Multi-Channel Delay Sensitive Scheduling for Convergecast Network

Burghal, Daoud; Kim, Kyeong Jin; Guo, Jianlin; Orlik, Philip V.; Hori, Toshinori; Sumi, Takenori; Nagai, Yukimasa

TR2020-036 April 08, 2020

Abstract

Motivated by an increasing interest in wireless networking in mission-critical applications, and a recent amendment of the time slotted channel hopping to IEEE 802.15.4, the multichannel delay sensitive scheduling is investigated in the many toone network, which is also known as the convergecast network. In such a network, each node has data to be transmitted to a gateway through multi-hop communications. As a realistic setting, packet release time at each node is not assumed to be uniform. Under this assumption, the goal of this work is to design a scheduling scheme that minimizes the schedule length and maximum endto-end delay, in which the former is essential for repetitive data acquisition, whereas the later improves the freshness of the acquired data. To achieve the scheduling goal, the problem is formulated as a multi-objective integer programming. To obtain a feasible solution and gain an insight into the problem, a lower bound on the schedule length is derived. Based on that, a new scheduling scheme is designed to minimize the two objectives simultaneously. Link level simulations verify the performance improvement of the proposed scheme over the existing schemes.

IEEE Wireless Communications and Networking Conference (WCNC)

Multi-Channel Delay Sensitive Scheduling for Convergecast Network

Daoud Burghal, Kyeong Jin Kim, Jianlin Guo, Philip V. Orlik, Toshinori Hori, Takenori Sumi, and Yukimasa Nagai

***Abstract*—Motivated by an increasing interest in wireless networking in mission-critical applications, and a recent amendment of the time slotted channel hopping to IEEE 802.15.4, the multi-channel delay sensitive scheduling is investigated in the many-to-one network, which is also known as the convergecast network. In such a network, each node has data to be transmitted to a gateway through multi-hop communications. As a realistic setting, packet release time at each node is not assumed to be uniform. Under this assumption, the goal of this work is to design a scheduling scheme that minimizes the schedule length *and* maximum end-to-end delay, in which the former is essential for repetitive data acquisition, whereas the later improves the freshness of the acquired data. To achieve the scheduling goal, the problem is formulated as a multi-objective integer programming. To obtain a feasible solution and gain an insight into the problem, a lower bound on the schedule length is derived. Based on that, a new scheduling scheme is designed to minimize the two objectives simultaneously. Link level simulations verify the performance improvement of the proposed scheme over the existing schemes.**

***Index Terms*—Convergecast network, delay sensitive scheduler, schedule length, end-to-end delay.**

I. Introduction

Data collection convergecast wireless networks have typically a many-to-one network structure, in which wireless nodes transmit or relay data packets to a central node, also known as a gateway. When nodes are equipped with sensors, as in wireless sensor networks (WSN), the network can be used to monitor the environment or provide data on-demand at the central node. Such a network has been used in many applications, e.g., environment monitoring and basic industrial autonomous control. In fact, the recent interest in the Internet of Things (IoT), and the possible service integration in the next-generation wireless networks, are predicted to stimulate an unprecedented demand on such networks.

Despite the extensive research and the commercial interest in WSN, industrial and mission-critical applications have been relatively limited due to stringent reliability constraints and harsh environmental conditions. Different from many other applications, intermittent connectivity and delays could cause *catastrophic* consequences in these applications.

To provide a unified platform that addresses some of these issues, the standardization communities have made efforts to improve aspects in the related standards. For instance, the IEEE 802.15.4e was proposed as an amendment to the IEEE 802.15.4, to provide a reliable wireless connectivity in constraint environments. Prior to IEEE 802.15.4e, ISA100.11a and wirelessHART technologies were developed to satisfy some wireless communication requirements in the industrial market that were not met with IEEE 802.15.4 [1]. In fact, the medium access control (MAC) layer of IEEE 802.15.4e is based on wirelessHART, in which a time slotted channel hopping (TSCH) was introduced to utilize the advantages of multi-channel *synchronized* time division multiple access (TDMA) [2]. Using dedicated time slots, in TDMA systems, packets can be transmitted with low interference. In addition, the channel hopping provides a frequency diversity and improves the interference mitigation capabilities. Thus, a less number of re-transmissions and link failures is expected, which reduces the delay and improves the reliability.

In this paper, we revisit the latency issue in multi-channel convergecast networks. Different from previous works, we consider a non-uniform data generation time (release time), which is encountered in industrial settings, where data may not be available to the "source" wireless nodes at the beginning of the time-frames. This is due to the fact that the source nodes sense/collect the packets from possibly heterogeneous and/or interdependent processes. Thus, the availability times of the packets are de facto out of the control of the wireless nodes as well as the network controller. Fig. 1 provides an illustration of this example. Previous works have focused on the case in which the packets are available at the beginning of each slot-frame, e.g., [3]–[6], which is not applicable in many industrial settings.

The definition of delay depends on the application. In some cases, the goal is to collect the data from the sensor nodes as soon as they become available, to guarantee the "freshness" of data at the gateway. This is referred to as minimizing the end-to-end (E2E) delay. Alternatively, the goal could be to minimize the time needed until the last packet is received by the gateway, i.e., to reduce the schedule length. This can be viewed as a one-shot scheduling problem [3]. For the delay sensitive industrial applications, *both* quantities are important. For instance, in a control center, a small schedule length allows more frequent command/response cycles, whereas the second goal gives equal importance to the delay encountered by all packets. We emphasize that our focus is on the convergecast network, which is different from that of the Wireless Sensor Actor Network (WSAN) [7]. Thus, unicast for message transmission is not considered in this paper.

D. Burghal is with Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA

K. J. Kim, J. Guo, P. V. Orlik, and Y. Nagai are with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139 USA

T. Hori, and T. Sumi are with Mitsubishi Electric Corporation IT R&D Center, Ofuna, Kamakura, Kanagawa, 247-8501, Japan

*This work was done while D. Burghal was working with MERL.

A. Previous Work

The minimum latency scheduling for the converegcast network has been considered in several works. For instance, the authors in [6], proposed a distributed scheme that minimizes the schedule length in a single channel network. The authors in [3], derived a lower bound on the schedule length and proposed LOCAL-TIMESLOTASSIGNMENT as a distributed scheme that achieves the bound when interference is eliminated. Other schemes, such as WAVE [4], TASA [5], and MODESA [8], addressed the problem of joint scheduling and channel allocation with a minimum delay requirement. Several other schemes investigated the latency for the convergecast network [9]. However, all the aforementioned schemes considered that the packets are available at the beginning of the scheduling, and their goals were to minimize only the schedule length, so that their proposed schedulers do not guarantee a minimum E2E delay. The authors in [10] considered the E2E delay in real time scheduling. However, as we discuss below, minimizing the E2E delay does not necessarily minimize the schedule length. In addition, the problem considered in [10] is different from ours due to the presence of the deadline for the data flows.

B. Contribution

In contrast to the existing work, this paper makes the following contributions.

- For a *given* routing tree, we define the delay sensitive data collection problem, taking into consideration the TSCH requirements.
- We formulate the joint minimization (of schedule length and E2E delay) as an iterative integer linear program (ILP).
- We conduct delay analysis under the assumption that interference is mitigated. This helps in reducing the search space of the ILP, and provides valuable insights to the problem.
- Due to the complexity of the ILP, we propose a new threshold-based multi-channel multi-hop scheduling scheme, which priorities packet transmissions based on local conditions, such as delay of packets in the local buffers and the size of nodes' buffers. It can be seen that the time complexity of the proposed scheme is polynomial in the number of nodes in the network.

II. Network Model and Assumptions

We consider a synchronized many-to-one wireless network, which consists of N wireless nodes and a gateway denoted by g, we use $\mathcal{N}$ to denote the set of the nodes, and define $\widetilde{\mathcal{N}} = \mathcal{N} \cup \{g\}$. Each node $c \in \mathcal{N}$ needs to transmit a data packet, when it is available, to the gateway. We also use c to represent the packet generated at node c. Note that we consider only a single packet for simplicity. The derivation and the scheme are valid for a larger number of packets with minor modifications. The data packet at c, the endogenous packet, is assumed to be available at time $T_0^c \geq 1$, in the unit of time slots. We refer to T_0^c as the release time of c's packet. In this work, $T_0^c = t$ means that data to be transmitted is available at the beginning of time slot t. We also use $T^c_{(i\to j)}$ to denote the time slot used by i in transmitting packet generated by c to j.

The network is assumed to be a multi-hop network, i.e., data can be transmitted over several hops before it reaches g. We assume that the system performs raw data collection, i.e., no data aggregation. We also assume that there are L frequency channels. In our simulations, similar to IEEE 802.15.4e, we use $L = 16$ unless other values are needed for the discussion. Furthermore, the nodes are assumed to use omni-directional antenna and half-duplex transceivers, so that a node can be scheduled to receive or transmit on one frequency channel at any given time slot. We adopt the collision communication model, in which transmission at link $i \to j$ fails, $j \in \widetilde{\mathcal{N}}$, if k, a neighbor of j and $k \neq i$, transmits on the same frequency channel, and same time instance, that i uses to communicate to j. We use $\mathcal{I}_c$ to denote the set of links $i \to j$ that interfere with node c's transmissions. The set $\mathcal{I}_c$ depends on the network topology.

In many-to-one scheduling problems, it is usually assumed that a route from the sensor nodes to the gateway is known *a priori*. For simplicity, we assume that the route from all nodes to g constitutes a routing tree, e.g., using the default RPL routing protocol [11].

The goal of this work is to identify a valid time-frequency allocation, denoted by $\mathcal{S}$, to nodes such that the schedule length as well as the maximum E2E delay are minimized simultaneously. Both quantities are vital in delay sensitive applications. Given this goal, it is reasonable to consider a centralized scheme with a static scheduling method. Note that other problems such as routing, synchronization, and decentralized schemes, are out of the scope of this manuscript and left for future work.

A. Network and Schedule Notations

In general, and throughout the paper, we use calligraphic uppercase letters to denote sets and tables, e.g., $\mathcal{S}$, and bold letters to denote indicator functions, e.g., $\mathbf{F}$. We denote the routing tree rooted from g as $\mathcal{E}_g$ and the one rooted from c as $\mathcal{E}_c$. We also denote the route from c to the gateway as $\mathcal{R}_c$, with a hop count from c to g equal to $R_c = |\mathcal{R}_c|$, where $|.|$ is the cardinality of a set. We denote the node on the route of c and h hopes away from c as $\mathcal{R}_c(h)$. We refer to the parent node of c as pa_c, where $\mathrm{pa}_c = \mathcal{R}_c(1)$. Note that i is considered as a child node of c if $\mathcal{R}_i(1) = c$. We refer to the set of child nodes of c as $\mathcal{C}_c$. For the scheduling problem, we use ch_c^i to denote the child node of c that has the data generated at i. Table I provides the key mathematical symbols used in this paper.

Example: From Fig. 1-(a), we have $T_2^0 = 5$, $\mathrm{pa}_2 = 1$, $\mathcal{R}_2 = \{1, g\}$, $\mathcal{C}_2 = \{3, 4, 5\}$ and $\mathrm{ch}_1^3 = 2$. From Fig. 1-(b), (also to be explained later), $T^5_{(4\to 3)} = 4$.

III. Optimization for Scheduling

A. Optimization Problem

We write the objective functions to reflect quantities of interest. The schedule length is the time that the last packet

TABLE I: Table of key mathematical symbols

T_0^c	Release time of a packet of c
$\mathcal{N}$	Set of wireless nodes
$\mathcal{E}_c$	Tree rooted from c
$\mathcal{R}_c$	Route from c to g
$T_{\max}$	Schedule length
T_Δ	Maximum end-to-end delay
$T^c_{(i\to j)}$	Time node i forwards a packet of c to j
$\mathcal{I}_c$	Set of links that interfere with c's transmission
T^i_{LB}	Lower bound on schedule length of a sub-tree rooted from i
$\Delta_{\max}$	Effective delay
$\mathcal{C}_c$	Child nodes of c
pa_c	Parent node of c
ch^c_k	Child node of k that can forward a packet of c
t_c	Earliest arrival time of a packet of c

arrives at the gateway. Thus, we have

$$T_{\max} = \max_{c} \ T^c_{(ch^c_g \to g)}. \tag{1}$$

On the other hand, the E2E delay for a packet generated at c can be written as:

$$T^c_\Delta = T^c_{(ch^c_g\to g)} - T^c_0 + 1.$$

In this work, we are interested in its worst-case scenario, i.e.,

$$T_\Delta = \max_{c} \ T^c_\Delta. \tag{2}$$

Thus, our goal is to jointly minimize $T_{\max}$ and T_Δ. It is possible to construct examples such that minimizing one of the objectives does not necessarily minimize the other. In the next subsection, we write the full optimization problem as a multi-objective optimization problem, which is typically solved by scalarization, ordering, or other methods to find the Pareto-optimal solution [12].

In this work, we choose a lexicographic ordering method, since it reflects the envisioned problem, in which $T_{\max}$ is more important than T_Δ. Thus, the goal is to minimize the schedule length, and then minimize the maximum E2E delay. For example, the central controller might need to repeat the control-acquisition process frequently, such that a period ends when the last packet is received. Thus, the frequency of this process is identified by the schedule length. At the same time, data that is generated at the beginning of the acquisition time does not need to be delayed until the last packet is received. This raises the need to minimize T_Δ within a given schedule. Furthermore, note that in some cases, the minimum value for T_Δ can be achieved for the minimum $T_{\max}$. This can be verified easily for the uniform release times case. More relevant discussion will be made in the next section.

As described above, we first minimize $T_{\max}$. Let the indicator function $\mathbf{X}^c_{(i\to j)}[n]$ be equal to one if the data generated at c is scheduled to be transmitted through the link from i to j, $(i \to j)$, at time slot n, and zero otherwise. Similarly, let $\mathbf{F}_{(i\to j),f}[n]$ be equal to one if the channel f is assigned to link $(i \to j)$ at time slot n, and zero otherwise, where $i, j \in \widetilde{\mathcal{N}}$, and $f \in \{1, ..., L\}$. Then, we can form the optimization problem as follows:

(OPT-1)

$$\min \quad \max_{c\in\mathcal{N}} T^c_{(\text{ch}^c_g\to g)}$$

subject to:

$$\sum_c \mathbf{X}^c_{(\text{ch}^c_i\to i)}[n] + \sum_c \mathbf{X}^c_{(i\to \text{pa}_i)}[n] \le 1, \ \forall i \in \mathcal{N}, n = 1, ... \tag{3}$$

$$\sum_{f=1}^{L} \mathbf{F}^c_{(i\to\text{pa}_i),f}[n] - \sum_{c\in\mathcal{E}_i} \mathbf{X}^c_{(i\to\text{pa}_i)}[n] = 0, \ \forall i, \forall n \tag{4}$$

$$\mathbf{F}^c_{(i\to\text{pa}_i),f}[n] + \sum_{(k,j)\in\mathcal{I}_i} \mathbf{F}_{(k\to j),f}[n] \le 1, \ \forall i, \forall c, \forall f, \forall n \tag{5}$$

$$T^c_{(i\to\text{pa}_i)} - T^c_{(\text{ch}_i\to i)} \ge 1, \qquad \forall i \in \mathcal{R}_c, \forall c \tag{6}$$

$$T^c_{(c\to\text{pa}_c)} - T^c_0 \ge 0, \qquad \forall c \tag{7}$$

$$\sum_n \mathbf{X}^c_{(i\to j)}[n] = 1, \qquad \forall i \in \mathcal{R}_c \cup \{c\}, \forall c \tag{8}$$

$$T^c_{(i\to\text{pa}_i)} = \sum_n n\mathbf{X}^c_{(i\to\text{pa}_i)}[n], \quad \forall i \in \mathcal{R}_c \cup \{c\}, \forall c \tag{9}$$

The first constraint, eq. (3), specifies that the nodes are half duplex, and thus it guarantees that an active node is either a transmitter or a receiver in a given time slot n. The second constraint, eq. (4), enforces the assignment of a frequency channel to an active link. The third constraint, eq. (5), is used to avoid assigning the same frequency channel to two nodes that could interfere with each other on the same time slot. The fourth constraint, eq. (6), enforces the fact that a node cannot transmit a packet before it receives it. The fifth constraint, eq. (7), is related to the non-uniform release times, so that a node can transmit its packet only when the data is generated. The sixth constraint, eq. (8), enforces the condition that a packet generated at c must be transmitted once at each hop. Finally, the last constraint, eq. (9), assigns the index of the schedule time slot to the link. For space limit, we omitted the support for some of the indices when the context is clear, (e.g., in eq. 3). Note that we can convert the optimization problem (OPT-1) into a linear optimization problem as follows:

$$\min \quad \beta$$

subject to:

$$T^c_{(\text{ch}^c_g\to g)} \le \beta, \qquad \forall c$$

constraints in (OPT-1)

Thus, (OPT-1) can be written as an ILP. Once the value of β is determined, say β^*, we can use it in optimizing the T_Δ as follows:

(OPT-2)

$$\min \quad \left(\max_{c\in\mathcal{N}} T^c_{(\text{ch}^c_g\to g)} - T^c_0 \right)$$

subject to:

$$T^c_{(\text{ch}^c_g\to g)} \le \beta^* + \epsilon, \qquad \forall c$$

constraints in (OPT-1)

where ϵ is a non-negative integer that reflects the flexibility in the schedule length. Similar to the technique above, this

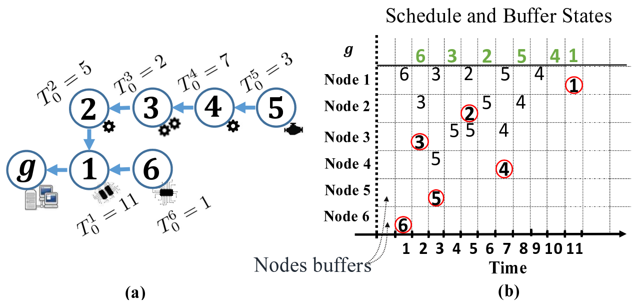


Fig. 1: An illustration to a network connected to processes. The network has a single sub-tree. (a) Network topology and release times (b) Buffer states over time for $T_{\max} = 11$ and $T_\Delta = 6$. The numbers in the buffers refer to the packet of type that correspond to that number, circulated numbers are released packets in that time slot. Note that packets can be transmitted on the same time slot as they released. A number in the buffer of g shows only newly received packets.

program can be converted into an ILP. The optimization problem has to be solved in two steps: First step determines the schedule length β^*. For a given ϵ, the second step determines the time slots when the links in route $\mathcal{R}_c$ are active, i.e., the values of $T^c_{(i\to \mathrm{pa}_i)}$, $i \in \mathcal{R}_c$, and the frequency assignments for these links, such that T_Δ is minimized. The ILP's are NP-hard in general and are usually challenging to solve due to the large number of constraints that hinders an efficient use of the ILP solvers. Thus, we provide delay analysis that could help designing an efficient scheduling scheme. Additionally, we utilize the bounds below to reduce the search space for the ILP which is used only for small networks in Sec. VI.

IV. Delay Analysis and Greedy Scheme on Subtree

We first consider a special case to get an insight about the general problem. Specifically, we consider that $|\mathcal{C}_g| = 1$, which means that the gateway has a single child node $\mathrm{ch}^c_g = \mathrm{ch}_g$, $\forall c$, i.e., the network has a single subtree rooted from ch_g.

A. Lower Bound on Schedule Length

We start by providing a lower bound on the schedule length.

Theorem 1: In the interference free sub-tree network, let $t_{(m)}$ be the smallest time to deliver the mth packet to the gateway. The schedule length for a network with N nodes is lower bounded by T_{LB}, where

$$T_{\mathrm{LB}} = \Delta_{\max} + 2N - 1$$

with an effective delay $\Delta_{\max}$ defined as

$$\Delta_{\max} = \max_{m} \quad t_{(m)} - 2m + 1 - \mathbf{1}_{(m<q)}$$

where q is arrival index for packet $c = \mathrm{ch}_g$.

Proof: See Appendix A. ■

Note that in the derivation of this bound, we assumed that interference is mitigated as in [3]. Interestingly, the derivation of the bound suggests that under the conditions provided in Theorem 1, any scheduling scheme that follows the order of earliest arrival time would achieve the lower bound on the schedule length.

B. Maximum End-to-End Delay

Although the schedule length and E2E delay are correlated, they are not identical. In a simple case, when all nodes have equal release time, e.g., $T^c_0 = 1$, $\forall c$, the minimum value of T_Δ is equal to $T_{\max}$. However, this is not the case for the general release time problem. For instance, the last arrival could be a packet generated at a node that is one hop away with a large release time. In addition, minimizing T_Δ could increase $T_{\max}$. Although this can be easily verified with examples in a *general* network structure, we next provide a condition to arrive to examples in sub-tree scenarios, which would prove to be valuable for proposing a greedy solution. Let us define the earliest arrival time of packet type c as follows:

$$t_c \triangleq R_c + T^c_0 - 1. \tag{10}$$

Let two packets generated at i and j have release times $T^i_0 \geq T^j_0$, and the delivery at g be such that $T^i_{(ch^i_g \to g)} < T^j_{(ch^j_g \to g)}$. Note that these conditions are necessary for Proposition 1 to avoid trivial conclusions. Additionally, let packet j has the maximum E2E delay, i.e., $T^j_\Delta = T_\Delta$. Then, we have the following proposition.

Proposition 1: Switching the order of packet delivery between i and j, a schedule that achieves the minimum schedule length for a given sub-tree reduces the E2E delay for packet j. However, this switching process *increases* the schedule length if

$$\min\left\{T^j_{(ch^j_g \to g)}, T^i_{(ch^i_g \to g)} + T^i_0 - T^j_0\right\} > t_j. \tag{11}$$

Proof: The proof is omitted for brevity. In fact, considering the case when switching order increases the schedule length, and using procedures similar to the derivation of Theorem 1, we can readily prove Proposition 1 in the interference free network. ■

Note that Proposition 1 considers a joint relation between packets i and j. However, the schedule length depends on other nodes, see Theorem 1. Additionally, note that T_Δ in the network might stay unchanged, since the increase in $T_{\max}$ could also increase the E2E delay for other packets.

Fig. 1 shows one example in a sub-tree along with a schedule when interference is mitigated. Note that $T_{\max} = 11$ (which is equal to the case of a uniform release times [3]), this can be also verified using Theorem 1. Also, note that the maximum E2E delay for packet $c = 5$ is six, i.e., $T_\Delta = T^5_\Delta = 6$. Noticing $T^2_{(1\to g)} = 6$ and $T^5_{(1\to g)} = 8$, we can reduce T_Δ from six to five by scheduling node 3 to forward packet 5 to node 2 in time slot 5 (instead of node 2 transmitting its packet to g). This will increase the schedule length to 12.

Finally, a simple lower bound on the maximum E2E delay is given by

$$T_\Delta \geq \max_{c} \ R_c. \tag{12}$$

C. Greedy Scheduling Scheme on Subtree

A greedy algorithm could prioritize the nodes that are far away from the gateway first. However, in that case $T_{\max}$ is unpredictable. Alternatively, based on the discussion in Sec. IV-B, we note that for two packets, i and j, when $T^i_0 \geq T^j_0$, and

$t_i = t_j$, we can switch the order without increasing $T_{\max}$ and possibly reduce T_Δ. When $t_i \leq t_j$, by switching the order of i and j, it is possible to reduce T_Δ without increasing $T_{\max}$, i.e., when (11) is not satisfied. However, to build a schedule based on that, which achieves our objectives, we need to iteratively verify the joint conditions and update the network schedule. Instead, we provide a simple node-based rule that utilizes the impact of sorting on the schedule length [3], and the insights above.

To minimize the schedule length, we propose the following scheme. For k that has large load (or has higher priority in general), we may schedule it as a transmitter or as a receiver. The latter case corresponds to switching the order between the oldest packet in k's buffer and the oldest packet in its children's buffer. Which might occur when the child node of k has the "older" packet.

Note that we do not restrict the buffer in k to be empty to schedule its child nodes. When the buffer of k is empty, all packets in $\mathcal{C}_k$ have possibly equal delivery time to g. Thus, prioritizing the one with the old packet is equivalent to swapping the order of arrival, which does not increase the schedule length when interference is mitigated.

However, when the buffer is not empty, defining "older" packet is tricky; it is possible that k could accumulate data in the buffer and misses chances for transmissions. Thus, it could increase the schedule length. To tackle this problem, we allow receiving more packets from the child nodes only when the packets are older than a threshold γ, which might depend on the node and/or the buffer state of the nodes. In the following, we use γ_{b_k}, where b_k represents the number of packets in the buffer of k. Thus, defining w_{loc} and w_{ch} to be the oldest release time in k's buffer and buffer of a child node of k, respectively, k transmits to pa_k if

$$w_{\text{loc}} - w_{\text{ch}} < \gamma_{b_k}.$$

The value for γ_{b_k} determines the trade-off between the increase in schedule length and the maximum E2E delay. As an example, in time slot 5 in Fig. 1, for $k = 2$, $w_{\text{loc}} = 5$ (its own packet), $w_{\text{ch}} = 3$ (packet type 5), and $\gamma_1 = 1$, node 2 is scheduled to receive a packet from node 3. Note that the value of γ_{b_k} is a design parameter to be investigated in the future work.

V. General Network

When the network has multiple sub-trees and/or only a limited number of frequency channels are available, the problem is more complicated to analyze. We instead generalize the lower bound and the scheme above. Defining $T_{\text{LB}}^{(i)}$ as the lower bound for the ith subtree, we have

$$\tilde{T}_{\text{LB}} = \max_i \; T_{\text{LB}}^{(i)}.$$

When it is required to schedule two or more conflicting nodes, an appropriate priority to the nodes should be used. In other words, we should decide which one can be scheduled first and thus reserve the resources. In this paper, we use a simple greedy graph coloring based on the priority of nodes. One possible ordering is based on the remaining load of a node. Nevertheless, we can use other metrics as well such as partial sorting or total load, see [3]. Next, we provide the general algorithm. We refer to the sorting technique as a "metric".

1: Initialize the schedule $\mathcal{S}$ and current time slot t = 0
2: **while** Receive packet at the gateway $<$ N **do**
3: $\quad \mathcal{Q} \leftarrow$ decreasing order of nodes based on the metric
4: $\quad$ Increment current time slot t = t+1
5: $\quad$ **for** k in $\mathcal{Q}$ **do**
6: $\qquad b_k \leftarrow$ size of load in the buffer of k
7: $\qquad w_{\text{loc}} \leftarrow$ smallest release time in the buffer of k
8: $\qquad \mathcal{C'}_k \leftarrow$ subset of $\mathcal{Q}_k$ that can transmit to k at time t
9: $\qquad w_{\text{ch}} \leftarrow$ smallest release time in the buffers of $\mathcal{C'}_k$
10: $\qquad$ **if** $w_k - w_{\text{ch}} \geq \gamma_{b_k}$ **then**
11: $\qquad\quad$ Determine ch_k corresponds to w_{ch}
12: $\qquad\quad$ Schedule link (ch_k, k)
13: $\qquad$ **else**
14: $\qquad\quad$ Schedule link (k, pa_k) if a channel is available and pa_k is not busy
15: $\qquad$ **end if**
16: $\qquad$ Update $\mathcal{S}$, sets of active links and busy channels
17: $\quad$ **end for**
18: **end while**

In the algorithm above, the steps to update the schedule $\mathcal{S}$, the set of active links and the set of used channels are omitted for brevity. Note that due to the limited number of frequency channels, only a subset of nodes may transmit in a given time slot. In line eight, we limit the search for nodes that may transmit to k by $\mathcal{C}'_k \subseteq \mathcal{C}_k$. Thus, we have the following:

$$w_{\text{ch}} = \min_{c \in \text{buffer of } \mathcal{C}'_k} T_0^c.$$

Finally, if L is large, when using the remaining load as the sorting metric, and assuming uniform release times the scheme above reduces to the one [3]. Using the above metric, a rough complexity analysis for this scheme is as follows. Taking the worst case that all the nodes have releases time at $\max_c(T_0^c)$, the schedule takes up to $\max_c(T_0^c + 2N - 1)$ time slots. Ignoring some constants, such as the number of child nodes per node, and noting that the load takes value in $\{0, ..., N\}$, we can assume that the sorting technique is $\mathcal{O}(N)$. Then we can show that the time complexity is $\mathcal{O}(\max_c(T_0^c N) + N^2)$. If $\max_c(T_0^c) = \mathcal{O}(N)$, then the complexity becomes $\mathcal{O}(N^2)$, which is polynomial in time. Note that if $\max_c(T_0^c)$ is large, many iterations can be skipped since their scheduling is not needed.

VI. Simulation

In this paper, we build experiments, using MATLAB, to investigate the performance of the following schemes:

- ILP: An implementation based on solving (OPT-1) and then using (OPT-2) with $\epsilon = 0$.
- LTA-0: An implementation of [3], where the schedule starts when all packets are released.
- LTA-1: A centralized implementation of LTA-0, where the controller schedules nodes according to the sorting metric with the maximum buffer size (BZ)=1.
- LTA-2: Same as LTA-1 with large buffer size.

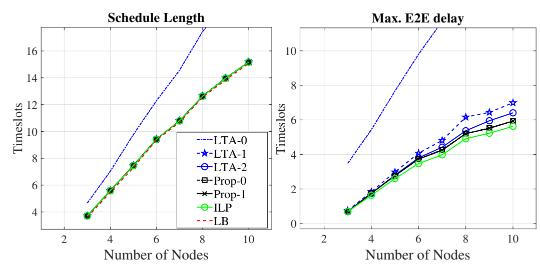


Fig. 2: Schedule length and maximum E2E delay for Case-1.

- Prop-0: An implementation of the proposed scheme, with $\gamma_0 = 0$ and $\gamma_x = \infty, \forall x > 0$.
- Prop-1: Same as Prop-0 with a gradual increase in γ; $\gamma_0 = 0, \gamma_1 = 3, \gamma_2 = 4, \gamma_3 = 5, \gamma_4 = 6, \gamma_5 = 100$.

In sorting the nodes, we prioritize the child of the gateway and sort them based on the remaining load. Other nodes are sorted based on total load, i.e., the total number of packets that a node has to transmit to its parent node. This reduces the complexity of the frequent sorting process. We perform the simulation for three main network structures:

- **Case-1**: A random network with two frequency channels, $L = 2$, and a small number of nodes.
- **Case-2**: A network with a single subtree and a large number of frequency channels, $L = 100$.
- **Case-3**: A random network with $L = 1$ and $L = 16$.

For each of these cases, we consider the schedule length and the maximum E2E delay. Unless it is stated otherwise, we consider the release time to be uniformly distributed in $[1, N]$, where N is the number of nodes in the network including the gateway. We also consider buffer size BZ $= 5$. Finally, the quantities are averaged over 500 network realizations.

A. Case-1

We have a small network with $N \in \{3, ..., 10\}$, in which the gateway is placed at the center of 20m ×20m area. The other nodes are uniformly distributed. The communication range of a node is 5m. Note that the network size and small communication range may not be practical, but meant for schemes demonstration. Fig. 2 shows the average schedule length and maximum E2E delay over 100 network realizations. In this figure, all the schemes except LTA-0 achieve a minimum schedule length. A small gap compared with the lower bound (LB) is observed due to a limited number of channels and the network structure, as discussed in Section V. For the E2E delay, the proposed schemes outperform the other schemes except the ILP. Interestingly, the buffer size seems to play a key role in minimizing the maximum E2E delay as we see LTA-2 versus LTA-1.

B. Case-2

We consider a network with nodes distributed in $450\text{m}\times450\text{m}$ region, with nodes range from 3 up to 200. The communication range is $50m$. The gateway is placed at the center. We enforce a single subtree network. We assume that there are a large number of channels, (i.e., enough to eliminate interference). Fig. 3-(a) shows that the performance gap, with respect to the LB on the schedule length, is negligible for all schemes except LTA-0. Similar to above, the proposed schemes provide a smaller T_Δ compared to the other schemes. The best performance is achieved using Prop-1. In contrast to Case-1, LTA-1 has a smaller T_Δ compared to LTA-2, since increasing the buffer size could break the gradual acquisition structure [3]. Nevertheless, as it is shown in Fig. 2 and shown later, this is not true for a limited value of L.

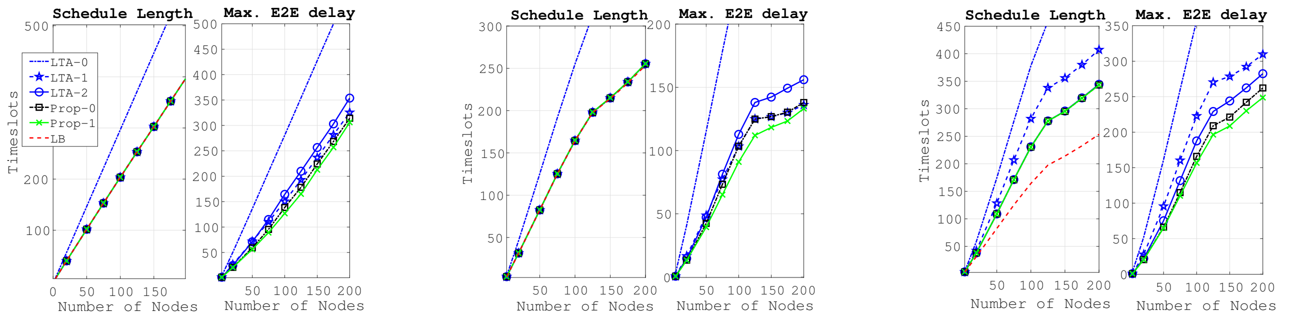
C. Case-3

A network realization is similar to Case 2. However, we allow the gateway to have more than one neighbors. For L = 16, in Fig. 3-(b), a small $T_{\max}$ is still achieved with all schemes except LTA-0. We notice a very small increase in the gap compared with its lower bound. For the T_Δ, the order of the schemes is similar to Fig. 3-(a). However, we notice that performance of LTA-1 coincides with those of the proposed schemes for a larger number of nodes. This indicates that the limiting behavior of the proposed schemes is similar to LTA-1, as network density increases. This phenomenon occurs earlier for Prop-0 starting at medium density. Recall that Prop-0 only accepts packets when its buffer is empty. Fig. 3-(c), shows the results when $L = 1$. We first notice that the gap between the LB and the scheme increases. Additionally, we notice that LTA-1 suffers from an increase schedule length due to the buffer restriction. For T_Δ, we notice similar behavior compared to Fig. 2. In general, Prop-1 shows a better T_Δ performance, which may result in a small increase in $T_{\max}$, as discussed in IV-C.

Finally, Fig. 4 shows the impact of buffer size on the performance for fixed values of $L = 16$ and $N = 100$. When BZ $= 1$, LTA-1 outperforms all other schemes. However, as the BZ increases, we notice a slight increase in the schedule length and decrease in the maximum E2E delay. Note that we used $\gamma_x = 11$ for $x \geq 5$ for Prop-1.

VII. Conclusion

We have considered the multi-hop data acquisition for the multi-channel delay sensitive convergecast network. We have proposed schemes considering a non-uniform packet release time at the nodes. The goal is to jointly minimize the schedule length and maximum E2E delay. We have first formulated the problem as a multi-objective integer program, then studied the schedule length and maximum E2E delay and derived a lower bound. Based on the analysis, we have proposed a new scheme that uses node sorting to prioritize nodes, and threshold based test to allocate channel and determine if the node is scheduled as a transmitter or a receiver. The simulation results have shown that the proposed scheme provides a relatively good reduction in the maximum E2E delay while maintaining a small schedule length irrespective of the number of available frequency channels. Notably, the proposed threshold based test has shown to provide a trade-off between the schedule length and maximum E2E delay.



(a) (b) (c)

Fig. 3: Schedule length and maximum E2E delay for (a) Case-2 with $L = 100$ (b) Case-3 with $L = 16$ (c) Case-3 with $L = 1$.

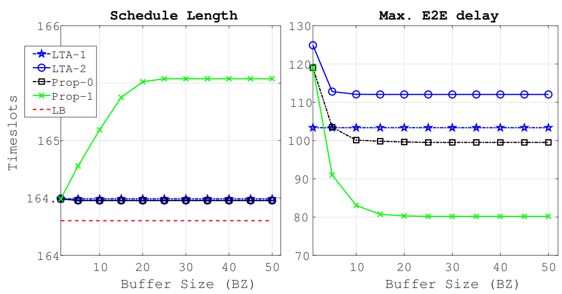


Fig. 4: Impact of buffer size on the schedule length and maximum E2E delay for $N = 100$ and $L = 16$.

APPENDIX A: DERIVATION OF THEOREM 1

We start with the following facts that can be verified easily:

(i) All the packets have to pass through ch_g.

(ii) Due to the half-duplex transceivers, delivering a packet type c to g through ch_g requires at least two time slots, except when $c = \text{ch}_g$, which needs at least one time slot.

(iii) A packet type c requires R_c hops to reach g.

(iv) The transmission of packet c cannot start before T_0^c.

Using (iii) and (iv), a packet generated at c cannot be delivered earlier than $R_c + T_0^c - 1$. For instance, the time when the first packet delivered to g is given by $t_{(1)} = \min_c \ (R_c + T_0^c - 1)$. Thus, $\{t_{(1)}, t_{(2)}, ..., t_{(N)}\}$ defines a non-decreasing sequence by construction. Using facts (ii), the mth packet has to wait $t_{(m-1)} + 2$ time slots to be delivered to g if the mth packet does not correspond to $c = \text{ch}_g$. However, it takes $t_{(m-1)} + 1$ time slots if the mth packet corresponds to $c = \text{ch}_g$. Let $T_{(m)}$ be the actual time that the mth packet delivered to the gateway. Then, we have $T_{(N)} \geq \max(t_{(N)}, T_{(N-1)} + 2 - \mathbf{1}_N)$, where $\mathbf{1}_N$ is equal to one if the Nth element corresponds to $c = \text{ch}_g$. We can rewrite this with the lower bound on T_{N-1} as follows:

$$T_{\text{LB}} = \max\left(t_{(N)}, \max(t_{(N-1)}, T_{(N-2)} + 4 - \mathbf{1}_{N-1} - \mathbf{1}_N)\right)$$
$$= \max(t_{(N)}, t_{(N-1)} + 2 - \mathbf{1}_N, T_{(N-2)} + 4 - \sum_{i \in \{N-1, N\}} \mathbf{1}_i).$$

We can do this iteratively then add and subtract $2N - 1$ from each term. Thus, we can have

$$T_{\text{LB}} = \max(t_{(N)} - 2N + 1, t_{(N-1)} + 2 - \mathbf{1}_N - 2N + 1,$$
$$\ldots, t_{(1)} + 2(N-1) - \sum_{i \in \{2, \ldots, N\}} \mathbf{1}_i - 2N + 1) +$$
$$2N - 1$$
$$= \max(t_{(N)} - 2N + 1, t_{(N-1)} - 2(N-1) + 1 - \mathbf{1}_N,$$
$$\cdots, t_{(1)} - 1 - \sum_{i \in \{2, \ldots, N\}} \mathbf{1}_i) + 2N - 1.$$

Define Δ_m as

$$\Delta_m = t_{(m)} - (2m - 1) - \sum_{i \in \{m+1, \ldots, N\}} \mathbf{1}_i$$
$$= t_{(m)} - (2m - 1) - \mathbf{1}_{m<q}. \qquad \text{(A.1)}$$

If $m = N$, then we consider $\{m+1, ..., N\}$ as an empty set. We defined q as the order of the packet $c = \text{ch}_g$, i.e., $t_{(q)}$ is the arrival time of $c = \text{ch}_g$, and $\sum_{i \in \{m+1, \ldots, N\}} \mathbf{1}_i = \mathbf{1}_{m<q}$. Thus, we can obtain

$$T_{\text{LB}} = \max(\Delta_N, ..., \Delta_1) + 2N - 1 = \Delta_{\max} + 2N - 1$$

where $\Delta_{\max} \triangleq \max(\Delta_N, ..., \Delta_1)$ is network *effective delay*.

REFERENCES

[1] D. D. Guglielmo, S. Brienza, and G. Anastasi, "IEEE 802.15.4e: A survey," *Computer Communications*, vol. 88, pp. 1–24, 2016.

[2] D. Chen *et al.*, "WirelessHART and IEEE 802.15.4e," in *IEEE Int. Conf. Ind. Technol*, 2014, pp. 760–765.

[3] O. D. Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 1, pp. 86–99, 2012.

[4] R. Soua, P. Minet, and E. Livolant, "Wave: A distributed scheduling algorithm for convergecast in IEEE 802.15.4e TSCH networks," *Trans. on Emerg. Telecom. Tech.*, 2015.

[5] M. R. Palattella *et al.*, "Traffic aware scheduling algorithm for reliable low-power multi-hop IEEE 802.15.4e networks," in *IEEE 23rd Int. Symp. PIMRC*, 2012, pp. 327–332.

[6] S. Gandham, Y. Zhang, and Q. Huang, "Distributed time-optimal scheduling for convergecast in wireless sensor networks," *Comp. Netw.*, vol. 52, no. 3, pp. 610–629, 2008.

[7] I. F. Akyildiz and I. H. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Netw.*, vol. 2, no. 4, pp. 351–367, 2004.

[8] R. Soua, P. Minet, and E. Livolant, "MODESA: An optimized multi-channel slot assignment for raw data convergecast in wireless sensor networks," in *IEEE 31st Int. Perform. Comput. Commun. Conf.*, 2012, pp. 91–100.

[9] D. Gong and Y. Yang, "Low-latency SINR-based data gathering in wireless sensor networks," *IEEE Trans. on Wireless Commun.*, vol. 13, no. 6, pp. 3207–3221, 2014.

[10] A. Saifullah *et al.*, "Real-time scheduling for wirelessHART networks," in *IEEE Real-Time Syst. Symp.*, 2010, pp. 150–159.

[11] O. Gaddour and A. Koubâa, "RPL in a nutshell: A survey," *Comp. Netw.*, vol. 56, no. 14, pp. 3163–3178, 2012.

[12] M. Ehrgott, *Multicriteria optimization.* Springer Science & Business Media, 2006.