper can be summarized as follows:

1) We develop a Latency Aware IPv6-packet Delivery (LAID) Scheme for BF-WSNs to reliably deliver IPv6 packets with the minimum end-to-end latency. In addition to the retransmission/ACK scheme applied in the MAC layer, network coding is adopted in the LAID to further improve packet delivery reliability.

2) We derive the analytical results for the successful transmission probability of a MAC frame, the average transmission time of an IPv6 packet, the energy consumption of an IPv6 packet, and the IPv6 packet latency for the proposed LAID. Moreover, we also formulate the Optimization Problem (OP) that minimizes the IPv6 packet latency, which takes the energy charging time, the data rate and the Maximum Number of Transmission Trials (MNTT) of a MAC frame into account. The OP achieves the minimum latency (including capacitor charging time) while keeping a high Packet Delivery Ratio (PDR) by using proper pairing of data rate and MNTT. Extensive simulation results show that the proposed LAID significantly outperforms the existing schemes with fixed data rate and MNTT.

The remainder of this paper is organized as follows. Related works are surveyed in Section 2, and the LAID scheme is introduced in Section 3. The latency of an IPv6 packet in the LAID scheme and the OP minimizing the latency are presented in Section 4. The performance analyses are shown in Section 5. We conclude this paper in Section 6.

II. RELATED WORKS

Harvesting energy from ambience for BF-WSN or EH-WSN nodes has been intensively investigated. In [14] and [15], the energy harvesting process was assumed to be a Bernoulli process in which a device harvests energy with a fixed probability in each time slot. Ho et al. [16] considered the harvested energy as a stochastic process and proposed a generalized Markovian model in view of the random nature in solar and piezoelectric energy sources. Additionally, a Markov chain model was introduced in [17] to evaluate the proposed Robust Probabilistic Flooding (RPF). Ventura et al. [18] considered the case where the time interval of energy source occurrences is exponentially distributed. Khan et al. [19] proposed the Trinomial Random Walk (TRW) model for the storage capacity of energy harvesting enabled sensors, which was applied in a comprehensive solar radiation data set. Moreover, Kansal et al. [20] used a prediction model based on an Exponentially Weighted Moving-Average (EWMA) filter to exploit the diurnal cycle in solar energy. Piorno et al. [21] presented the solar prediction algorithm called Weather-Conditioned Moving Average (WCMA). Lin et al. [22] investigated the energy model that allows different energy sources in heterogeneous environments in which nodes can work with multiple energy sources. Seah et al. [7] derived the energy harvesting rate using difference technology on 10 cm² energy harvesting material. Tacca et al. [23] considered constant energy harvesting rates, i.e., a fixed amount of energy was harvested in each time slot. The results presented in the above surveyed works, however, may not be applicable to the contemporary WSNs that apply IEEE 802.15.4 standard in the MAC and physical layers because they do not take into account the data rates and the parameters used in the (re)transmission/ACK mechanism introduced in this standard.

To improve packet delivery in WSNs, various coding based schemes have been proposed. Mutschlechner et al. [24] used erasure codes to transmit information from mobile sensor nodes to stationary base station, and showed that the communication reliability could be considerably improved without impacting the resulting delay. Srouji et al. [25] proposed a reliable packet delivery scheme, called Reliable Data Transfer Scheme (RDTS), in which each intermediate node performs erasure coding and adaptively calculates the number of redundant packets for the next hop, and showed that RDTS does bring in longer network lifetime. Yang et al. [2] considered using solar energy surplus to adaptively adjust the redundancy level of erasure codes so that the packet delivery ratio can be improved without significantly impacting the network lifetime. Salhi et al. [26] developed the Reliable Coding for ZigBee (Re-CoZi) to enable robust XOR coding for WSNs to improve reliability using echo-feedback packet reception and decoding acknowledgement. In [27], a retransmission scheme based on an energy-efficient and network coding was presented, which enables the intermediate node to recover the lost packets such that the energy consumed for retransmissions is reduced. By considering the simultaneous use of gradient broadcast routing, fountain codes and infra-flow network coding, Apavatjrut et al. [28] proposed the XLT-GRAB strategy, which improves the reliability, the delay, and the network lifetime. Zhu et al. [29] used the enhanced Reed-Solomon (E-RS) code to achieve the minimum energy consumption while keeping data gathering ratio over a preset threshold.

To reduce latency, Luo and Sun [30] constructed the ar-
architecture of an end-to-end communication system based on 6LoWPAN gateway, which features encapsulating 6LoWPAN adaptation layer in a network adapter driver in a personal computer, and showed that IPv6 hosts could be interactive with IP-based sensor nodes through 6LoWPAN gateway with acceptable latency and packet loss. Ludovici et al. [31] analyzed different routing solutions for 6LoWPANs such as ROR, MUR and enhanced ROR in terms of latency and energy consumption when transmitting IP fragmented packets. With a real commercial deployment, Hui et al. [32] showed that it is possible to simultaneously achieve an average duty cycle less than 0.4%, an average message delivery rate larger than 99.9%, and an average per-hop latency less than 125 ms over 12 months in different environments. Sagar et al. [33] introduced a passive wake-up radio device called range enhancing energy harvester mote, which uses the energy harvester circuit combined with an ultra-low-power pulse generator to trigger the wake-up of the mote so as to reduce the latency without increasing energy consumption.

The main difference between the proposed LAID and the above surveyed schemes is that in our LAID, capacitor charging time, data rate, and the parameters used in the retransmission/ACK mechanism in the 802.15.4 MAC layer are jointly optimized to achieve the minimum latency. Moreover, our LAID maintains high PDR and low energy consumption.

III. LAID SCHEME

Before we describe the LAID in detail, we briefly describe the components of a BF-WSN node and the channel model first.

A. Components of a BF-WSN Node

Generally speaking, the node in our BF-WSN we consider here has an energy storage unit to store harvested energy, which enables the node to operate uninterruptedly in case when either the node fails to harvest energy from ambience or the amount of energy being harvested is not sufficient for its normal operation. The major components of a BF-WSN node are shown in Fig. 1 [34], where the components “Energy Harvesting Device” and “Energy Storage Device” are used to harvest and store energy, respectively. Usually, capacitor is used as the energy storage device. Hence, we use energy-harvesting time and capacitor charging time (or charging time for short) interchangeably.

B. Channel Model

We use \( (A_i, A_{i+1}) \) to represent the wireless link that connects nodes \( A_i \) and \( A_{i+1} \), and \( d_{i,i+1} \) for the distance between

\[
A_i \text{ and } A_{i+1}. \quad \text{In the IEEE 802.15.4 system using Direct Sequence Spread Spectrum (DSSS) and Offset Quadrature Phase Shift Keying (O-QPSK) modulation, the BER of link (} A_i, A_{i+1} \text{) can be calculated by} [35] \\
\]

\[
b_{i,i+1} = Q(\sqrt{2\psi(d_{i,i+1})B_N/R_i}), \\
\]

where \( B_N \) is the noise bandwidth, \( R_i \) is the data rate for \( A_i \), and the received Signal to Noise Ratio (SNR) with distance \( d_{i,i+1} \) is given by

\[
\psi(d_{i,i+1}) = P_t - P_{\text{loss}}(d_0) - 10\eta \log_{10}(d_{i,i+1}/d_0) - P_{th}, \quad \text{(3)}
\]

where \( P_t \) is the transmitted power, \( P_{\text{loss}}(d_0) \) is the path loss at \( d_0 \) (=1m), \( \eta \) is the path loss exponent ranging from 2 to 4, and \( P_{th} \) is the noise power (or noise floor) in dB. The noise floor depends on the environment as well as time. To make the transmitted packet received successfully, the received power is required to be greater than \( P_{th} \).

C. LAID Scheme

A gateway, which is located at the boundary of the IP network and our BF-WSN, is responsible for breaking IPv6 packet into small pieces with each suitable for a single MAC frame carried by BF-WSN nodes. In this paper, we classify the packets passing through the gateway into two categories: down-going packets, which traverse from the IP network to the BF-WSN, and up-going packets, which move in the opposite direction. Down-going packets usually contain commands to control BF-WSN nodes while up-going ones carry data sensed by BF-WSN nodes. Noticing that the up-going packets are small and can be carried within one IPv6 packet, i.e., it is not needed to break them up, we only consider delivering the down-going packets, which is illustrated in Fig. 2. In the figure, \( A_0 \) is the gateway and nodes \( A_1, A_2, \ldots, A_n \) stand for BF-WSN nodes. The route from the source \( A_0 \) to the destination \( A_n \) contains \( n \) hops through the intermediate nodes \( A_1, A_2, \ldots, A_{n-1} \).

In the 6LoWPAN protocol, Route-over Routing (ROR) and Mesh-under Routing (MUR) are introduced to deliver the fragments of an IPv6 packet. It has been shown in [12] that PDR, i.e., the probability of successfully delivering an IPv6 packet from source \( A_0 \) to destination \( A_n \), under ROR is the
same as that under MUR. Hence, without losing generality, we focus on ROR, in which the fragments are delivered in
a hop-by-hop manner. That is, an IPv6 packet is fragmented and
transmitted by A0, and is reassembled at A1 when all the
fragments are successfully received. Then, the IPv6 packet
is fragmented and transmitted again by A1, and is reassembled
at A2. This process repeats until the IPv6 packet reaches As.

We refer to an IEEE 802.15.4 MAC frame as a BF-WSN
frame below, which is no more than 127 bytes [36] whereas the
payload of an IPv6 packet is usually greater than 1000 bytes.
The LAID scheme, which is used to deliver the fragments of
an IPv6 packet from the gateway to BF-WSN nodes, consists of
two procedures: Proc_GATEWAY and Proc_MOTE. The former
is used at the gateway while the latter is applied at BF-WSN
nodes. In these procedures, node i (i = 0, 1, · · · , n − 1) sets
its MAC frame size (in bits) to

\[
l_i = \max_{x \in \{1, 2, \cdots, n\}} \{x | x < \frac{1}{b_{i, i+1}} H_p, H_m, 10 \times 8 \leq x \leq 127 \times 8\},
\]

where \(b_{i, i+1}\) is the BER of link (Ai, Ai+1) as given in (1),
\(H_m\) represents the header size of BF-WSN MAC frame (in
bits), and \(H_p\) is the PHY header size. From (4), we obtain
\((l_i + H_p)b_{i, i+1} < 1, i = 0, 1, \cdots, n − 1\). That is, the average
number of errored bits resulting from transmitting an \(l_i\)-bit frame
over each hop on the route from A0 to An is less than 1,
which aims to improve packet delivery reliability. In addition,
\(l_i \leq 127 \times 8\) guarantees that the size of a BF-WSN MAC frame
is not greater than 127 bytes, which is required in the
IEEE 802.15.4 standard; and \(H_m + 10 \times 8 \leq l_i\) guarantees that
the frame has at least 10 bytes of payload size. Then, node i
breaks an IPv6 packet up into

\[
s_i = \lfloor L/(l_i - H_m) \rfloor
\]
fragments, where \(\lfloor \cdot \rfloor\) is the ceiling function and \(L\) is the size
of the IPv6 packet.

Moreover, for a given MAC frame, we use \(K_i\) to represent
the MNTT applied at Ai to deliver the frame over link
(Ai, Ai+1), which is the maximum number of transmissions
permitted for the same frame, including the first transmission
and the subsequent retransmissions. Then, the probability of
successfully transmitting the \(l_i\)-bit MAC frame and the \(H_p\)-bit PHY
header over link (Ai, Ai+1) with MNTT \(K_i\) is

\[
p_{i,i+1} = 1 - \left(1 - b_{i, i+1}\right)^{H_p + l_i} K_i, i = 0, 1, \cdots, n - 1.
\]

Noticing that the gateway is located at the IP network
boundary, we assume the gateway does not apply energy
harvesting. The main steps of Proc_GATEWAY, which is triggered when
a down-going IPv6 packet reaches gateway A0, are as follows:

Step 1. A0 determines whether the IPv6 packet size \(L\) is
greater than \(l_0 - H_m\). If not, set \(s_0 = 1\) and go to Step 3.

Step 2. A0 breaks the IPv6 packet into \(s_0\) fragments, where
\(s_0\) is given in (5).

Step 3. A0 encodes the \(s_0\) fragments to generate \(M_0\) packets
using the network coding scheme with the property that any \(s_0\)
of the \(M_0\) encoded packets can recover the original fragments
of the IPv6 packet [12] [24] [25], where

\[
M_0 = \min\left\{\left\lfloor \frac{s_0}{\alpha \cdot p_{0,1}} \right\rfloor, 30s_0\right\}.
\]

Here, \(p_{0,1}\) is the success probability of transmitting a frame
with size of \(l_0\) over link (A0, A1) and \(\alpha \geq 1\) is a constant,
called redundant degree of network coding. The role of \(\alpha\) is
to increase the number of encoded packets, which increases
the probability that the recipient A1 successfully receives
\(s_0\) encoded packets so that A1 can retrieve the \(s_0\) original
fragments through decoding operation. We limit the number of
encoded packets to 30 times of \(s_0\) to avoid generating too
more packets.

Step 4. Node A0 determines its data rate \(R_0\) and MNTT
\(K_0\) using the OP in (43).

Step 5. A0 transmits the \(M_0\) encoded packets to A1 one
after another with data rate \(R_0\) and MNTT \(K_0\) while counting
the received ACK frames from A1. A0 stops transmitting
and removes the encoded packets from the buffer upon it receives
\(s_0\) ACK frames.


Step 1. Node Ai acknowledges with an ACK frame to Ai−1 as
soon as it successfully receives a fragment from Ai−1. If
the number of the received frames is equal to \(s_{i−1}\), Ai decodes
the original \(s_{i−1}\) fragments and reassembles the original IPv6
packet, from which the necessary information on routing the
IP packet is obtained.

Step 2. Node Ai breaks the IPv6 packet into \(s_i\) fragments
and encodes them into \(M_i\) packets using the network coding
scheme with the property that any \(s_i\) of the \(M_i\) encoded packets
can recover the original fragments of the IPv6 packet [12] [24]
[25]. Here, \(M_i = \min\{\lceil \alpha s_i/p_{i,i+1} \rceil, 30s_i\}\).

Step 3. Node Ai checks its energy storage device to see
if the residual energy suffices to transmit. If not, Ai defers
transmission until it harvests enough energy.

Step 4. Node Ai determines its data rate \(R_i\) and MNTT
\(K_i\) using the OP in (43).

Step 5. Node Ai transmits the \(M_i\) encoded packets to Ai+1 one
after another with data rate \(R_i\) and MNTT \(K_i\), counts the
number of the ACK frames from Ai+1, and stops transmitting
the remaining encoded packets when it has received \(s_i\) ACK
frames from Ai+1.

To save the energy consumed by the intermediate nodes in solving the OP in (43), Step 4 in Proc\_mote, which determines its data rate and MNTT, is conducted after fixed number of IPv6 packets are transmitted. That is, the pairing of data rate and MNTT from the OP’s solution has been applied until the new pairing is determined.

In fact, there are various types of BF-WSNs in which nodes can harvest energy from ambient light such as sunlight, radio transmissions, and other energy sources. Generally speaking, the amount of energy harvested from sunlight is much more than that from radio transmissions, e.g., Wireless Identification Sensing Platform (WISP) [37]. Hence, in the applications with more harvested energy, we can let each intermediate node solve the OP to obtain its pairing of data rate and MNTT; while in the applications with less harvested energy, the pairing can be determined by gateways, which have the sufficient energy supply, and then a gateway delivers the pairing to the participating nodes. In fact, the pairing can be piggybacked on data packets.

It should be pointed out that the MNTT $K_i$ and date rate $R_i$ are the critical parameters in the procedures of Proc\_Gateway and Proc\_mote. It can be seen from (1) that $R_i$ influences the BER $b_{i+1}$ over link $(A_i, A_{i+1})$ and further affects the success probability $p_{i,i+1}$. In general, as $R_i$ increases, the time consumed in transmitting one bit decreases, but $p_{i,i+1}$ decreases; and the growth in $K_i$ brings in high packet delivery ratio, but it may introduce longer delay. Therefore, how to set the optimal $K_i$ and $R_i$ to minimize packet latency is a critical problem in designing the LAID.

IV. IPv6 PACKET LATENCY UNDER THE LAID SCHEME

A. Timing in the LAID Scheme

In this section, we derive the latency for an IPv6 packet to be delivered from gateway $A_0$ to the destination $A_n$. As mentioned previously, an IPv6 packet is broken into $s_0$ fragments at $A_0$. We use $\xi_j$ to represent the preparation time of an IPv6 packet at node $A_i$ ($i = 1, 2, \ldots, n$), which is defined as the duration from the instant when $A_i$ receives the entire IPv6 packet from its preceding node $A_{i-1}$ to the instant when $A_i$ starts competing for the channel to transmit the first fragment of the IPv6 packet to its succeeding node $A_{i+1}$. In addition, we use $T_i$ to represent the transmission time of the IPv6 packet at $A_i$, which is defined as the duration from the instant when $A_i$ starts competing for the channel to transmit the first fragment to the instant when the IPv6 packet is transmitted to its succeeding node $A_{i+1}$ (i.e., all fragments of the IPv6 packet are transmitted). Moreover, we denote by $l_i(i = 1, 2, \ldots, n)$ the $i$-hop duration of the packet, which is the time period from the instant when the IPv6 packet reaches gateway $A_0$ to the instant when the entire IPv6 packet is received by $A_i$. That is, $l_i$ is the time for the packet to go over $i$ hops. Especially, $l_n$ is the latency of the IPv6 packet going from gateway $A_0$ to destination $A_n$. We refer to $l_n$ as the end-to-end latency. The relations among $t_i$, $\xi_i$, and $T_i$ are illustrated in Fig. 3.

From Fig. 3, we obtain

$$
T_i = \sum_{j=0}^{i-1} (\xi_j + T_j), i = 1, 2, \ldots, n. \quad (8)
$$

From (8), we obtain the expected end-to-end latency of an IPv6 packet as follows:

$$
E[l_n] = \sum_{j=0}^{n-1} (E[\xi_j] + E[T_j]) \quad (9)
$$

B. Transmission Time and Energy Consumption over One Hop

Next, we derive the transmission time $T_i$ and the corresponding energy consumption over one hop. From Step 3 of Proc\_Gateway and Step 5 of Proc\_mote, which are presented in Section III, we are aware that: 1) node $i + 1$ can recover the original IPv6 packet on the condition that the node receives $s_i$ encoded packets; 2) node $i$ stops transmitting the remaining encoded packets as soon as it receives $s_i$ ACK frames; and 3) $A_i$ generates totally $M_i$ encoded packets and transmits them over link $(A_i, A_{i+1})$ one after another until $A_{i+1}$ successfully receives $s_i$ encoded packets ($i = 0, 1, \ldots, n - 1$).

1) Time and Energy Consumption of Transmitting a Fragment over One Hop

For a given fragment of the IPv6 packet at $A_i$, at most $K_i$ transmission trials are performed. All possible transmission trials are summarized in Table I, where $\sigma_C$, $\sigma_A$, and $\sigma_T$ are the channel access time, the time consumed for the sender to

<table>
<thead>
<tr>
<th>TX trial</th>
<th>Status</th>
<th>Probability</th>
<th>Consumed Time</th>
<th>Transmitted bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>Success</td>
<td>$q_{i,i+1}$</td>
<td>$\sigma_C + \frac{H_p + l_i}{R_i} + \sigma_A$</td>
<td>$H_p + l_i$</td>
</tr>
<tr>
<td>2nd</td>
<td>Success</td>
<td>$(1 - q_{i,i+1})q_{i,i+1}$</td>
<td>$2(\sigma_C + \frac{H_p + l_i}{R_i}) + \sigma_T + \sigma_A$</td>
<td>$2(H_p + l_i)$</td>
</tr>
<tr>
<td>3rd</td>
<td>Success</td>
<td>$(1 - q_{i,i+1})^2q_{i,i+1}$</td>
<td>$3(\sigma_C + \frac{H_p + l_i}{R_i}) + 2\sigma_T + \sigma_A$</td>
<td>$3(H_p + l_i)$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$(K_i - 1)$th</td>
<td>Success</td>
<td>$(1 - q_{i,i+1})^{K_i-2}q_{i,i+1}$</td>
<td>$(K_i - 1)(\sigma_C + \frac{H_p + l_i}{R_i}) + (K_i - 2)\sigma_T + \sigma_A$</td>
<td>$(K_i - 1)(H_p + l_i)$</td>
</tr>
<tr>
<td>$K_i$th</td>
<td>Success</td>
<td>$(1 - q_{i,i+1})^{K_i-1}q_{i,i+1}$</td>
<td>$K_i(\sigma_C + \frac{H_p + l_i}{R_i}) + (K_i - 1)\sigma_T + \sigma_A$</td>
<td>$K_i(H_p + l_i)$</td>
</tr>
<tr>
<td>Failed</td>
<td>$(1 - q_{i,i+1})^{K_i}$</td>
<td>$K_i(\sigma_C + \frac{H_p + l_i}{R_i} + \sigma_T)$</td>
<td>$K_i(H_p + l_i)$</td>
<td></td>
</tr>
</tbody>
</table>
receive the ACK frame from the recipient, and the value of the timer set at the sender for a transmitted frame (i.e., the frame is retransmitted if the sender does not receive its ACK frame), respectively. In addition, $q_{i,i+1}$ is the success probability of transmitting the fragment to $A_{i+1}$ in one trial, i.e.,

$$q_{i,i+1} = (1 - b_{i,i+1})^{H_p + l_i}. \quad (10)$$

Now, we explain the main points in Table I. As one successful transmission trial suffices for the fragment, the $j$-th transmission trial takes place only when all the previous $j - 1$ trials failed, which has the probability of $(1 - q_{i,i+1})^{j - 1}$ and takes time $(j - 1)(\sigma_C + \frac{H_p + l_i}{R_i} + \sigma_T)$ since each trial takes time $\sigma_C$ in competing for the channel, time $(H_p + l_i)/R_i$ in transmitting the frame with $l_i$-bit PHY payload size plus $H_p$-bit PHY header size using data rate of $R_i$, and time $\sigma_T$ for the expiration of the retransmission timer. Additionally, a successful transmission at the $j$-th trial has the probability $(1 - q_{i,i+1})^{j-1}q_{i,i+1}$ and takes time $(j - 1)(\sigma_C + \frac{H_p + l_i}{R_i} + \sigma_T) + (\sigma_C + \frac{H_p + l_i}{R_i} + \sigma_A) = j(\sigma_C + \frac{H_p + l_i}{R_i}) + (j - 1)\sigma_T + \sigma_A(j = 1, 2, \cdots, K_i - 1)$, where $\sigma_A$ is the time taken in receiving an ACK frame. Similarly, for the $K_i$-th trial, we can derive the last two rows in Table I.

From Table I, we obtain the expected time consumed for successfully transmitting the fragment from $A_i$ to $A_{i+1}$ with MNTT $K_i$ as

$$\eta^{(S)}_{i,i+1} = \frac{1}{1 - (1 - q_{i,i+1})^{K_i}} \sum_{j=1}^{K_i} [(1 - q_{i,i+1})^{j-1}q_{i,i+1}] [j(\sigma_C + \frac{H_p + l_i}{R_i}) + (j - 1)\sigma_T + \sigma_A] = \frac{1}{1 - (1 - q_{i,i+1})^{K_i}} [(1 - q_{i,i+1})^{j-1}q_{i,i+1}] (\sigma_C + \frac{H_p + l_i}{R_i} + \sigma_T) + (\sigma_A - \sigma_T)$$

and the expected duration for the fragment failed over link of $(A_i, A_{i+1})$ with MNTT $K_i$ as

$$\eta^{(F)}_{i,i+1} = K_i(\sigma_C + \frac{H_p + l_i}{R_i} + \sigma_T). \quad (11)$$

In addition, from Table I, we obtain the expected energy consumption for $A_i$ to successfully transmit a fragment of the IPv6 packet to $A_{i+1}$ with MNTT $K_i$ as follows:

$$\xi^{TX DS}_{i} = \frac{1}{1 - (1 - q_{i,i+1})^{K_i}} \sum_{j=1}^{K_i} [(1 - q_{i,i+1})^{j-1}q_{i,i+1}] [j(H_p + l_i)\theta_1] = \frac{1}{1 - (1 - q_{i,i+1})^{K_i}} [(1 - q_{i,i+1})^{j-1}q_{i,i+1}] (H_p + l_i)\theta_1, \quad (12)$$

where $\theta_1$ is the energy consumption for transmitting one bit. In addition, the energy consumed by $A_i$ when it fails in transmitting the fragment over link $(A_i, A_{i+1})$ with MNTT $K_i$ is

$$\xi^{TX DF}_{i} = K_i(H_p + l_i)\theta_1. \quad (13)$$

Similarly, the expected energy consumption of $A_i$ in successfully receiving a fragment of the IPv6 packet from $A_{i-1}$ with MNTT $K_{i-1}$ is

$$\xi^{RX DS}_{i} = \frac{1}{1 - (1 - q_{i-1,i})^{K_{i-1}}} [(1 - q_{i-1,i})^{j-1}q_{i-1,i}] (H_p + l_{i-1})\theta_0 \quad (14)$$

and the expected energy consumption of $A_i$ failed in receiving a fragment of the IPv6 packet from $A_{i-1}$ over link of $(A_{i-1}, A_i)$ with MNTT $K_{i-1}$ is

$$\xi^{RX DF}_{i} = K_{i-1}(H_p + l_{i-1})\theta_0. \quad (15)$$
where $\theta_0$ is the energy consumption for receiving one bit. Here, $\theta_1$ and $\theta_2$ can be obtained using the energy consumption model presented in [38].

In addition, we obtain the expected energy consumption at $A_i$ in transmitting the ACK frame to $A_{i-1}$ to acknowledge the received fragment as

$$\bar{\varepsilon}_{i}^{RXA} = (H_p + L_A)\theta_1,$$

where $L_A$ is the size of ACK frame (in bits). Moreover, the expected energy consumption at $A_i$ in receiving the ACK frame from $A_{i+1}$ is

$$\bar{\varepsilon}_{i}^{RxA} = (H_p + L_A)\theta_0.$$  

2) Time and Energy Consumption for Transmitting IPv6 Packet over One Hop

The IP packet delivery has two outcomes: success and failure. The number of the encoded packets transmitted by $A_i$ ranges from $s_i$ to $M_i$. In the case when the outcome is success, exactly $s_i$ of the $M_i$ encoded packets are received since $A_i$ stops transmission as soon as $A_{i+1}$ receives $s_i$ packets, whereas when it fails, at most $s_i - 1$ of them are received. Generally, in order to let $A_{i+1}$ receive $s_i$ packets, $A_i$ may transmit $s_i + j$ encoded packets, where $j = 0, 1, \ldots, M_i - s_i$. The event that $A_{i+1}$ receives $s_i$ packets out of the $s_i + j$ encoded packets transmitted by $A_i$ is equivalent to the event that $A_{i+1}$ receives the last packet, i.e., the $(s_i + j)$-th packet, and exactly $s_i - 1$ out of the previous $s_i + j - 1$ packets. This event occurs with probability

$$\sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i} (1-p_{i+1})^j$$

and meanwhile $A_i$ consumes time of $s_i n_{i+1} + j n_{i+1}$ and energy of $s_i \varepsilon_i^{TXDS} + \varepsilon_i^{RxA}$ and energy of $s_i \varepsilon_i^{RXF} + \varepsilon_i^{RXA}$, where $j = 0, 1, \ldots, M_i - s_i$. This is because a frame goes from $A_i$ to $A_{i+1}$ with successful probability $p_{i+1}$; and a successful packet transmission consumes time of $n_{i+1}$ and energy of $\varepsilon_i^{TXDS} + \varepsilon_i^{RxA}$ whereas a failed packet transmission consumes time and energy as $n_{i+1}$ and $\varepsilon_i^{TXDF}$, respectively. In the case when the outcome is failure, the event that $s_i - 1$ out of $M_i$ packets are received occurs with probability $\sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i-1} (1-p_{i+1})^j$ and it consumes time of $j n_{i+1} + (M_i - j) n_{i+1}$ and energy of $j \varepsilon_i^{TXDS} + \varepsilon_i^{RxA}$ and $(M_i - j) \varepsilon_i^{TXDF}$.

As a result, under the condition that the IPv6 packet is successfully delivered to $A_{i+1}$, we have the expected transmission time $E[T_i]$ (see Fig. 3) as follows:

$$E[T_i] = \sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i} (1-p_{i+1})^j$$

$$\sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i-1} (1-p_{i+1})^j$$

$$[s_i n_{i+1} + j n_{i+1}] ; \ i = 0, 1, \ldots, n - 1.$$

Additionally, the expected time consumed by $A_i$ in transmitting the IPv6 packet to $A_{i+1}$, including both successful and failed transmissions, is as follows:

$$E[T_i] = \sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i} (1-p_{i+1})^j$$

$$\sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i-1} (1-p_{i+1})^j$$

$$[s_i n_{i+1} + j n_{i+1}] ; \ i = 0, 1, \ldots, n - 1.$$  

Similarly, under the condition that the IPv6 packet is successfully delivered to $A_{i+1}$, the expected energy consumption at $A_i$ in transmitting the IPv6 packet to $A_{i+1}$ and receiving the ACK from $A_{i+1}$ is

$$E[\bar{\varepsilon}_i^{TX}] = \sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i} (1-p_{i+1})^j$$

$$\sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i-1} (1-p_{i+1})^j$$

$$[s_i \varepsilon_i^{TXDS} + \varepsilon_i^{RxA}] + j \varepsilon_i^{TXDF}, \ i = 0, 1, \ldots, n - 1.$$  

The expected energy consumption at $A_i$ in transmitting the IPv6 packet to $A_{i+1}$ and receiving the ACK frames from $A_{i+1}$, including both successful and failed transmissions, is

$$E[\bar{\varepsilon}_i^{TX}] = \sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i} (1-p_{i+1})^j$$

$$\sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i+1})^{s_i-1} (1-p_{i+1})^j$$

$$[s_i \varepsilon_i^{TXDS} + \varepsilon_i^{RxA}] + j \varepsilon_i^{TXDF}, \ i = 0, 1, \ldots, n - 1.$$  

Likewise, under the condition that the IPv6 packet is successfully delivered from $A_{i-1}$ to $A_i$, the expected energy consumption at $A_i$ in receiving the IPv6 packet from $A_{i-1}$ and acknowledging the received encoded packets to $A_{i-1}$ is

$$E[\bar{\varepsilon}_i^{RX}] = \sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i-1})^{s_i-1} (1-p_{i-1})^j$$

$$\sum_{j=0}^{M_i-s_i} \binom{s_i+j-1}{s_i-1} (p_{i-1})^{s_i-1} (1-p_{i-1})^j$$

$$[s_i \varepsilon_i^{RXDS} + \varepsilon_i^{TXA}] + j \varepsilon_i^{RXDF}, \ i = 1, 2, \ldots, n.$$  

Additionally, the expected energy consumption at $A_i$ in receiving the IPv6 packet form $A_{i-1}$ and acknowledging the received encoded packets to $A_{i-1}$, including both successful and failed transmissions, is
\[
E[e_i^{Rx}] = \sum_{j=0}^{M_i - 1} \left( \frac{s_i - 1 + j - 1}{s_i - 1} \right) \left( p_{i-1,i} \right)^{s_i - 1 - j} \left( 1 - p_{i-1,i} \right)^j \left[ s_i - 1 \left( e_i^{Rx_{D_S}} + e_i^{Rx_{D_F}} \right) + j e_i^{Rx_D} \right] + \sum_{j=0}^{M_i - 1} \left( \frac{M_i - 1}{j} \right) \left( p_{i-1,i} \right)^j \left( 1 - p_{i-1,i} \right)^{M_i - 1 - j} \left[ j e_i^{Rx_{D_S}} + e_i^{Rx_{D_F}} \right],
\]
\[\text{for } i = 1, 2, \ldots, n. \tag{24}\]

In summary, the total energy expended by the source-to-destination route is
\[
E[e^{Total}] = E[e_0^{Tx}] + \sum_{i=1}^{n-1} \left( E[e_i^{Rx}] + E[e_i^{Tx}] \right) + E[e_n^{Rx}] = \sum_{i=1}^{n-1} E[e_i^{Tx}] + \sum_{i=1}^{n} E[e_i^{Rx}]. \tag{25}\]

Moreover, for \(i = 0, 1, \ldots, n - 1\), the probability that the IPv6 packet is successfully delivered over the \((i + 1)\)-hop, i.e., link \((A_i, A_{i+1})\), is
\[
\delta_{PDR}^{(n)} = \prod_{i=0}^{n-1} \sum_{j=0}^{M_i - s_i} \left( \frac{s_i + j - 1}{s_i - 1} \right) \left( p_{i,i+1} \right)^{s_i - 1} \left( 1 - p_{i,i+1} \right)^j.
\]

\[C. \text{ The End-to-end Latency}\]

Next, we begin to derive the expected end-to-end latency \(E[t_{\text{A}}]\). We assume any node on the source-to-destination route forwards the IPv6 packet immediately after some necessary procedures (such as reassembling the packet) are performed. Compared to the time expended in data communication, computing time in a node can be ignored. In addition, for tractability, we do not consider the delay resulting from abnormal conditions, such as congestion occurrence, hardware failure, etc. Besides, we assume gateway \(A_0\) does not harvest energy. Hence, \(\xi_0\) in Fig. 3 can be ignored, i.e., \(\xi_0 = 0\). Thus, \(t_1 = T_0\). In node \(A_i (i = 1, 2, \ldots, n - 1)\), packet preparation time \(\xi_i\) mainly includes the time consumed in recharging the energy storage device when the residual energy of the node is not sufficient for its transmissions.

For \(i \in \{1, 2, \ldots, n\}\), we denote the residual energy of \(A_i\) at time \(t\) by \(e_i(t)\); and the initial energy of \(A_i\) is set to \(e_0\), which is sufficient to receive an IPv6 packet. In addition, we use \(\xi\) to denote the energy threshold and \(\bar{\tau}\) the maximum energy the storage device holds. When the residual energy reaches \(\bar{\tau}\), no more energy can be saved in the device. A node transmits the packet immediately once its residual energy is more than \(\xi\), which is assumed large enough for transmitting all \(M_i\) encoded packets. Moreover, we use function \(g_i(t)\) to represent the instantaneous energy harvesting rate of \(A_i\) at time \(t(i = 1, 2, \ldots, n)\).

Now, we study the residual energy of the nodes at times of \(t_1, t_2, \ldots, t_n\) under the condition that the IPv6 packet is successfully delivered to the destination (i.e., all of its fragments are successfully delivered to the destination). From Fig. 3, we observe that at time \(t_1\) the residual energy of \(A_1\) is
\[
e_1(t_1) = \min \{\bar{\tau}, e_0 + \int_{0}^{t_1} g_1(x)dx - E[e_1^{Rx}]\}, \tag{27}\]

and the expected residual energy of the other nodes are
\[
e_j(t_1) = \min \{\bar{\tau}, e_0 + \int_{0}^{t_1} g_j(x)dx\}, j = 2, 3, \ldots, n. \tag{28}\]

where \(\int_{0}^{t_1} g_1(x)dx\) is the harvested energy during \(t_1\), and \(E[e_1^{Rx}]\), given in (24), is the expected energy consumption at node \(A_1\) in receiving the \(s_0\) fragments of the IPv6 packet and transmitting \(s_0\) ACK frames. As a result, the preparation time \(\xi_1\) is determined by
\[
\xi_1 = \begin{cases} 0, & \text{if } e_1(t_1) \geq \xi; \\ \min \{t | \int_{t_{i+1}}^{t_{i+1}+t_1} g_1(x)dx + e_1(t_1) > \xi\}, & \text{otherwise}. \end{cases} \tag{29}\]

In the above expression, the first line indicates \(A_1\) is able to transmit right away when its residual energy is over the threshold whereas the second line represents the charging time needed to harvest energy such that the residual energy is larger than the threshold \(\xi\).

At time \(t_2\) when the IPv6 packet reaches \(A_2\), the expected residual energies of \(A_1, A_2\), and \(A_j (j = 3, 4, \ldots, n)\) can be expressed by (30)-(32).
\[
e_1(t_2) = \min \{\bar{\tau}, e_0 + \int_{0}^{t_2} g_1(x)dx - E[e_1^{Rx}] - E[e_1^{Tx}]\}, \tag{30}\]
\[
e_2(t_2) = \min \{\bar{\tau}, \min \{\bar{\tau}, e_0 + \int_{0}^{t_1+\xi_1} g_2(x)dx\} + \int_{t_1+\xi_1}^{t_2} g_2(x)dx - E[e_2^{Rx}]\}, \tag{31}\]
\[
e_j(t_2) = \min \{\bar{\tau}, e_0 + \int_{0}^{t_2} g_j(x)dx\}, j = 3, 4, \ldots, n. \tag{32}\]

Additionally, the preparation time \(\xi_2\) is determined by
\[
\xi_2 = \begin{cases} 0, & \text{if } e_2(t_2) \geq \xi; \\ \min \{t | \int_{t_{i+2}}^{t_{i+2}+t_2} g_2(x)dx + e_2(t_2) > \xi\}, & \text{otherwise}. \end{cases} \tag{33}\]

At time \(t_3\) when the IPv6 packet reaches \(A_3\), node \(A_1\) does not involve data delivery but charges its energy storage device. Hence, the residual energy of \(A_1\) is
\[
e_1(t_3) = \min \{\bar{\tau}, e_0 + \int_{0}^{t_3} g_1(x)dx - E[e_1^{Rx}] - E[e_1^{Tx}]\}, \tag{34}\]

the residual energy of \(A_2\) changes to
\[
e_2(t_3) = \min \{\bar{\tau}, \min \{\bar{\tau}, e_0 + \int_{0}^{t_1+\xi_1} g_2(x)dx\} + \int_{t_1+\xi_1}^{t_3} g_2(x)dx - E[e_2^{Rx}] - E[e_2^{Tx}]\}; \tag{35}\]
Moreover, the residual energy of $A_3$ is
\[
e_{3}(t) = \min\{\tau, \min\{\tau, e_0 + \int_{0}^{t_3+\xi_2} g_3(x)dx\} + \int_{t_3+\xi_2}^{t_3} g_3(x)dx - E[e_{R3}], \}
\]  

(36)

Moreover, the residual energy of the other nodes are
\[
e_{j}(t_3) = \min\{\tau, e_0 + \int_{0}^{t_3} g_j(x)dx\}, j = 4, 5, \cdots, n. \quad (37)
\]

In addition, the preparation time $\xi_3$ is determined by
\[
\xi_3 = \begin{cases} 
0, & \text{if } e_{3}(t_3) \geq \xi; \\
\min\{t\} \int_{t_3+\xi_2} g_3(x)dx + e_{3}(t_3) > \xi, & \text{otherwise.}
\end{cases}
\]

(38)

Generally, we can derive the residual energy of the nodes at time $t_i$, when the IPv6 packet is delivered to $A_i$. That is, at time $t_i$, the residual energy of $A_j$ is
\[
e_{j}(t_i) = \begin{cases} 
\min\{\tau, e_0 + \int_{0}^{t_i} g_j(x)dx\}, & i \in [1, j-1]; \\
\min\{\tau, \min\{\tau, e_0 + \int_{0}^{t_i+\xi_1} g_j(x)dx\} + \int_{t_i+\xi_1}^{t_i} g_j(x)dx - E[e_{Rj}], \} & i = j; \\
\min\{\tau, \min\{\tau, e_0 + \int_{0}^{t_i+\xi_1} g_j(x)dx\} + \int_{t_i+\xi_1+\xi_j}^{t_i+\xi_1+\xi_j} g_j(x)dx - E[e_{Rj}], \} & i \in [j + 1, n].
\end{cases}
\]

(39)

Moreover, the residual energy of $A_n$ is
\[
é_{n}(t_i) = \begin{cases} 
\min\{\tau, e_0 + \int_{0}^{t_i} g_n(x)dx\}, & i \in [1, n-1]; \\
\min\{\tau, \min\{\tau, e_0 + \int_{0}^{t_i+\xi_1} g_n(x)dx\} + \int_{t_i+\xi_1+\xi_n}^{t_i+\xi_1+\xi_n} g_n(x)dx - E[e_{Rn}], \} & i = n.
\end{cases}
\]

(40)

Furthermore, summarizing (29), (33), and (38), we obtain
\[
\xi_i = \begin{cases} 
0, & \text{if } e_{i}(t_i) \geq \xi; \\
\min\{t\} \int_{t_i+\xi_1} g_i(x)dx + e_i(t_i) > \xi, & \text{otherwise.}
\end{cases}
\]

(42)

where $i = 1, 2, \cdots, n$.

As a result, the expected packet latency over $n$ hops, i.e., the end-to-end latency $E[t_n]$ shown in (9), can be found when (19) and (42) are applied.

D. The Optimization Problem (OP)

To minimize the end-to-end packet latency $E[t_n]$, including the charging time, for an IPv6 packet delivery, we only need to minimize packet delay over each hop on the source-to-destination route. That is, node $A_i(i = 0, 1, \cdots, n - 1)$ uses the following OP to minimize end-to-end packet delay:

\[
min E[T_i] + E[\xi_i]
\]

w.r.t. $K_i, R_i$

s.t. \[
\begin{cases} 
K_i \in \{1, 2, \cdots, 8\}; \\
R_i \in \Omega.
\end{cases}
\]

(43)

In the OP, the MNTT $K_i$ takes a value in the set of \{1, 2, \cdots, 8\} as the maximum number of retransmissions is set to 7 in the IEEE 802.15.4 standard; $\Omega$ is the set of available data rates supported by the IEEE 802.15.4g standard; and $E[T_i]$ and $E[\xi_i]$ are given in (19) and (42) respectively. The OP can be simply solved by enumeration because the number of feasible pairs of $K_i$ and $R_i$ are quite small.

V. PERFORMANCE EVALUATION

According to the IEEE 802.15.4 standard [36], a node contends for a channel before transmission, which consumes time of $\sigma_C$. In the standard, a backoff algorithm requires the node to randomly pick a Backoff Counter (BC) in the set $\{0, 1, \cdots, 2^{BE} - 1\}$, where $BE$ is the exponential backoff initialized to the value of the parameter $MacMinBE$ defined in the MAC layer [36], and then the node defers for BC slots each having a length of a $UnitBackoffPeriod$ symbol periods. Thus, when the channel is idle, the average backoff time of the node is $(2^{BE} - 1)/2$ slots. The experiments in [39] showed that in a small-scale network, a node needs 1 backoff on average; and even in a network with 3000 nodes, a node needs approximately 1.5 backoffs on average. Hence, we assume the average number of backoffs is 1 in our study, which yields

\[
\sigma_C = \frac{2^{BE} - 1}{2} UBP + CCA.
\]

(44)

Here, $UBP$ stands for $aUnitBackoffPeriod$; and $CCA$ means Clear Channel Assessment, which is the time required to assess an idle channel in the PHY layer.

Considering O-QPSK-A modulation introduced in the IEEE 802.15.4g standard [40] supports multi-rate transmissions, we adopt the PHY layer that applies O-QPSK-A modulation over the 868-870 MHz frequency band, which has 4 data rates of 6.25, 12.5, 25, 50 kbps so that $\Omega = \{6.25, 12.5, 25, 50\}$ in (43). With this modulation, the symbol period is 320 $\mu$s, and $CCA$ takes 4 symbol periods. Moreover, the default values of $MacMinBE$ and $aUnitBackoffPeriod$ are 3 and 20, respectively [40]. Thus, from (44), we have $\sigma_C = [((2^3 - 1)/2 \times 20 + 4) \times 320 = 23680 \mu$s.

With O-QPSK-A PHY, the value of the timer for a transmitted frame, i.e., $\tau_T$, is set to parameter $macAckWaitDuration$, which is equal to the sum of the parameters $aUnitBackoffPeriod$, $aTurnaroundTime$, $phySHRDuration$, $phyPHRDuration$, and $phyPSDUduration$. Here, $aTurnaroundTime$ is the RX-to-TX (receiver-to-transmitter) or TX-to-RX turnaround time, $phySHRDuration$ is the duration of the synchronization header (SHR) in the PHY, $phyPHRDuration$ is the duration of the PHY header (PHR), and $phyPSDUduration$ is the duration of an ACK frame [40]. Moreover, $phySHRDuration$ and $phyPHRDuration$ take 48 and 15 symbol periods, respectively; and $aTurnaroundTime = 1000 \mu$s. Furthermore, $phyPSDUduration$
IPv6 protocol, we choose IPv6 packet size as $L = 1300$ bytes in the MAC layer [36]. Considering the MTU is 1280 bytes in the link layer, we take the maximum size of PHY Service Data Unit (PSDU) in the Link Layer as $a_{\text{MaxPHYPacketSize}} = 127$, which is the size of ACK frame. As a result, $\sigma_T = 20 \times 320 + 15 \times 320 + 6400 = 33960 \mu s$. In addition, we adopt the following default values: $\text{macMaxFrameRetries} = 3$, which leads to $a_{\text{MaxPHYPacketSize}} = 127$, which maximizes the maximum size of PHY Service Data Unit (PSDU) in the PHY layer [36]. Considering the MTU is 1280 bytes in the IPv6 protocol, we choose IPv6 packet size as $L = 1300$ bytes [41]. The rest of the parameters are summarized in Table II.

As for energy harvesting rate, similar to [42], we use the solar irradiation data for Chicago O’Hare International Airport from the National Solar Radiation Data Base, which was published by The U.S. Department of Energy [43]. The average over data of direct and diffuse solar radiation received on a global horizontal surface during the 60-minute period in June, July and August of 2010 yield Fig. 4. From the figure, we observe that there is no solar radiation during periods of 0:00-4:00 and 20:00-24:00. It should be stressed that, the data shown in Fig. 4 are on 60-minute period basis, e.g., the peak of the figure indicates that the 60-minute period lasting from 11:00 to 12:00 can harvest energy of 623 Wh. Equivalently, 623 Ws (i.e., Joule) can be harvested for each second during period of 11:00 to 12:00. Considering that a sensor node has small size, we assume the surface of the device used to capture solar energy is a square with side of 2 cm, i.e., its area is 4 cm$^2$ or one 2500th of 1 m$^2$. Thus, in the simulation, we divide the data in Fig.4 by 2500. Especially, we study three cases with the lowest, medium, and the highest amount of harvested energy, which take the data during 19:00-20:00, 7:00-8:00, and 11:00-12:00, respectively. Thus, we have the energy harvesting rate function shown in (45), where the units of $x$ and $g(x)$ are second and Joule, respectively, and the notation of (7:00, 8:00] represents the period of 7:00 (exclusive) to 8:00 (inclusive). In (45), the function values are represented in fraction whose numerator is the same as the corresponding value in Fig. 4 and the 2500 in the denominator is for the square device surface with side of 2 cm.

$$g(x) = \begin{cases} 296.40/2500, & x \in (7:00, 8:00]; \\ 623.30/2500, & x \in (11:00, 12:00); \\ 1.93/2500, & x \in (19:00, 20:00). \end{cases}$$  

Moreover, we apply the energy consumption model presented in [38], from which $\theta_1 = E_{\text{elec}} + \epsilon_f d^2$ and $\theta_0 = E_{\text{elec}}$, where $d$ is the distance between the transmitter and the receiver, $E_{\text{elec}} = 50$ nJ/bit, and $\epsilon_f = 10$ pJ/bit/m$^2$.

Considering the expected transmission time $E[T_i]$ in (19), the total energy consumption in (25) and PDR in (26) play important roles in the LAID, we first verify them via simulation. In the simulation, we set radio range to 10, 20, 30, and 40 m, respectively. In addition, we randomly generate source-to-destination paths with 10, 20, ···, and 50 hops, respectively. The distance of the two neighboring nodes is randomly generated to meet the given radio range (the distance is regenerated if it is greater than the given radio range). In each path, we let the source node deliver 3000 IPv6 packets to

Fig. 4. The average solar radiation received on horizontal surface per hour.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_T$</td>
<td>1 dBm</td>
<td>$H_{\text{elec}}$</td>
<td>9 B</td>
</tr>
<tr>
<td>$H_N$</td>
<td>30 kHz</td>
<td>$H_{\text{loss}}$</td>
<td>39 B</td>
</tr>
<tr>
<td>$\eta$</td>
<td>2</td>
<td>$L_A$</td>
<td>5 B</td>
</tr>
<tr>
<td>$P_{\text{th}}$</td>
<td>-98 dBm</td>
<td>$\alpha$</td>
<td>0.5 J</td>
</tr>
<tr>
<td>$\eta$</td>
<td>2</td>
<td>$\epsilon_0$</td>
<td>0.3 J</td>
</tr>
</tbody>
</table>

TABLE II
PARAMETERS

$= (L_A/6250) \times 10^6 = 6400 \mu s$, where $L_A$ is the size of ACK frame in bits, i.e., $L_A = 40$ bits, and the lowest data rate, i.e., 6.25 kbps, is applied to reduce the loss probability of the ACK frame. As a result, $\sigma_T = 20 \times 320 + 1000 + 48 \times 320 + 15 \times 320 + 6400 = 33960 \mu s$. In addition, we adopt the following default values: $\text{macMaxFrameRetries} = 3$, which leads to $a_{\text{MaxPHYPacketSize}} = 127$, which is the maximum size of PHY Service Data Unit (PSDU) in the PHY layer [36]. Considering the MTU is 1280 bytes in the IPv6 protocol, we choose IPv6 packet size as $L = 1300$ bytes [41]. The rest of the parameters are summarized in Table II.

As for energy harvesting rate, similar to [42], we use the solar irradiation data for Chicago O’Hare International Airport from the National Solar Radiation Data Base, which was published by The U.S. Department of Energy [43]. The average over data of direct and diffuse solar radiation received on a global horizontal surface during the 60-minute period in June, July and August of 2010 yield Fig. 4. From the figure, we observe that there is no solar radiation during periods of 0:00-4:00 and 20:00-24:00. It should be stressed that, the data shown in Fig. 4 are on 60-minute period basis, e.g., the peak of the figure indicates that the 60-minute period lasting from 11:00 to 12:00 can harvest energy of 623 Wh. Equivalently, 623 Ws (i.e., Joule) can be harvested for each second during

period of 11:00 to 12:00. Considering that a sensor node has small size, we assume the surface of the device used to capture solar energy is a square with side of 2 cm, i.e., its area is 4 cm$^2$ or one 2500th of 1 m$^2$. Thus, in the simulation, we divide the data in Fig.4 by 2500. Especially, we study three cases with the lowest, medium, and the highest amount of harvested energy, which take the data during 19:00-20:00, 7:00-8:00, and 11:00-12:00, respectively. Thus, we have the energy harvesting rate function shown in (45), where the units of $x$ and $g(x)$ are second and Joule, respectively, and the notation of (7:00, 8:00] represents the period of 7:00 (exclusive) to 8:00 (inclusive). In (45), the function values are represented in fraction whose numerator is the same as the corresponding value in Fig. 4 and the 2500 in the denominator is for the square device surface with side of 2 cm.

$$g(x) = \begin{cases} 296.40/2500, & x \in (7:00, 8:00]; \\ 623.30/2500, & x \in (11:00, 12:00); \\ 1.93/2500, & x \in (19:00, 20:00). \end{cases}$$  

Moreover, we apply the energy consumption model presented in [38], from which $\theta_1 = E_{\text{elec}} + \epsilon_f d^2$ and $\theta_0 = E_{\text{elec}}$, where $d$ is the distance between the transmitter and the receiver, $E_{\text{elec}} = 50$ nJ/bit, and $\epsilon_f = 10$ pJ/bit/m$^2$.

Considering the expected transmission time $E[T_i]$ in (19), the total energy consumption in (25) and PDR in (26) play important roles in the LAID, we first verify them via simulation. In the simulation, we set radio range to 10, 20, 30, and 40 m, respectively. In addition, we randomly generate source-to-destination paths with 10, 20, ···, and 50 hops, respectively. The distance of the two neighboring nodes is randomly generated to meet the given radio range (the distance is regenerated if it is greater than the given radio range). In each path, we let the source node deliver 3000 IPv6 packets to
the destination. All the results shown in the subsequent figures are from the average values over the 3000 packet deliveries. Additionally, the simulation results indicate that, for the three energy harvesting rates in (45), the latency of the LAID shares similar tendency. So do the energy consumption and PDR of the LAID. Hence, we only present the results when the first line of (45) is applied, which is a moderate value. That is, the energy harvesting rate $g(x) = 296.40/2500 = 0.11856 \text{ J/s}$.

The simulation and analytical results of the end-to-end IPv6 packet latency, the energy consumption, and the PDR under the LAID are shown in Fig. 5(a), Fig. 5(b), and Fig. 5(c), respectively. The upper parts of the figures compare the simulation results with the analytical ones derived directly from (19), (25), and (26), while the lower parts show the errors of them. Here, we define the error as $|x - y|/x$, where $x$ and $y$ are the analytical and the simulation values, respectively. From Fig. 5, it can be clearly seen that the simulation and analytical results for latency, energy consumption, and PDR match very well because the errors are small. We are happy to see that PDR of the LAID is close to 1 (see Fig. 5(c)), which indicates the LAID is with very high reliability in packet delivery. We owe the high reliability to the network coding that encodes all the fragments of an IPv6 packet into multiple packets so that the receiving nodes are able to receive sufficient number of encoded packets to recover the ongoing IPv6 packet regardless of losing some encoded packets. It should be pointed out that, the packet latency shown in Fig. 5(a) does not include the charging time due to (19) excluding charging time.

Next, we compare the LAID with four schemes, each of which picks a Fixed Data Rate (FDR) in $\Omega = \{6.25, 12.5, 25, 50\}$ for transmissions. Accordingly, they are referred to as FDR6.25, FDR12.5, FDR25, and FDR50, respectively. In the FDR schemes, we set $\text{MNTT} = 4$, i.e., the maximum number of retrials in the MAC layer defaults to 3.

Setting the radio range to 10, 20, 30, 40, 50, respectively, we obtain Fig. 6, in which EHR stands for energy harvesting rate. This figure compares the LAID with FDR6.25, FDR12.5, FDR25, and FDR50 in terms of packet latency (including capacitor charging time) and the energy consumption. From Fig. 6(a), we observe that: 1) the LAID achieves the best latency, which fulfills the aim of this paper; and 2) increase in radio range almost does not affect latency (it only brings with slight increase in latency), which agrees with our intuition because radio propagates in ray velocity so that time expended in traversing tens of meters can be ignored. Fig. 6(b) illustrates that: 1) the energy consumptions of all the schemes share the same increasing trend, i.e., the greater radio range the more energy consumption; and 2) their energy consumptions are very close (see Fig. 6(c)). In a word, the LAID is able to achieve much smaller packet delay than the FDR schemes by using nearly the same energy as them.

Setting the number of hops, i.e., $n$, to 10, 20, 30, 40, 50, respectively, we obtain Fig. 7, where $R$ stands for the radio range. From Fig. 7 we observe that, with variation of $n$, the LAID remains the best in latency (see Fig. 7(a)) while its energy consumption is close to the other four schemes (see Fig. 7(b)). Moreover, increase in $n$ causes the latency and the energy consumption of all the schemes to increase. This is
because the IPv6 packet has to traverse more hops before it reaches the destination, resulting in higher packet latency and expending more energy.

Finally, we want to stress that, as mentioned previously, the PDRs of all the schemes are close to 1 (see Fig. 5(c)). So, we omit the figures for PDR.

VI. CONCLUSION

With the advances in IoT, Battery-Free Wireless Sensor Networks (BF-WSNs) have been widely deployed for many applications. One of the important communication tasks is to exchange information between nodes in the Internet and the BF-WSN nodes in order to integrate BF-WSNs into the Internet, one of the visions for IoT. Due to the huge number of envisioned communications devices connected to future Internet, IPv6 have to be used in the future. Unfortunately, IPv6 packets tend to have large size, and an IPv6 packet may not be fitted in an MAC frame for most wireless sensor networks, including BF-WSNs, without fragmentation. In fact, the 6LoWPAN protocol, which is standardized by the Internet Engineering Task Force (IETF) to deliver IPv6 packets over IEEE 802.15.4 standard based WSNs, adopts fragmentation technique. Unfortunately, with the 6LoWPAN protocol, end-to-end packet latency may be intolerable, which may cause IPv6 packets unable to reach the destination on time.

The possible factors affecting end-to-end packet latency in BF-WSNs are as follows. Firstly, packet delivery has to proceed in intermittent way. This is because the nodes have to enter low power mode to save and harvest energy when their residual energy are below threshold, which temporarily terminates the packet delivery, and wake up when they have sufficient energy for transmitting, which resumes the packet delivery. Secondly, packet delivery suffers from loss due to unreliable wireless links because the nodes are usually not able to transmit with high power level so that wireless links between the nodes prone to being broken, which may cause a fragment of the ongoing IPv6 packet unable to be delivered to a neighboring node even when the fragment is retransmitted with the greatest allowed parameter of maximum number of transmission retrials set in the MAC layer, thus preventing the destination from reassembling the original IPv6 packet. Therefore, how to efficiently deliver large-sized IPv6 packets over BF-WSNs suitable for small-sized packets is important and challenging, and should be carefully investigated. In this paper, we have addressed this important problem and have designed a Latency Aware IPv6 Packet Delivery (LAID) scheme to be implemented at the gateway nodes, which are located at the boundaries of the Internet and the BF-WSNs, and the nodes in the BF-WSNs in order to deliver IPv6 packets over the BF-WSNs in lowest latency. Through extensive evaluation, we have demonstrated that our LAID can considerably reduce the end-to-end packet latency over BF-WSNs by tuning the data rate and the MNTT in the MAC layer in BF-WSNs while maintaining high packet delivery ratio and consuming low harvested energy. We owe the high reliability, measured by PDR, to the network coding applied in the LAID, which encodes all fragments of an IPv6 packet into multiple packets.
so that loss of some encoded packets does not affect the recipient to recover the original IPv6 packet.

In addition to IEEE 802.15.4 standard based BF-WSNs, the proposed LAID may be applicable in other kinds of battery-free wireless networks. The key to the LAID is in solving the OP in (43) so as to find the optimal pairing of the data rate and the MNTT. Surely, it consumes energy to solve the OP. Hence, in practice, we should carefully consider when and where to solve the OP. The hint is to let the nodes solve the OP at a fixed interval, and the duration of the fixed interval depends on how much energy the nodes can harvest. In the extreme case when the energy consumed in solving the OP is considerably greater, compared to the harvested energy, we can let the gateway solve the OP and then piggyback the pairing of the optimal data rate and MNTT to the respective nodes.

REFERENCES


Yiu-hua Zhu (MO1-SM07) received his B.S. degree in mathematics from Zhejiang Normal University, Zhejiang, China, in July 1982; his M.S. degree in operation research and cybernetics from Shanghai University, Shanghai, China in April 1993; and his Ph.D. degree in computer science and technology from Zhejiang University, Zhejiang, China, in March 2003. Dr. Zhu is a professor at Zhejiang University of Technology, Hangzhou, Zhejiang, China. He is a member of China Computer Federation Technical Committee on Sensor Network. His current research interests include information dissemination, stochastic modeling and analysis, power management, mobility management for wireless networks, and network coding. He has served as technical program committee members or co-chairs in the international conferences IEEE ICC, WCNC, GlobeCom, DCOSS, etc. He is the recipient of the Best Paper Award of Chinacom 2008. He has published more than 170 research papers in proceedings and journals including IEEE Transactions on Wireless Communications, IEEE Transactions on Vehicular Technology, and more.

Shu-Wei Qiu received his B.S. degree from Hanshan Normal University, Chaozhou, China, in 2003; his M.S. degree from Guangdong University of Technology, Guangzhou, China, in 2009. He is currently a candidate student for his Ph.D degree in Zhejiang University of Technology. His current research interests include information dissemination for wireless networks, especially for battery-free wireless sensor networks and wireless local area networks.

Kaikai Chi received the B.S. and M.S. degrees from Xidian University, Xi’an, China, in 2002 and 2005, respectively, and the Ph.D. degree from Tohoku University, Sendai, Japan, in 2009. He is currently an associate professor in the School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His current research focuses on wireless ad hoc network and wireless sensor network. He was the recipient of the Best Paper Award at the IEEE Wireless Communications and Networking Conference in 2008. He has published more than 30 referred technical papers in proceedings and journals like IEEE Transactions on Wireless Communications and IEEE Transactions on Parallel and Distributed Systems.

Yuguang “Michael” Fang (F’08) received the M.S. degree from Qufu Normal University, China, in 1987, and Ph.D. degrees from both Case Western Reserve University, Cleveland, OH, USA and Boston University, Boston, MA, USA, in 1994 and 1997, respectively. He joined the Department of Electrical and Computer Engineering, University of Florida, FL, USA, in 2000 and has been a Full Professor since 2005. He received the US NSF CAREER Award in 2001, the US ONR Young Investigator Award in 2002, 2015 IEEE Communications Society CISTC Technical Recognition Award, 2014 IEEE Communications Society WTC Recognition Award, and is a recipient of the Best Paper Award in IEEE International Conference on Network Protocols in 2006. He also received a 2010C2011 UF Doctoral Dissertation Advisor/Mentoring Award. He served as the Editor-in-Chief of IEEE Wireless Communications and is currently serving as the Editor-in-Chief of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He is a Fellow of IEEE.