Master of Science

A SCHEDULING METHOD BASED ON PACKET COMBINING TO IMPROVE END-TO-END DELAY WITH CONSTRAINED LATENCY FOR TSCH NETWORKS

The Graduate School of the University of Ulsan

Department of Electrical and Computer Engineering

DINH LOC MAI

A SCHEDULING METHOD BASED ON PACKET COMBINING TO IMPROVE END-TO-END DELAY WITH CONSTRAINED LATENCY FOR TSCH NETWORKS

Supervisor: Professor KIM Myung Kyun

A Thesis

Submitted to

The Graduate School of the University of Ulsan

In partial Fulfillment of the Requirements for the Degree of

Master of Science

By

DINH LOC MAI

Department of Electrical and Computer Engineering

University of Ulsan, Korea

JUNE 2020

DEDICATE TO

My Parents, My Sisters,

Those who find satisfaction in the fulfillment of this work

A SCHEDULING METHOD BASED ON PACKET COMBINING TO IMPROVE END-TO-END DELAY WITH CONSTRAINED LATENCY FOR TSCH NETWORKS

This certifies that this Thesis of Mai Dinh Loc is approved by:

Committee Chair: Professor Seok Hoon Yoon:

____

Committee Member: Professor Insoo Koo:

____

Committee Member: Professor Myung Kyun Kim:

____

Department of Electrical and Computer Engineering

University of Ulsan

May 2020

ABSTRACT

Wireless sensor networks (WSN), are new evolution for Internet of Thing (IoT) application to monitor real-time environmental status, especially low power network in the field of industrial area. The evolution through the year of the wireless technology give a number of possible applications of such technology in different fields of life such as agriculture, smart home, smart city and industrial monitoring. For example, in agriculture, the application collects humidity, temperature and soil moisture from different vegetable gardens periodically to monitor each garden's status at the central station. Based on this information, we took further actions to improve the crop yield with low-cost. In tunnels, the application collects carbon dioxide gas levels from different places periodically to monitor the carbon dioxide gas level status; based on this information, the application notified unsafe places in tunnels to employees who work in these tunnels.

In this study, we proposed heuristic scheduling algorithm that's based on data-aggregation and prioritizes each packet transmission dynamically based on its laxity (i.e., the remaining time before the end-to-end deadline) with different period apply to TSCH network to increase schedulability, reduce number of data transmissions, E2E delay, high reliability.

ACKNOWLEDGEMENT

I would like also to deeply thank my supervisor Prof. KIM Myung Kyun for his paralleled and continuous support during my study period. His tireless desire to inculcate the knowledge was more than helpful throughout my period of study.

I also would like to thank all the lab members of CNLAB at UoU for support each other and great collaboration in whatever circumstances we were working through.

I would like to thank my parents for their long-lived support, love, and prayers that made me quickly recover from discouragements during my Master study. I would like to thank my elder sister for being a source of happiness and encouragement in my life

Republic of Korea, Ulsan, JUNE 2020

DINH LOC MAI

Table contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1: INTRODUCTION

1.1 General Context

Wireless sensor networks (WSN), are new evolution for Internet of Thing (IoT) application to monitor real-time environmental status, especially low power network in the field of industrial area. The evolution through the year of the wireless technology give a number of possible applications of such technology in different fields of life such as agriculture, smart home, smart city and industrial monitoring[1][2][3][4]. For example, in agriculture, the application collects humidity, temperature and soil moisture from different vegetable gardens periodically to monitor each garden's status at the central station. Based on this information, we took further actions to improve the crop yield with low-cost. In tunnels, the application collects carbon dioxide gas levels from different places periodically to monitor the carbon dioxide gas level status; based on this information, the application notified unsafe places in tunnels to employees who work in these tunnels [5].

The complexity in wired communication can be found in any other field such as smart cities, environmental monitoring and especially in the industrial area where the evolution of smart industries implies indisputably wireless communication

The applications which use Wireless Sensor Networks (WSNs) technology in the industrial area have been increasing such as monitoring applications, radiation check, leakage detection, distributed and process control [4][5][6]. Real-time applications are sensitive to the delays and demand high communication reliability as well as a satisfaction of a considerable scalability. The end-to-end delay is constrained by upper bounds which varies according to the field of applications; tens of milliseconds for discrete manufacturing, seconds for process control and minutes for asset monitoring. To solve those challenges, wireless technology approaches which reduce the cost of wired installation, have been considered.

Timeslot Channel Hopping (TSCH) network is low-power network which uses the IEEE 802.14.5e standard for Media Access Control (MAC) protocol [7]. Time slotted operation is a well-proven approach to achieve highly reliable low-power networking through scheduling and channel hopping. It combines time slotted access with multi-channel and channel hopping capabilities. The development of this standard took into consideration the requirements of Wireless Personal Area Networks (WPAN) either in home or industrial area. It operates in the 2.4GHz ISM frequency band and supports various network topology such as star, mesh and tree. The IETF working Group 6TISCH [8] has been working on the standardized mechanisms to run IPv6 on top of the TSCH. In the 6TiSCH architecture, low-power wireless devices build a multi-hop Low power and Lossy Networks (LLN), which will be connected to Internet through one or more LLN Border routers [9].

Latency is very important for real-time application in the industrial. Some study work on scheduling for wireless sensor network to improve reliability of data transmission, latency such as

- P-CLLF [10], CLLF[11] is centralized scheduling for tree topology in which sensor nodes have different period. However, this scheduling have a lot of number of transmission and with tight period, schedulability is low.
- [12], [13] are data aggregation algorithm to improve latency and reduce number of transmissions for wireless sensor network, in which sensor nodes transmit data packet periodically. However, all nodes have same period.

Latency is very important for real-time application in the industrial. Some study work on scheduling for wireless sensor network to improve reliability of data transmission, latency such as. As our set of scenarios shown our proposal improves schedulability with tight-latency and reduces number of transmission and reduces E2E delay compare with existing approaches in PCLLF and CLLF.

1.2 Problem Statement

In Wireless Sensor Network (WSN), sensor nodes sense data from the environment, the main purpose is that all the data from all sensor nodes are collected by a central station (or node) and processed locally or by an external entity such as an application on cloud server. In wireless sensor network, MAC protocols can be classified into two categories: contention-base such as CSMA/CS, SCSMA [14] and scheduled-base such as [10][11]. Scheduling algorithm in a wireless sensor network can be classified into two categories: distributed [15] and centralized [11] . Wireless sensor networks have several advantages of being deployed in the industrial application zone, however, there are some issues which need to be considered as following:

- Reliability: In industrial networks, the reliability of data transmission is one of most important priority when it comes to designing the system. In WSNs, however, packet can be loss which occurs frequently. The wireless signal is attenuated by the environment such as blocks, machines or fading that is one of reason. Especially, the biggest reason which is the collision of the packets because the packets are transmitted simultaneous. The packet loss not only proves to be energy-inefficient but also creates system instability. Therefore, the reliability of data transmission is one of the most important factors as well as the avoidance of collision between the sensor nodes in the network, and even from the hidden node.

- Energy consumption: In industrial applications, a monitoring and control system has been maintained period to guarantee the stability of the system. In low-power network, energy consumed efficient is very important to increase life time of sensor node. Thus, each sensor communication device should have working time to meet the maintenance period of the system. However, it is not easy to achieve efficient energy consumption because of the collision.
- Latency: In real-time application, latency is very important to monitor environment status such as temperature, humility. To guarantee latency requirements, it needs scheduling for each transmission to reduce latency.

1.3 Contribution

Following the discussion above, the thesis work aims to:

- Proposed a scheduling algorithm that's based on dynamically packet combining and prioritizes each packet transmission dynamically based on its laxity (i.e., the remaining time before the end-to-end deadline) apply to TSCH network to increase schedulability, reduce number of data transmissions, E2E delay.
- Evaluating the method by comparing with other existing one to make sure that it is working perfectly, and all nodes can receive the schedule table in real-time.

1.4 Thesis outline

The remainder of this thesis is organized as follows. Some backgrounds are described and an overview on certain related works will be given in chapter 2. A brief description of the PC-PCLLF is given in Chapter 3. Performance of our approach is evaluated in chapter 4. The conclusion will be given in Chapter 5.

Chapter 2: BACKGROUND

This chapter provides an overview TSCH network and related work. First, we describe background of TSCH network.

2.1 TSCH Network

Time Slotted Channel Hopping or Time Synchronized Channel Hopping (TSCH) is a channel access method for shared medium networks. TSCH can be known as a combination of Time division multiple access and Frequency-division multiple access mechanisms. It uses diversity in timeslot and frequency to increase reliability [16].

Wireless communications are often referred as unreliable due to the unpredictability of the wireless medium. While wireless communication technique has many advantages such as no wires maintenance, costs reduction and low-power. The lack of reliability slows down the adoption of wireless networks technologies.

TSCH aims is reducing the impact of the wireless medium unpredictability to improve reliability of low-power wireless networks. Every node shares a schedule that allows nodes to know when node wakes up or sleeps to communicate other node so it is good at saving energy of node.

Channel Hopping

TSCH network supports 16 channels on the 2.4 Ghz band. Each used channel is identified by a channel offset. For each scheduled cell, the schedule specifies a slotOffset and a channelOffset that is timeslot when node wake up and channel which node uses in this timeslot. When sender transmits packet to receiver on channelOffset 5, receiver receives packet from sender on the same channelOffset 5. The channelOffset is translated by both nodes into a frequency using the following function:

$$\text{Frequency} = F\ \{(\text{ASN} + \text{channelOffset}) \bmod n_{ch}\}$$

Where Absolute Slot Number (ASN) is the total number of slots that elapsed since the network was started. The ASN is incremented at each slot and shared by all devices in the network. n_{ch} is number of channel which is used in this network.

In the Fig. 3, we can see an example of assignment timeslot and channel given a certain network. We can see that two transmissions can be assigned at two different channels in the same timeslot as long as they are not in duplex conflict and also one transmission can be assigned two different timeslots within the same slotframe. The assignment depends on the chosen scheduling process.

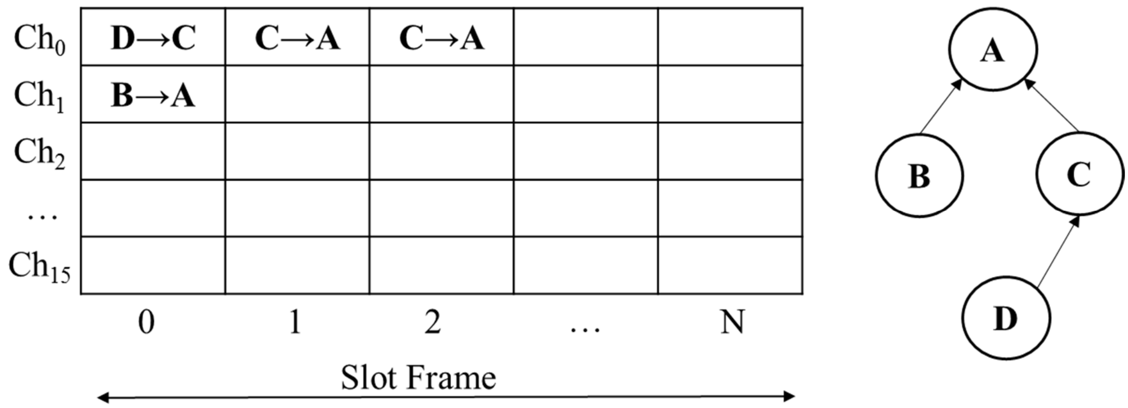

Figure 1: Example of Timeslot and Channel Assignment for a certain Network

Multipath-Fading Mitigation

Multipath propagation can create internal destructive interferences of a wireless signal known as multipath fading [17]. This phenomenon can be overcome by shifting the location of the communicating nodes or by switching the communication carrier frequency.

The channel hopping mechanism of TSCH allows to reduce the impact of multipath fading by changing the communication carrier frequency in each timeslot.

2.2 Related work

A packet combining technique is used in data gathering which is collected from sensor nodes in WSNs which aims to reduce energy consumption, reduces the number of transmissions, latency, and energy consumption.

With constrained latency, some studies work to improve latency and E2E delay such as PCLLF[10] and CLLF[11].

- PCLLF [10], CLLF [11]: Author propose scheduling algorithm for network which nodes has different timeslot to adapt latency requirement to apply to real-time application in industrial. This algorithm base on dynamic prioritization of transmission which is defines by algorithm. The priority of transmission is depend of number duplex conflict. Based on priority, algorithm chooses which transmission which is scheduled in this timeslot. There is non-collision so it is high reliability. However there is many transmission in the network.

Some studies work with data-aggregation to reduce data traffic with centralized scheduling TEACSA[18], DAHDA[19], OCCN[20] and distributed scheduling DAS[21], PPSDA[22].

- With centralized scheduling:
  - TEACSA[18]: the authors proposed scheduling approach which is based on the concept of an aggregation tree. A time slot scheduling method is used so this approach minimizes the time required for data gathering by aggregation convergecast. The value of the transmission is determined by sum of number of child and receiver's depth. The tree is constructed by appending to it the path which has least overhead in each iteration. To supplement the scheduling algorithm one neighbor degree ranking scheme is implemented to assign time slots to the SNs in an effective manner[23]. It reduces data traffic, however, data transmissions is not periodically and latency of each node is not constrained.
  - DAHDA[19] the authors proposed a new algorithm namely Dynamic Adaptive Hierarchical Data Aggregation (DAHDA) which is extended the functionality of LEACH-C. In DAHDA, all of clusters are assigns weights which based on the residual energy and density. Based on the weights, the algorithms decide which node is Cluster Heads (CHs) where data packet is combined. It improves energy consumption because it reduces data traffic, however, data transmissions is not periodically and latency of each node is not constrained.
  - OCCN[20] the authors proposed a new method namely Optimal Clustering in Circular Networks (OCCN) which aims to reduce energy consumption, data traffic and increase the lifetime of wireless sensor networks[24]. In that method, which was proposed for a circular area surrounding a sink, one hop communication between the cluster heads and the sink was replaced by an optimal multi-hop communication. Data packets are combined at cluster head nodes. It reduces number of traffic, however, data transmission is not periodically and latency of each node is not constrained.
- With distributed scheduling:
  - DAS[21] the authors proposed distributed energy-balance algorithm which aims to balance the energy consumption for aggregators. In this algorithm, first, the forms trees rooted at nodes which are termed virtual sinks, balances the number of children at a given level to level the energy consumption and then try to assign timeslot for all nodes to satisfy minimum latency and energy consumption. Data packets are combined at parent node. This algorithm reduces latency and data traffic, data transmission is not periodically and latency of each node is not constrained.
  - PPSDA[22] the authors proposed a privacy-preserving in-network aggregation protocol for wireless sensor network which is based on the concept of data slicing, mixing and merging. Sensor nodes slice its data to some pieces then send this pieces to neighbor nodes which has same key with this node. After receiving pieces data from neighbor nodes, this

node combines its own data then sends to sink node. This algorithm don't reduces number of traffic and is low reliability.

We proposed scheduling which is based on dynamically packet combining and dynamically prioritizes the packets with constrained latency.

Chapter 3: ALGORITHM

3.1 System Models

We consider TSCH network in which there is number of sensor nodes, one sink node and one network manager.

In TSCH, we defined periodic cycles which is called slotframes, as shown in Fig. 1. In one slotframe, it has fixed number of timeslot which is defined as length of slotframe as shown in Fig 2: N_S. A single time slot length used IEEE.802.154e standard is 10ms that is long enough time for the transmitter to send a maximum-length packet which is 127 bytes, and receiver transmits ACK packet if required as shown in Fig 3. In TSCH, Absolute Slot Number (ASN) is used to determine how many timeslot has already elapsed from network started. There is 16 available channel which is used to communicate between nodes which of value is from 11 to 26. In each timeslot, the transmitter transmits packet after TXOffset value and the receiver starts listening packet after RXoffset value from beginning of this timeslot as shown in Fig 3. This mechanism requires the different clock time when two nodes wakeup is less than Guardtime to able to receive packet. In TSCH, we define assignment cell (timeslot, channel) which is timeslot node wakes up and channel which node uses to transmit or receive packet. To eliminate collision and interference, we use only one transmission for one cell assignment that's mean an assignment cell is used for link v_iv_j. In one timeslot, maximum of parallel transmissions equals number of channel which is used.

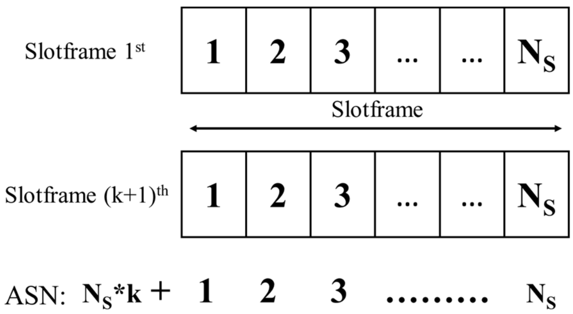

Figure 2: A TSCH superframe consisting of N_S time slots and the transmission structure in a slotframe.

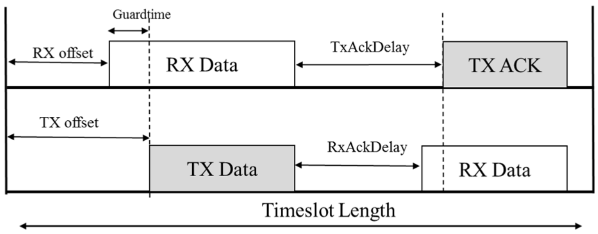

Figure 3: Timing in one timeslot in TSCH network

3.2 Application Model

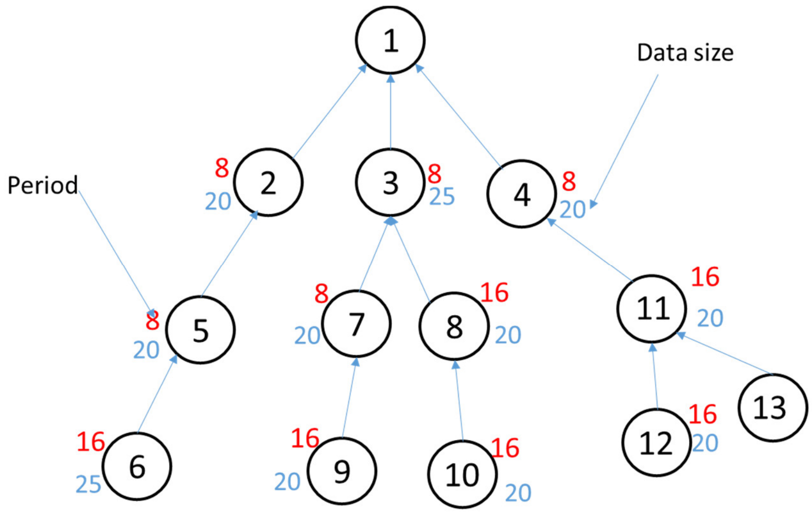

Figure 4: Network example

In Fig 4, it shows network example in which period value of each node that is number of timeslot node generates a packet and data size that is data length that is this node has to transmit to network manager.

In our paper, the TSCH network is formed as tree network modeled which is a Directed Acyclic Graph (DAG) G = (V,E), where V={v_0, v_1, v_2, …, v_n} is the set of all devices (sensor nodes) and arcs in E are links. The link in E is identified by a pair of nodes that are parent and child nodes, for example link vivj where v_i is child node and v_j is parent node and data packet is transmitted from vi to vj. In the network, packet collision can occur when duplex conflicting links uses same timeslot. Two links are duplex conflicting links when they use same sender or receiver node. The set of all links duplex conflicting with v_iv_j is CLNF(v_iv_j), for example CLNF(v_7v_3) = {v_9v_7, v_3v_1, v_8v_3} in Fig. 4. Packet collision can occur when interfering links uses same timeslot and channel. Two links are interfering links when two receiver nodes are neighbor nodes.

Node v_0 is sink node which is central controller, other nodes are sensor nodes or actuators. Each sensor node senses environment status such as temperature, humidity then transmit periodically to sink node. Each node can only receive or transmit packet concurrently and support multiple non-overlapping channels.

In period of transmission, each node v_p generates a flow F_p and F = {F_1, …, F_n} is set of all of flows in this period. Because period of nodes can be different, so number packet of node has to transmitted to sink node can be different. Each node is determine period of node has to transmit packet. When start this period, all nodes generate packet. Each node has fixed data length to send to sink node and data length can be different value between nodes. Each flow F_i is initially released periodically with period

T_i and starts from beginning of slotfame. The j^{th} instance of Fi in a slotframe transmits the j^{th} packet which is denoted by $p_{i\,j}(L)$ where L is data length that packet contains. The released timeslot of F_i is at timeslot (j-1)*T_i in a slotframe. The transmission of $p_{i\,j}$ is from node v_s to v_r where node v_r is k hop away from sink node and has data length which is L bytes, it is named by $T_{i,k}^{j}(L)$. U is set of transmission of all packets that be scheduled in the network. For example in Fig.1, with F_3 there is transmission list which is $T_{3,1}^{1}(30)$, $T_{3,0}^{1}(30)$. The transmission $T_{3,0}^{1}$ can be released after transmission $T_{3,1}^{1}$ finished. The length of slotframe is determined by Least Common Multiple (LCM) of periods of all nodes in the network. Number packet which is generated by F_i in one slotframe is determined by LCM/T_i. For example with node 3, we have $T_3 = 16$ and LCM = 16 so number of packet which F_3 must send to sink node is 16/16 = 1.

For example in fig 2, F_3 has deadline which is D_3 which means packet $p_{3,1}$ that is packet from node 3 must arrive at sink node at the latest timeslot D_3. We defined Di which is deadline of flow F_i, each packet $p_{i,j}(L)$ generated by F_i must arrive at the destination before (j-1)*T_i+ D_i. Packet from F_i can combine with other flow base on combining mechanism.

The combined packets are denoted $TC_{i,k}^{j}(L)$ with v_j is node has data in this packet which has highest hop-count, and k is hop away from receiver this packet to sink node.

With F_6 in Fig.4, we have number packet that node 6 generates in one slotframe is 16/16 = 1 so set of transmission which is referenced packet from node 6 to sink node (node 1) is {$T_{6,2}^{1}$, $T_{6,1}^{1}$, $T_{6,0}^{1}$} as shown in Fig. 5

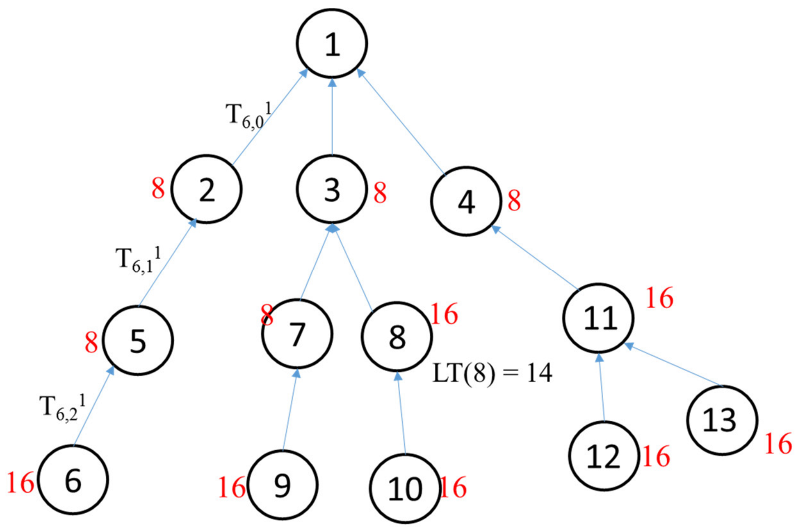

Figure 5: Transmissions which node 6 generates in one slot frame

$T_{6,2}^{1}$ that is referenced that packet from node 6 is transmitted from node 6 to node 5. $T_{6,1}^{1}$ that is referenced that packet from node 6 is transmitted from node 5 to node 2. $T_{6,0}^{1}$ that is referenced that packet from node 6 is transmitted from node 2 to node 1

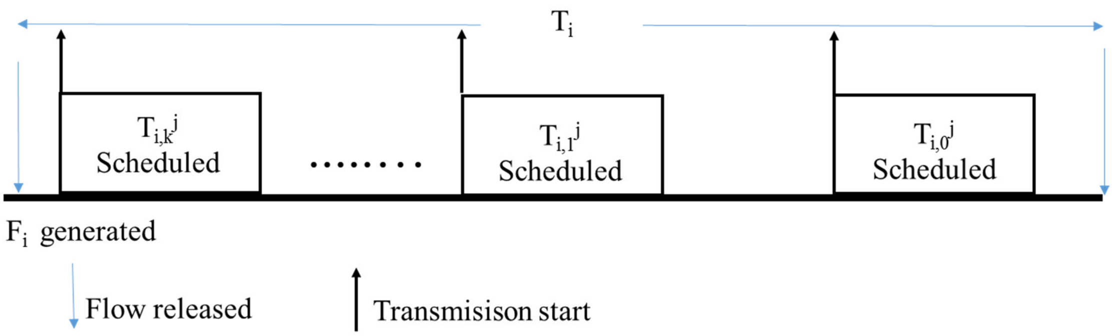

Figure 6: Transmissions of first packet generated by flow Fi

3.3 Grouping node mechanism.

We proposed grouping node mechanism that is dynamically grouping node which based on data length of each packet, earliest combining timeslot and latest timeslot to determine which packet can combine with other packet and which node will combine packets to reduce of number of transmissions and increase schedulability. This progress is repeated each timeslot**.** Condition for determining which node will combine packet with other packet.

- Total data size in subtree at this time slot where root node is the node we consider is less than Maximum Transmission Unit (MTU).
- LT(i) - ECT(i) >= NCi where NCi is number of child node which has packet in queue this timeslot of node i.

We proposed ECT(i) is earliest combined timeslot when all of packet in subtree node i can arrive to node i.

- ECT(i) = MAX(ETR(j)) + NCi where j is child of node i and NCi is number children of node i.
- ECT(i) = current timeslot if node is highest hop-count to sink node when node i has packet in a queue in the path contains node i.

We proposed LT(i) that is latest timeslot node i must to transmit packet to parent of node i to guarantee schedulability.

LT(i) depend deadline of packet in the subtree.

LT(i) = MIN ($D_{i,j}(k,t)$) with node k in subtree i

In timeslot 0, all nodes have packet in the queue. Fig.5 showed ECT value of all node in the network in timeslot 0.

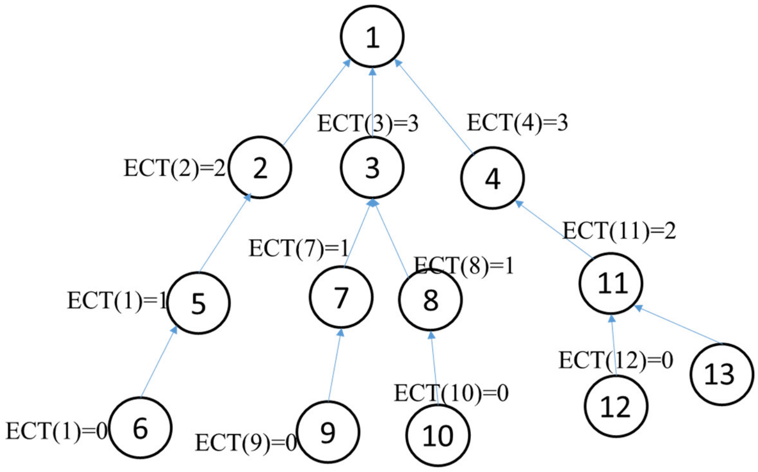

Figure 7: Earliest combining timeslot for all node in timeslot 0

In Fig. 8, we show LT value of all node in the subtree.

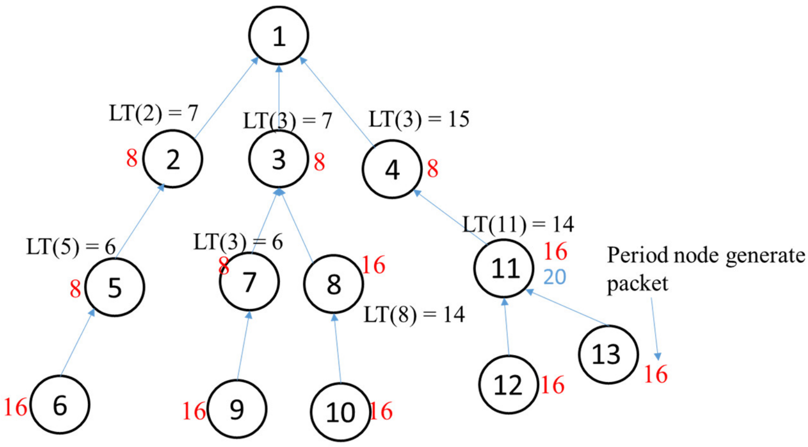

Figure 8: Latest timeslot for all node in timeslot 0

In 802.15.4.e standard, we have maximum packet size is 127 bytes and MAC header length is 21 bytes so MTU = 127-21 = 106 bytes. We choose MTU = 100 bytes. In Fig. 4, at timeslot 0, example with node 5 base con condition which we proposed above. We have total data size in subtree 5 is 45 bytes < MTU = 100 bytes and LT(5) - ECT(5) = 6-1 >= 1 so data node 5 can combine with packet node 6. With node 2, we have total data size in subtree is 65 bytes < MTU= 100 bytes and LT(2) - ECT(2) = 7 - 2 = 5 >= 1 so data from node 2 can combine with data from node 5,6. We repeat this action with all of parent node which child of this node has packet in the queue and we have result about which data packet com other and which node will combine packet.

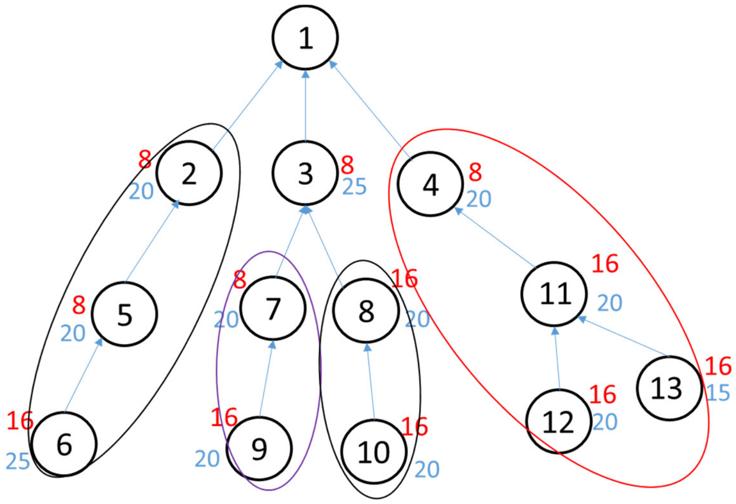

Figure 9: Group node results at timeslot 0

In Fig 9, at timeslot 0, it shows data packet node 5 will combine with node 6 at node 5.After node 5 receiving packet from 6, this packet will be combine with packet which is generated at node 5 as explained next section. Data from node 2 will combine with nodes 6, 5 at node 2, data from node 7 will combine with node 9 at node 7, data from node 8 will combine with node 10 at node 8, data from node 11 will combine with node 12, 13 at node 11, data from node 4 will combine with node 11, 12, 13 at node 4. This action is described in next section.

3.4 Priority Assignment

In this section, we use priority value for each transmission it is based on remaining of available timeslot which node can be scheduled to increase schedulability of this case. Higher remaining of available timeslot that means this transmission is easier to schedule within limited laxity to deadline. It indicates lower priority, and has larger space for transmission which can be scheduled.

To determine priority value of each transmission, we use timeslot window for each transmission $T_{i,k}^{\ j}$ or $TC_{i,k}^{j}$ at timeslot t is defined $TW(T_{i,k}^{\ j}, t)$ it is updated every timeslot. The timeslot window shows when transmission can be released and latest timeslot this transmission must be scheduled to guarantee latency of packet. Earliest timeslot and latest timeslot of transmission $T_{i,k}^{\ j}$ when transmission can be scheduled that defined by $ET(T_{i,k}^{\ j})$ and $LST(T_{i,k}^{\ j})$ at timeslot S.

- $ET(T_{i,k}^{\ j}) = MAX(N_i - k + T_i * j, S)$
- $LST(T_{i,k}^{\ j}) = T_i * j - k - 1$

With transmission $T_{i,k}^{\ j}$ is k hop away to destination node (sink node) and N_i is hops away to destination node (sink node).

$ET(T_{i,k}{}^{j})$ of transmission $T_{i,k}{}^{j}$ is depended on current timeslot S, and it computed in each timeslot. $LST(T_{i,k}{}^{j})$ is not depended on current timeslot, it has fixed value.

For example with packet $p_{m,n}$ will be combined with packet $p_{i,j}$, transmission which transmits packet $p_{m,n}$ is $T_{ab}{}^{k}$ [x:y] and transmission which transmits packet $p_{i,j}$ is $T_{cd}{}^{z}$ [t:q] Combining packet $p_{i,j}$ and $p_{m,n}$ that's referenced by combined transmission $T_{ab}{}^{k}$ [x:y], $T_{cd}{}^{z}$ [t:q]:

$T_{ab}{}^{k}$ [x:y] + $T_{cd}{}^{z}$ [t:q] -> $TC_{ab}{}^{k}$ [MAX(x,t):MIN(y,q)] : adds transmission $TC_{ab}{}^{k}$ and removes $T_{ab}{}^{k}$, $T_{cd}{}^{z}$

Table 1: Time window value for all transmissions at timeslot 0

	$TW(T_{i,k}{}^{j},0)$		$TW(T_{i,k}{}^{j},0)$		$TW(T_{i,k}{}^{j},0)$		$TW(T_{i,k}{}^{j},0)$		$TW(T_{i,k}{}^{j},0)$
$T_{2,0}{}^{1}$	[0:7]	$T_{6,2}{}^{1}$	[0:13]	$T_{7,1}{}^{1}$	[0:6]	$T_{8,1}{}^{1}$	[0:14]	$T_{13,2}{}^{1}$	[0:13]
$T_{2,0}{}^{2}$	[8:15]	$T_{6,1}{}^{1}$	[1:14]	$T_{7,0}{}^{1}$	[1:7]	$T_{8,0}{}^{1}$	[1:15]	$T_{13,1}{}^{1}$	[1:14]
$T_{5,1}{}^{1}$	[0:6]	$T_{6,0}{}^{1}$	[2:15]	$T_{7,0}{}^{2}$	[8:14]	$T_{11,1}{}^{1}$	[0:14]	$T_{13,0}{}^{1}$	[2:15]
$T_{5,0}{}^{1}$	[1:7]	$T_{3,0}{}^{1}$	[0:7]	$T_{7,1}{}^{2}$	[9:15]	$T_{11,0}{}^{1}$	[1:15]	$T_{10,2}{}^{1}$	[0:13]
$T_{5,1}{}^{2}$	[8:14]	$T_{3,0}{}^{2}$	[8:15]	$T_{9,2}{}^{1}$	[0:13]	$T_{12,2}{}^{1}$	[0:13]	$T_{10,1}{}^{1}$	[1:14]
$T_{5,0}{}^{2}$	[9:15]	$T_{4,0}{}^{1}$	[0:7]	$T_{9,1}{}^{1}$	[1:14]	$T_{12,1}{}^{1}$	[1:14]	$T_{10,0}{}^{1}$	[2:15]
		$T_{4,0}{}^{2}$	[8:15]	$T_{9,0}{}^{1}$	[2:15]	$T_{12,0}{}^{1}$	[2:15]		

In table 2, we present transmissions of packets which is generated by each flow before grouping node. Flow F_5 in one slot frame will generate L_S/P_5 = 16/8 = 2 packet that are { $T_{5,1}{}^{1}$, $T_{5,0}{}^{1}$, $T_{5,1}{}^{2}$, $T_{5,0}{}^{2}$} . Time window [*x*: *y*] denotes that, the time window starts from time slot *x* and ends timeslot y in the current slotframe. With $T_{m,n}{}^{k}$[x:y], this transmission can be scheduled from timeslot x to y in one slotframe. After timeslot y, if $T_{m,n}{}^{k}$ is not scheduled that's mean this transmission misses deadline so with this case we get result that is unschedulable.

As shown in Fig. 7, in timeslot 0, we have

- Packet from node 5 which is referenced by $T_{5,1}{}^{1}$, $T_{5,0}{}^{1}$ and packet from node 6 which is referenced by $T_{6,1}{}^{1}$, $T_{6,0}{}^{1}$ from node 5 to sink node. At node 5, packet of node 5 and node 6 will be combined, this action is referenced by
  - ✓ $T_{5,1}{}^{1}$[0:6] + $T_{6,1}{}^{1}$[1:14] -> $TC_{6,1}{}^{1}$[1:6] we remove $T_{5,1}{}^{1}$, $T_{6,1}{}^{1}$and add $TC_{6,1}{}^{1}$ to transmission list.
  - ✓ $T_{5,0}{}^{1}$[1:7] + $T_{6,0}{}^{1}$[2:13] -> $TC_{6,0}{}^{1}$[2:7] we remove $T_{5,0}{}^{1}$, $T_{6,0}{}^{1}$and add $TC_{6,0}{}^{1}$ to transmission list.
- Combined packet from node 5, 6 which is referenced $TC_{6,0}{}^{1}$ and packet from node 2 which is referenced by $T_{2,0}{}^{1}$. At node 2, combined packet of node 5,6 and packet from node 2 will be combined, this action is referenced by
  - ✓ $TC_{6,0}{}^{1}$[2:7] + $T_{2,0}{}^{1}$[0:7] -> $TC_{6,0}{}^{1}$[2:7] we remove $T_{2,0}{}^{1}$[0:7] from transmission list

- Packet from node 9 which is referenced by $T_{9,1}^2$, $T_{9,0}^1$ and packet from node 7 which is referenced by $T_{7,1}^1$, $T_{7,0}^1$ from node 7 to sink node. At node 7, packet of node 9 and node 7 will be combined, this action is referenced by
  - ✓ $T_{9,1}^1[1:14] + T_{7,1}^1[0:6]$ -> $TC_{9,1}^1[1:6]$ we remove $T_{7,1}^1$, $T_{9,1}^1$and add $TC_{9,1}^1$ to transmission list.
  - ✓ $T_{9,0}^1$ $[2:15] + T_{7,0}^1[1:7]$ -> $TC_{9,0}^1[2:7]$ we remove $T_{9,0}^1$, $T_{7,0}^1$and add $TC_{9,0}^1$ to transmission list.
- Packet from node10 which is referenced by $T_{10,1}^1$, $T_{10,0}^1$ and packet from node 8 which is referenced by $T_{8,1}^1$, $T_{8,0}^1$ from node 8 to sink node. At node 7, packet of node 9 and node 7 will be combined, this action is referenced by
  - ✓ $T_{10,1}^1[1:14] + T_{8,1}^1[0:14]$ -> $TC_{10,1}^1[1:14]$ we remove $T_{10,1}^1$, $T_{8,1}^1$and add $TC_{10,1}^1$ to transmission list.
  - ✓ $T_{10,0}^1[2:15] + T_{8,0}^1[1:15]$ -> $TC_{10,0}^1[2:15]$ we remove $T_{10,0}^1$, $T_{8,0}^1$ and add $TC_{10,0}^1$ to transmission list.
- Packet from node 11 which is referenced by $T_{11,1}^1$, $T_{11,0}^1$, packet from node 12 which is referenced by $T_{12,1}^1$, $T_{12,0}^1$ and packet from node 13 which is referenced by $T_{13,1}^1$, $T_{13,0}^1$ from node 8 to sink node. At node 7, packet of node 9 and node 7 will be combined, this action is referenced by
  - ✓ $T_{13,1}^1[1:14] + T_{12,1}^1[1:14] + T_{11,1}^1[0:14]$ -> $TC_{13,1}^1[1:14]$ we remove $T_{13,1}^1$, $T_{12,1}^1$, $T_{11,1}^1$and add $TC_{13,1}^1$ to transmission list.
  - ✓ $T_{13,0}^1[2:15] + T_{12,0}^1[2:15] + T_{11,1}^1[1:15]$ -> $TC_{13,0}^1[2:15]$ we remove $T_{13,0}^1$, $T_{12,0}^1$, $T_{11,0}^1$and add $TC_{13,0}^1$ to transmission list.
- Combined packet from node 13, 12, 11 which is referenced $TC_{13,0}^1$ and packet from node 2 which is referenced by $T_{2,0}^1$. At node 2, combined packet of node 5,6 and packet from node 2 will be combined, this action is referenced by
  - ✓ $TC_{13,0}^1[2:15] + T_{4,0}^1[0:7]$ -> $TC_{13,0}^1[2:7]$ we remove $T_{4,0}^1[0:7]$ from transmission list

Table 2: Time window value for all transmissions at timeslot 0 after grouping

	$TW(T_{i,k}^j,0)$		$TW(T_{i,k}^j,0)$		$TW(T_{i,k}^j,0)$		$TW(T_{i,k}^j,0)$		$TW(T_{i,k}^j,0)$
$T_{2,0}^2$	[8:15]	$T_{6,2}^1$	[0:13]	$T_{3,0}^1$	[0:7]	$T_{10,2}^1$	[0:13]	$T_{9,2}^1$	[0:13]
$T_{5,1}^2$	[8:14]	$TC_{6,1}^1$	[1:6]	$T_{3,0}^2$	[8:15]	$TC_{10,1}^1$	[1:14]	$TC_{9,1}^1$	[1:6]
$T_{5,0}^2$	[9:15]	$TC_{6,0}^1$	[2:7]	$T_{13,2}^1$	[0:13]	$TC_{10,0}^1$	[2:15]	$TC_{9,0}^1$	[2:7]
$T_{4,0}^2$	[8:15]	$T_{7,0}^2$	[8:14]	$TC_{13,1}^1$	[1:14]				
$T_{12,2}^1$	[0:13]	$T_{7,1}^2$	[9:15]	$TC_{13,0}^1$	[2:15]				

The second parameter that accounts for prioritizing $T_{m,n}^k$ in the current time slot *S* is the average collision posed by the collisions that each remaining transmissions (i.e., $T_{m,n}^k$, $T_{m,n-1}^k$,… , $T_{m,0}^k$) might deal with in time slot S. To do so, we consider number of duplex conflicting collisions. A duplex conflicting collision is imposed by a transmission $T_{m,n}^k$ existed on a conflicting link in CNF($v_p v_q$) where its life time(TW($T_{m,n}^l$, S) overlaps with TW($T_{i,j}^k$, S)). We defined $N_{cnf}(T_{m,n}^k,T)$ that is number of duplex conflict transmissions with transmission $T_{m,n}^k$ in timeslot T and $N_{cnf}^{avg}(T_{m,n}^k,T)$ that is average number of duplex conflict transmission with $T_{m,n}^k$ in timeslot T

$$N_{cnf}^{avg}(T_{m,n}^k,T) = \sum_{i<=k} N_{cnf}(T_{m,n}^k,T)/(k+1)$$

The priority of transmission $T_{m,n}^k$ or $TC_{m,n}^k$ in the timeslot T is defined by

$$PR(T_{m,n}^k, T) = |TW(T_{m,n}^k,T)| - N_{cnf}^{avg}(T_{m,n}^k,T) \quad [10]$$

$$PR(TC_{m,n}^k, T) = |TW(TC_{m,n}^k,T)| - N_{cnf}^{avg}(TC_{m,n}^k,T)$$

Where $|TW(T_{m,n}^k,T)|$ or $|TW(TC_{m,n}^k,T)|$ is number available of timeslot which transmission $T_{m,n}^k$ or $TC_{m,n}^k$ can be scheduled in time slot T. For example, in Table 3 we present $N_{cnf}^{avg}(T_{m,n}^k,T)$, PR($T_{m,n}^k$, T) for all of transmission after grouping in timeslot 0, in the network given in Fig. 4.

Table 3: $N_{cnf}^{avg}(T_{m,n}^k,T)$, PR($T_{m,n}^k$, T) for each transmission after grouping in timeslot 0

	TW($T_{m.n}^k$,0)	$N_{cnf}^{avg}(T_{m,n}^k,0)$	PR($TC_{m,n}^k$, T)		TW($T_{m.n}^k$,0)	$N_{cnf}^{avg}(T_{m,n}^k,0)$	PR($TC_{m,n}^k$, T)
$T_{12,2}^1$	[0:13]	3.33	9.67	$T_{3,0}^1$	[0:7]	6	1
$T_{9,2}^1$	[0:13]	2.33	11.67	$T_{6,2}^1$	[0:13]	2.67	11.23
$T_{10,2}^1$	[0:13]	5	8	$T_{13,2}^1$	[0:13]	3.33	9.67

3.5 PC-PCLLF: Packet Combining With Path Conflict Aware Least Laxity First Algorithm

Our proposed PC-PCLLF scheduling algorithm is defined as follows. The input parameters of the PC-PCLLF algorithm are G, F, D, T, U, n_{ch}, S, and the output is a feasible scheduling solution:

Algorithm 1: PCLLF (G, D, T, U, n_{ch}, S)

1: Define the current time slot S=0, and define Pr($T_{m,n}^k$,0) for each $T_{m,n}^k$

2: While (not all the transmissions are scheduled) do

2.1: Group node, combine packet, and update transmission list

2.2: Calculate the set Released(S)

2.3: Calculate Pr($T_{m,n}^k$, S) for all transmissions

2.4: If there is a transmission that is not schedulable or miss dealine

2.4.1 Return “unschedulable”, Exit.

2.5: Else

2.5.1 Initiate the channel counter, m=0

2.5.2 While ($m \leq n_{ch}$) do

2.5.2.1 Select a released transmission with the highest priority, T*
When several transmissions have the same priority, assign a higher priority to a transmission with the smallest length for its time window.

2.5.2.2 Assign T* to current slot S on current channel m

2.5.2.3 Remove T* and all transmissions conflicting with T* from Released(S)

2.5.2.4 Go to the next channel offset, m=m+1

2.5.3 End while

2.5.4 Go to next slot S = S+1

2.6 End if

3: End While

4: Return the Schedule

In the first step, algorithm defines transmission list for all of packet. In line 2.1, each timeslot S, we detect node and which packet can be combined as explained above to combine transmission and consider in node’s queue which is not is in any group in grouping step. If this node’s queue there are two packet, we combine this two packet if total data size is less than MTU. Then we update transmission list. In each node queue .after that, we calculate Released(S) based on transmission list. We calculate priority value of all transmission as described above. Our algorithm order all released transmission based on priority value as explained in above section then we choose highest priority. Otherwise, our algorithm continues to schedule the highest priority transmissions in the current time slot in line 2.5.2. We repeat this progress until all transmission are scheduled or there is a transmission which misses deadline.

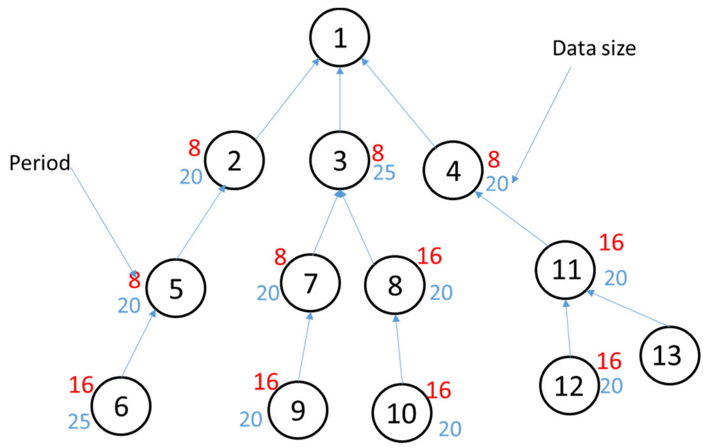

Figure 10: Network topology example

With example in Fig.8, We use 4 channel offset.

At time slot 0, we group node and combine packet as shown in section. Then, we have Release(0) is { $T_{6,2}^1$, $T_{10,2}^1$, $T_{9,2}^1$, $T_{12,2}^1$, $T_{13,2}^1$, $T_{3,0}^1$}.

Table 4: Priority of released transmission in timeslot 0

	$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$		$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$
$T_{12,2}^1$	[0:13]	3.33	9.67	$T_{3,0}^1$	[0:7]	6	1
$T_{9,2}^1$	[0:13]	2.33	11.67	$T_{6,2}^1$	[0:13]	2.67	11.23
$T_{10,2}^1$	[0:13]	5	8	$T_{13,2}^1$	[0:13]	3.33	9.67

Highest priority is lowest value. Based on priority of this transmissions, lowest value is $T_{3,0}^1$. This transmission is scheduled for timeslot 0 and channel 0. Then with channel 1, we remove transmission $T_{3,0}^1$ and conflicting transmission with transmission $T_{3,0}^1$, we have list transmission which is ordered by priority value is { $T_{10,2}^1$, $T_{13,2}^1$, $T_{12,2}^1$, $T_{6,2}^1$, $T_{9,2}^1$} and we choose highest priority is $T_{10,2}^1$ which is scheduled for timeslot 0 and channel 1. Then with channel 2, we remove transmission $T_{10,2}^1$ and conflicting transmission with transmission $T_{10,2}^1$, we have list of transmissions which is ordered by priority value is {$T_{13,2}^1$, $T_{12,2}^1$, $T_{6,2}^1$, $T_{9,2}^1$} and we choose highest priority is $T_{13,2}^1$ which is scheduled for timeslot 0 and channel 2. Then with channel 3, we remove transmission $T_{13,2}^1$ and conflicting transmission with transmission $T_{30,2}^1$, we have list of transmissions which is ordered by priority value is {$T_{6,2}^1$, $T_{9,2}^1$} and we choose highest priority is $T_{6,2}^1$ which is scheduled for timeslot 0 and channel 3. 4 channel is scheduled then it go to timeslot 1.

Table 5: schedule table in timeslot 0

	0	1	2	3	4	5	6	7	8
0	$T_{3,0}^1$								
1	$T_{10,2}^1$								
2	$T_{13,2}^1$								
3	$T_{6,2}^1$								

With $T_{3,0}^1$, node 3 transmit packet which has 25 bytes to node 1. With $T_{10,2}^1$, node 10 transmit packet which has 20 bytes to node 8. At node 8 we have 2 packet that are from node 8 and node 10, we combine two packets so we update data size of this transmission so we have $TC_{10,1}^1(40)$ with 40 is data size that is contained by this packet. With $T_{13,2}^1$, node 13 transmit packet which has 20 bytes to node 11. At node 11, we have 2 packets that are from node 11 and node 13, we combine two packets so we update data size of transmission so we have $TC_{13,1}^1(35)$. With $T_{6,2}^1$, node 6 transmit packet which has 25 bytes to node 5. At node 5, we have 2 packets from node 6, 5 and combine two packets so we update data size of transmission which is referenced to transmit this combined packet $TC_{6,1}^1(45)$

In timeslot 1, we update ECT and LT value for all of nodes and group node again. With subtree which has root node is node 3, we have total data size is 80 bytes, and ECT(3) = 3 and LT(3) = 7 so we can combine all packet from node 9,7,10,8 at node 3. This action is referenced by

- Combined packet from node 9, 7 which is referenced $TC_{9,0}^1$ and combined packet from node 10, 8 which is referenced by $T_{10,0}^1$
  - ✓ $T_{9,0}^1[2:7] + T_{10,0}^1[2:15] \rightarrow TC_{13,1}^1[2:7]$

In timeslot 1, we have Released(1) is $\{T_{9,2}^1, T_{10,1}^1, T_{6,1}^1, T_{13,2}^1\}$

Table 6: Priority of released transmission in timeslot 1

	$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$		$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$
$TC_{10,1}^1$	[1:14]	1.5	11.5	$TC_{6,1}^1$	[1:6]	1	4
$T_{9,2}^1$	[1:13]	1	11	$T_{12,2}^1$	[1:13]	1	11

Highest priority is lowest value. Based on priority of this transmissions, lowest value is $TC_{6,1}^1$. This transmission is scheduled for timeslot 1 and channel 0. Then with channel 1, we remove transmission $TC_{6,1}^1$ and conflicting transmission with transmission $TC_{6,1}^1$, we have list transmission which is ordered by priority value is $\{T_{12,2}^1, T_{9,2}^1, TC_{10,1}^1,\}$ and we choose highest priority is $T_{12,2}^1$ which is scheduled for timeslot 1 and channel 1. Then with channel 2, we remove transmission $T_{12,2}^1$ and conflicting transmission with transmission $T_{12,2}^1$, we have list transmission which is ordered by priority value is

$\{T_{9,2}^1$, $TC_{10,1}^1\}$ and we choose highest priority is $T_{9,2}^1$ which is scheduled for timeslot 1 and channel 2. Then with channel 3, we remove transmission $T_{9,2}^1$ and conflicting transmission with transmission $T_{9,2}^1$, we have list transmission which is ordered by priority value is $\{ TC_{10,1}^1\}$ and we choose highest priority is $TC_{10,1}^1$ which is scheduled for timeslot 1 and channel 3. 4 channel is scheduled then go to timeslot 2.

Table 7: schedule table in timeslot 1

	0	1	2	3	4	5	6	7	8
0	$T_{3,0}^1$	$TC_{6,1}^1$							
1	$T_{10,2}^1$	$T_{12,2}^1$							
2	$T_{13,2}^1$	$T_{9,2}^1$							
3	$T_{6,2}^1$	$TC_{10,1}^1$							

With $TC_{6,1}^1$, node 5 transmit combined packet which has 45 bytes to node 2. At node 2 we have 2 packet that are from node 5 and node 2, we combine two packets so we update data size of this transmission so we have $TC_{6,0}^1(65)$ with 65 is data size that is contained by this packet. With $T_{12,2}^1$, node 12 transmits packet which has 20 bytes to node 11. At node 11, we have 2 packets that are packet from node 12 and combined packet node 11, we combine two packets so we update data size of transmission so we have $TC_{13,1}^1(55)$. With $T_{9,2}^1$, node 9 transmits packet which has 20 bytes to node 7. At node 7, we have 2 packets from node 9, 7 and combine two packets so we update data size of transmission which is referenced to transmit this combined packet $TC_{9,1}^1(45)$. With $TC_{10,1}^1$, node 8 transmits combined packet which has 40 bytes to node 3.

In timeslot 2, all of packet are already detected to combine so group node does not change. Then, we have Released(2) is $\{TC_{6,0}^1, TC_{9,1}^1, TC_{13,1}^1\}$

Table 8: Priority of released transmission in timeslot 2

	$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$		$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$
$TC_{6,0}^1$	[2:7]	2	3	$TC_{13,1}^1$	[2:14]	1	11
$TC_{9,1}^1$	[2:6]	1	3				

Highest priority is lowest value. Based on priority of this transmissions, lowest value is $TC_{6,1}^1$, $TC_{9,1}^1$. Higher priority of two transmission that is transmission which has lower LST (6) that is $TC_{9,1}^1$. This transmission is scheduled for timeslot 2 and channel 0. Then with channel 1, we remove transmission $TC_{6,0}^1$ and conflicting transmission with transmission $TC_{6,0}^1$, we have list transmission which is ordered by priority value is $\{TC_{9,1}^1$, $TC_{13,1}^1\}$ and we choose highest priority is $TC_{9,1}^1$ which is scheduled for timeslot 2 and channel 1. Then with channel 2, we remove transmission $T_{9,1}^1$ and conflicting transmission with transmission $T_{9,1}^1$, we have list transmission which is ordered by priority value is { $TC_{13,1}^1$} and we choose highest priority is $T_{13,1}^1$ which is scheduled for timeslot 2 and channel 2. Then we don't have any transmission on list after removing $TC_{13,1}^1$ so go to timeslot 3.

Table 9: scheduled table in timeslot 2

	0	1	2	3	4	5	6	7	8
0	$T_{3,0}^1$	$TC_{6,1}^1$	$TC_{9,1}^1$						
1	$T_{10,2}^1$	$T_{12,2}^1$	$TC_{6,0}^1$						
2	$T_{13,2}^1$	$T_{9,2}^1$	$TC_{13,1}^1$						
3	$T_{6,2}^1$	$TC_{10,1}^1$							

With $TC_{9,1}^1$, node 7 transmit combined packet which has 40 bytes to node 3. At node 3, there are two packet that are combined packet which has 40 bytes data from node 7, 9 and combined packet which has 40 bytes from node 8, 10, we combine this packets to packet which has 80 bytes so we update data size for transmission $TC_{9,0}^1$(80). With $TC_{6,0}^1$, node 2 transmit combined packet which contains 65 bytes to node 1. With $TC_{13,1}^1$, node 11 transmit combined packet which contains 55 bytes to node 4. At node 4 there are two packets that are packet which has 20 bytes data from node 4 and combined packet which is received from node 11 and contains 55 bytes so we combine two packets to packet which has 75 bytes data, update transmission $TC_{13,1}^1$(75) that's is transmission which is referenced node 4 transmit this packet to node 1.

In timeslot 3, all of packet are already detected to combine so group node does not change. Then, we have Released(3) is { $TC_{9,0}^1$, $TC_{13,0}^1$}.

Table 10: Priority of released transmission in timeslot 3

	$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$		$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$
$TC_{9,0}^1$	[3:7]	1	3	$TC_{13,0}^1$	[3:7]	1	3

Two transmission $TC_{9,0}^1$, $TC_{13,0}^1$ which have same priority value and deadline of transmission so we choose $TC_{9,0}^1$. This transmission is scheduled for timeslot 3 and channel 0. Then with channel 1, we remove transmission $TC_{9,0}^1$ and conflicting transmission with transmission $TC_{9,0}^1$, we have list transmission which is ordered by priority value is {}. Because we have empty list so go to timeslot 4.

Table 11: scheduled table in timeslot 3

	0	1	2	3	4	5	6	7	8
0	$T_{3,0}^1$	$TC_{6,1}^1$	$TC_{9,1}^1$	$TC_{9,0}^1$					
1	$T_{10,2}^1$	$T_{12,2}^1$	$TC_{6,0}^1$						
2	$T_{13,2}^1$	$T_{9,2}^1$	$TC_{13,1}^1$						
3	$T_{6,2}^1$	$TC_{10,1}^1$							

With transmission $TC_{9,0}^1$, node 3 transmits combined packet which contains 80 bytes to node 1.

In timeslot 4, all of packet are already detected to combine so group node does not change. Then, we have Released(4) is $\{TC_{13,0}^1\}$.

Table 12: Priority of released transmission in timeslot 4

	$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$		$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$
$TC_{13,0}^1$	[3:7]	1	3				

There is only transmission $TC_{13,0}^1$ in released set, we choose $TC_{13,0}^1$. This transmission is scheduled for timeslot 4 and channel 0. After removing transmission $TC_{13,0}^1$ from released set, we have empty list so go to timeslot 5

Table 13: scheduled table in timeslot 4

	0	1	2	3	4	5	6	7	8
0	$T_{3,0}^1$	$TC_{6,1}^1$	$TC_{9,1}^1$	$TC_{9,0}^1$	$TC_{13,0}^1$				
1	$T_{10,2}^1$	$T_{12,2}^1$	$TC_{6,0}^1$						
2	$T_{13,2}^1$	$T_{9,2}^1$	$TC_{13,1}^1$						
3	$T_{6,2}^1$	$TC_{10,1}^1$							

In timeslot 5, 6, 7 there is no transmission in released set.

In timeslot 8, node 8, 9, 3, 7, 4 generate packets. We group node as showed in Fig .9

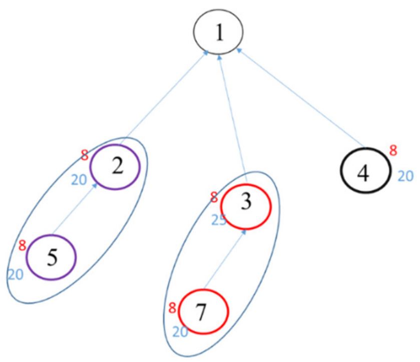

Figure 11: Group node results at timeslot 8

Table 14: Time window value for all transmissions at timeslot 8

	$TW(T_{i,k}^{j},8)$		$TW(T_{i,k}^{j},0)$		$TW(T_{i,k}^{j},0)$		$TW(T_{i,k}^{j},0)$		$TW(T_{i,k}^{j},0)$
$T_{5,1}^{2}$	[8:14]	$T_{7,1}^{2}$	[8:14]	$T_{2,0}^{2}$	[8:15]	$T_{3,0}^{2}$	[8:15]	$T_{4,0}^{2}$	[8:15]
$T_{5,0}^{2}$	[9:15]	$T_{7,0}^{2}$	[9:15]						

We consider node 2, total data size in subtree which has node 2 is root node is 40 bytes and (LT(2) = 15)- (ECT(2) = 9) = 6 > 1so we can combine packet node 5 and node 2 based on condition 1. This action is referenced by

- Combined packet from node 2, 5 which is referenced $T_{5,0}^{2}$ and packet from node 2 which is referenced by $T_{2,0}^{2}$
  - ✓ $T_{5,0}^{2}$[9:15]+ $T_{2,0}^{2}$[8:15] -> $TC_{5,0}^{2}$[9:15]

With same node 3 total data size subtree which has node 3 is root node is 45 bytes and (LT(2) = 15)-(ECT(2) = 9) = 6 > 1so we can combine packets from node 7 and node 3. This action is referenced by

- Combined packet from node 7, 3 which is referenced $T_{7,0}^{2}$ and packet from node 2 which is referenced by $T_{3,0}^{2}$
  - ✓ $T_{7,0}^{2}$[9:15]+ $T_{3,0}^{2}$[8:15] -> $TC_{7,0}^{2}$[9:15]

Table 15: Time window value for all transmissions at timeslot 8 after grouping

	$TW(T_{i,k}^{j},8)$		$TW(T_{i,k}^{j},0)$		$TW(T_{i,k}^{j},0)$		$TW(T_{i,k}^{j},0)$		$TW(T_{i,k}^{j},0)$
$T_{5,1}^{2}$	[8:14]	$TC_{5,0}^{2}$	[9:15]	$T_{7,1}^{2}$	[8:14]	$TC_{7,0}^{2}$	[9:15]	$T_{4,0}^{2}$	[8:15]

In timeslot 8, we have Released(8) is {$T_{5,1}^{2}$, $T_{7,1}^{2}$, $T_{4,0}^{2}$ }.

Table 16: Priority of released transmission in timeslot 8

	$TW(T_{m.n}^{k},0)$	$N_{cnf}^{avg}(T_{m,n}^{k},0)$	$PR(T_{m,n}^{k}, T)$		$TW(T_{m.n}^{k},0)$	$N_{cnf}^{avg}(T_{m,n}^{k},0)$	$PR(T_{m,n}^{k}, T)$
$T_{5,1}^{2}$	[8:14]	1	5	$T_{7,1}^{2}$	[8:14]	1	5
$T_{4,0}^{2}$	[8:15]	2	5				

Transmissions $T_{5,1}^{2}$ and $T_{7,1}^{2}$ have same lowest priority value and same lowest deadline so we choose $T_{5,1}^{2}$ which is scheduled for timeslot 8 and channel 0. Then with channel 1, we remove transmission $T_{5,1}^{2}$ and conflicting transmission $T_{5,1}^{2}$ from released set, we have list transmission which is ordered by priority value is {$T_{7,1}^{2}$, $T_{4,0}^{2}$} and we choose highest priority is $T_{7,1}^{2}$ which is scheduled for timeslot 8 and channel 1. Then with channel 2, we remove transmission $T_{7,1}^{2}$ and conflicting transmission with $T_{7,1}^{2}$ from released set, we have list transmission which is ordered by priority value is {$T_{4,0}^{2}$} and we choose highest priority is $T_{4,0}^2$. Then with channel 4, we remove transmission $T_{4,0}^2$ and conflicting transmission $T_{4,0}^2$ from released set, we have empty so go to timeslot 9.

Table 17: scheduled table in timeslot 8

	0	1	2	3	4	8	9	10	11
0	$T_{3,0}^1$	$TC_{6,1}^1$	$TC_{9,1}^1$	$TC_{9,0}^1$	$TC_{13,0}^1$	$T_{5,1}^2$			
1	$T_{10,2}^1$	$T_{12,2}^1$	$TC_{6,0}^1$			$T_{7,1}^2$			
2	$T_{13,2}^1$	$T_{9,2}^1$	$TC_{13,1}^1$			$T_{4,0}^2$			
3	$T_{6,2}^1$	$TC_{10,1}^1$							

With $T_{5,1}^1$, node 5 transmit packet which has 20 bytes to node 2. At node 2, there are two packet that are combined packet which has 20 bytes data from node 5 and packet which received from node 5 we combine this packets to packet which has 40 bytes so we update data size for transmission $TC_{5,0}^1(40)$. With $T_{7,1}^1$, node 7 transmit combined packet which has 20 bytes to node 3. At node 3, there are two packet that are combined packet which has 20 bytes data from node 7,and packet which has 40 bytes from node 3, we combine this packets to packet which has 40 bytes so we update data size for transmission $TC_{7,0}^1(40)$. With $T_{4,0}^2$, node 4 transmits 20 bytes to node 1.

In timeslot 9, we have Released(9) is $\{TC_{5,0}^2, TC_{7,0}^2\}$.

Table 18: Priority of released transmission in timeslot 9

	$TW(T_{m.n}^k,0)$[1]	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$		$TW(T_{m.n}^k,0)$	$N_{cnf}^{avg}(T_{m,n}^k,0)$	$PR(T_{m,n}^k, T)$
$TC_{5,0}^2$	[9:15]	1	5	$TC_{7,0}^2$	[9:15]	1	5

Transmissions $TC_{5,0}^2$ and $TC_{7,0}^2$ have same lowest priority value and same lowest deadline so we choose $T_{5,0}^2$ which is scheduled for timeslot 9 and channel 0. Then with channel 1, we remove transmission $TC_{5,0}^2$ and conflicting transmission with $TC_{5,0}^2$ from released set, we have empty so go to timeslot 10.

Table 19: scheduled table in timeslot 9

	0	1	2	3	4	8	9	10	11
0	$T_{3,0}^1$	$TC_{6,1}^1$	$TC_{9,1}^1$	$TC_{9,0}^1$	$TC_{13,0}^1$	$T_{5,1}^2$	$TC_{5,0}^2$		
1	$T_{10,2}^1$	$T_{12,2}^1$	$TC_{6,0}^1$			$T_{7,1}^2$			
2	$T_{13,2}^1$	$T_{9,2}^1$	$TC_{13,1}^1$			$T_{4,0}^2$			
3	$T_{6,2}^1$	$TC_{10,1}^1$							

With $TC_{5,0}^2$, node 2 transmits combined packet which has 40 bytes to node 1.

---

[1]

In timeslot 10, we have Released(10) is $\{TC_{7,0}^2\}$. There is only $TC_{7,0}^2$ in set so this transmission is scheduled for timeslot 10 and channel 0.

Finally, with example in Fig. 8, after applying our proposal scheduling, we have scheduled table

Table 20: Final scheduled table

	0	1	2	3	4	8	9	10	11
0	$T_{3,0}^1$	$TC_{6,1}^1$	$TC_{9,1}^1$	$TC_{9,0}^1$	$TC_{13,0}^1$	$T_{5,1}^2$	$TC_{5,0}^2$	$TC_{7,0}^2$	
1	$T_{10,2}^1$	$T_{12,2}^1$	$TC_{6,0}^1$			$T_{7,1}^2$			
2	$T_{13,2}^1$	$T_{9,2}^1$	$TC_{13,1}^1$			$T_{4,0}^2$			
3	$T_{6,2}^1$	$TC_{10,1}^1$							

In final schedule table:

- With transmission $T_{6,2}^1$, $TC_{6,1}^1$,$TC_{6,0}^1$
  - ✓ $T_{6,2}^1$ is scheduled for timeslot 0 and channel 3. That is mean: in timeslot 0, node 6 transmits packet to node 5 at channel 3, after receiving this packet, node 5 combine this packet with its own packet.
  - ✓ $TC_{6,1}^1$ is scheduled for timeslot 1 and channel 0. That is mean: in timeslot 1, node 5 transmits combined packet to node 2 at channel 0, after receiving this packet, node 2 combine this packet with its own packet.
  - ✓ $TC_{6,0}^1$ is scheduled for timeslot 2 and channel 1. That is mean: in timeslot 2, node 2 transmits combined packet to node 1, node 1 receives this packet so node 1 receives data from node 2, 6, 5 which is in this combined packet.
- With transmission $T_{13,2}^1$, $T_{12,2}^1$, $TC_{13,1}^1$, $TC_{13,0}^1$
  - ✓ $T_{13,2}^1$ is scheduled for timeslot 0 and channel 2. That is mean in timeslot 0, node 13 transmits packet to node 11 at channel 2. After receiving this packet, node 11 combines this packet with its own packet.
  - ✓ $T_{12,2}^1$ is scheduled for timeslot 1 and channel 1. That is mean: in timeslot 1, node 12 transmits combined packet to node 11 at channel 1. After receiving this packet, node 11 combine this packet with combined packet in node 11.
  - ✓ $TC_{13,1}^1$ is scheduled for timeslot 2 and channel 2. That is mean: in timeslot 2, node 11 transmits combined packet to node 4 at channel 2. After receiving this packet, node 4 combines this packet with its own packet.
  - ✓ $TC_{13,0}^1$ is scheduled for timeslot 4 and channel 0. That is mean: in timeslot 4, node 11 transmits combined packet to node 1 at channel 0. Node 1 receives this packet so node 1 receives data from node 13, 12, 11, 4.

- With transmission $T_{10,2}^1$, $T_{9,2}^1$, $TC_{10,1}^1$, $TC_{9,1}^1$, $TC_{9,0}^1$
  - ✓ $T_{10,2}^1$ is scheduled for timeslot 0 and channel 1. That is mean in timeslot 0, node 10 transmits packet to node 8 at channel 1. After receiving this packet, node 8 combines this packet with its own packet.
  - ✓ $T_{9,2}^1$ is scheduled for timeslot 1 and channel 1. That is mean in timeslot 1, node 9 transmits packet to node 7 at channel 1. After receiving this packet, node 7 combines this packet with its own packet.
  - ✓ $TC_{10,1}^1$ is scheduled for timeslot 1 and channel 2. That is mean in timeslot 1, node 8 transmit combined packet to node 3 at channel 2.
  - ✓ $TC_{9,1}^1$ is scheduled for timeslot 2 and channel 0. That is mean in timeslot 2, node 7 transmit combined packet to node 3 at channel 0. After receiving this packet, node 3 combines this packet with combined packet in queue.
  - ✓ $TC_{9,0}^1$ is scheduled for timeslot 3 and channel 0. That is mean: in timeslot 3, node 3 transmits combined packet to node 1 at channel 0. Node 1 receives this packet so node 1 receives data from node 9, 7, 10, 8
- With transmission $T_{3,0}^1$
  - ✓ $T_{3,0}^1$ is scheduled for timeslot 0 and channel 0. That's mean in timeslot 0 node 3 transmits packet to node 1 at channel 0.
- With transmission $T_{5,1}^2$, $TC_{5,0}^2$
  - ✓ $T_{5,1}^2$ is scheduled for timeslot 8 and channel 0. That is mean in timeslot 8, node 5 transmits packet to node 2 at channel 0. After receiving this packet, node 2 combines this packet with its own packet.
  - ✓ $TC_{5,0}^2$ is scheduled for timeslot 9 and channel 0. That is mean: in timeslot 9, node 2 transmits combined packet to node 1 at channel 0. Node 1 receives this packet so node 1 receives data from node 5, 2.
- With transmission $T_{7,1}^2$, $TC_{7,0}^2$
  - ✓ $T_{7,1}^2$ is scheduled for timeslot 8 and channel 1. That is mean in timeslot 8, node 7 transmits packet to node 3 at channel 0. After receiving this packet, node 3 combines this packet with its own packet.
  - ✓ $TC_{7,0}^2$ is scheduled for timeslot 10 and channel 0. That is mean: in timeslot 10, node 3 transmits combined packet to node 1 at channel 0. Node 1 receives this packet so node 1 receives data from node 7, 3.
- With transmission $T_{4,0}^2$
  - ✓ $T_{4,0}^2$ is scheduled for timeslot 8 and channel 0. That's mean in timeslot 8 node 4 transmits packet to node 1 at channel 0.

Chapter 4: PERFORMANCE EVALUATION

In this section, we evaluate the performance of PC-PCLLF, mainly in terms of schedualbility ratio, End-to-End delay, number of transmissions. It is the number of experiments in which an approach successfully schedules. We compare the performance of PC-PCLLF with the heuristic algorithms which is called Collision free-LLF (CLLF) and P-CLLF.

As the first case, we conduct experiments to evaluate the impact of network scale and period of flows on the performance of PC-PCLLF, CLLF and PCLLF. The experiments are arranged using case studies with loose, and tight periods. In the experiments with loose periods, we evaluate the performance of algorithms on case studies with periods randomly selected among the range [2^5-2^8] time slots. In the tight experiments, we exploit case studies with periods randomly selected among the range [2^4-2^6]. Data length which each node has to transmit to sink node we random value in this set {15, 20, 25, 30} (byte).

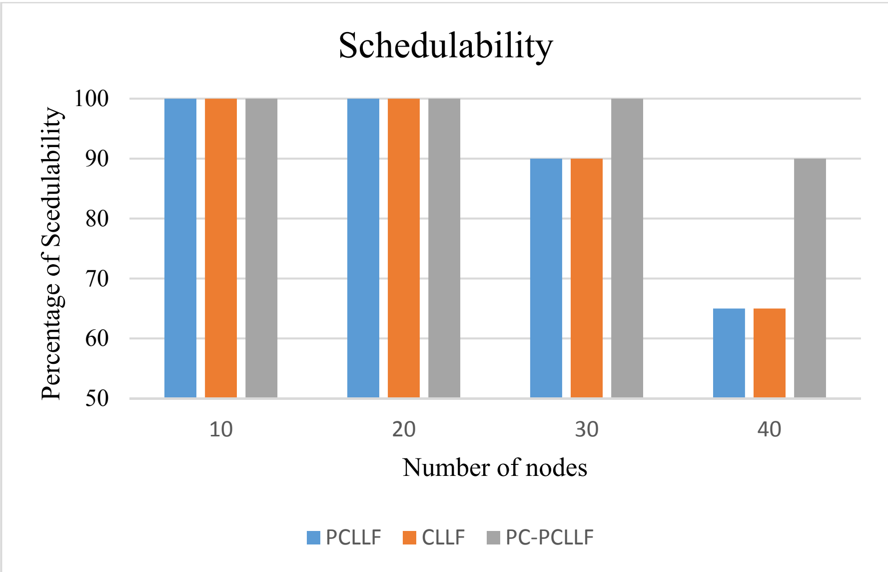

Figure 12: Schedulability with loose period

Fig. 12 shows the schedulability ratio per number of field devices in the network with loose period. The horizontal axis marks the number of nodes in the network, which is varied among the range [10, 20, 30, 40]. That is, the number of nodes on the horizontal axis denotes the scale of the problem defined for experiment. We observe that, PC-PLLF outperforms CLLF and PCLLF in all experiments, and the performance grows with increasing level of difficulty and the scale of networks. The main reason is that in each timeslot PC-PLLF consider which packets can be combined this based on condition, it reduces of number transmission so it increase schedualbilty. With our results. Our approach can apply to application which each node has period in range [2^5-2^8] timeslot with number node to 40 nodes such as monitoring in industrial, agriculture.

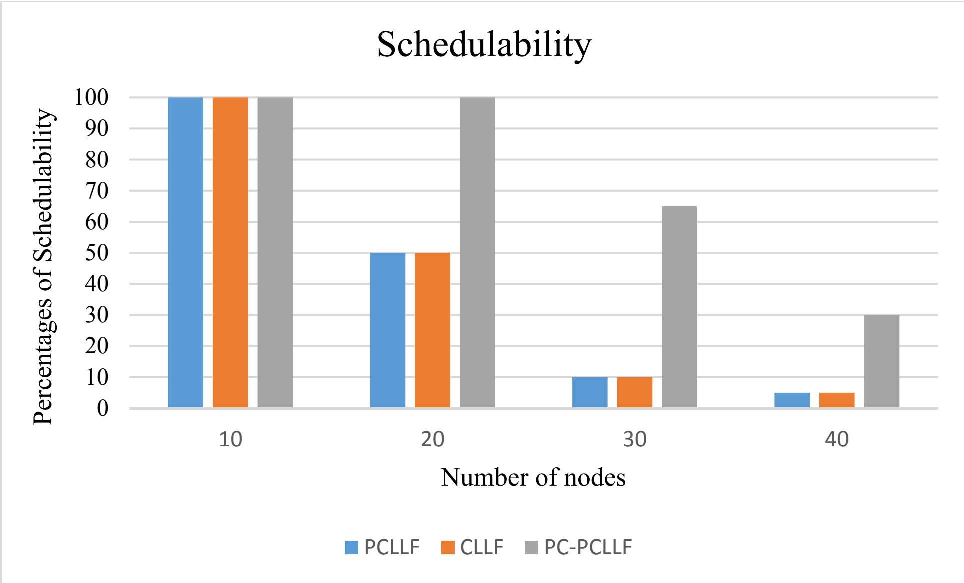

Figure 13: Schedulability with tight period

With tight period, when we increase number of nodes to 30, 40, schedulabiply of PCLLF and CLLF is very low. With our approach, the result is much higher than this approach. The reason our approach use packet combine, that optimizes number of transmissions so it increases schedulability. With same loose period, PC-PLLF outperforms CLLF and PCLLF in all experiments. With our results. Our approach can apply to application which each node has period in range [2^5-2^8] timeslot with number node to 20 nodes. It can be seen that, as the scale of networks increases or periods become tighter, the performance of all the approaches reduces.

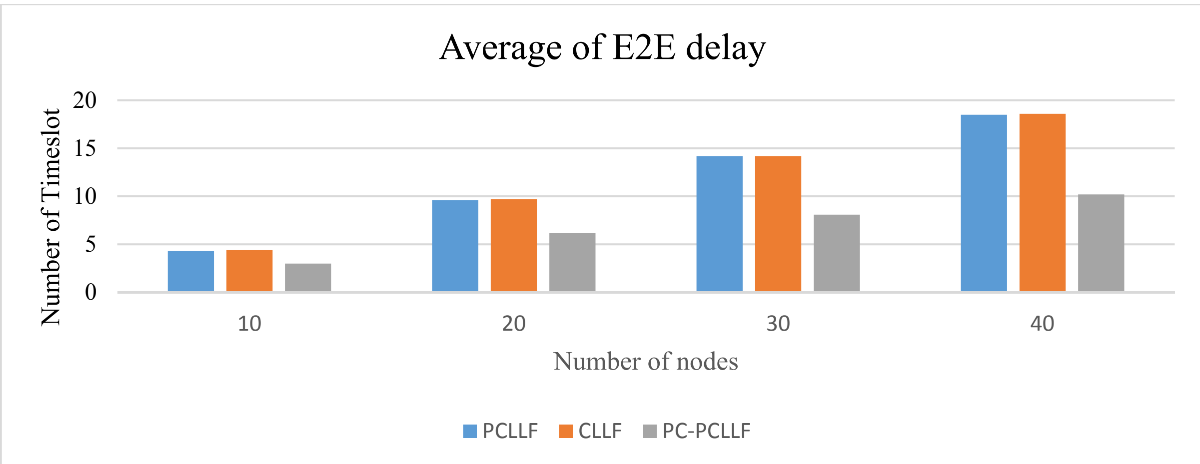

Figure 14: Average of E2E delay with loose period

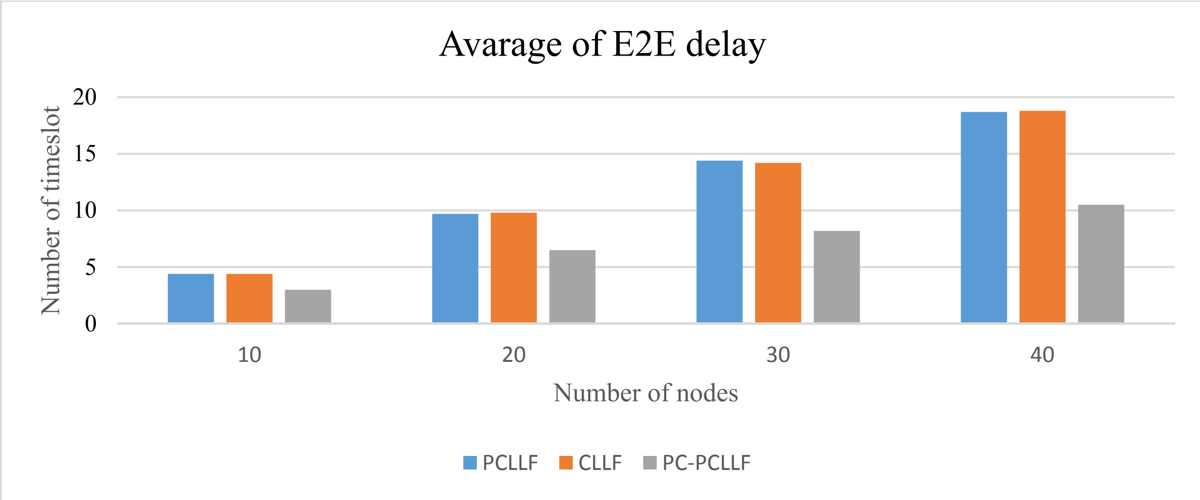

Figure 15: Average of E2E delay with tight period

Fig.14 and Fig.15 shows average of E2E delay with loose and tight period, PC-PLLF outperforms CLLF and PCLLF in all experiments. The results of PCLLF and CLLF are almost same.. The main reason, our approach optimize packet size so it optimize of number of transmission, and it reduces of E2E delay. When we increase of number of nodes, E2E delay increases because number of transmissions increase as shown in Fig 16, 17.

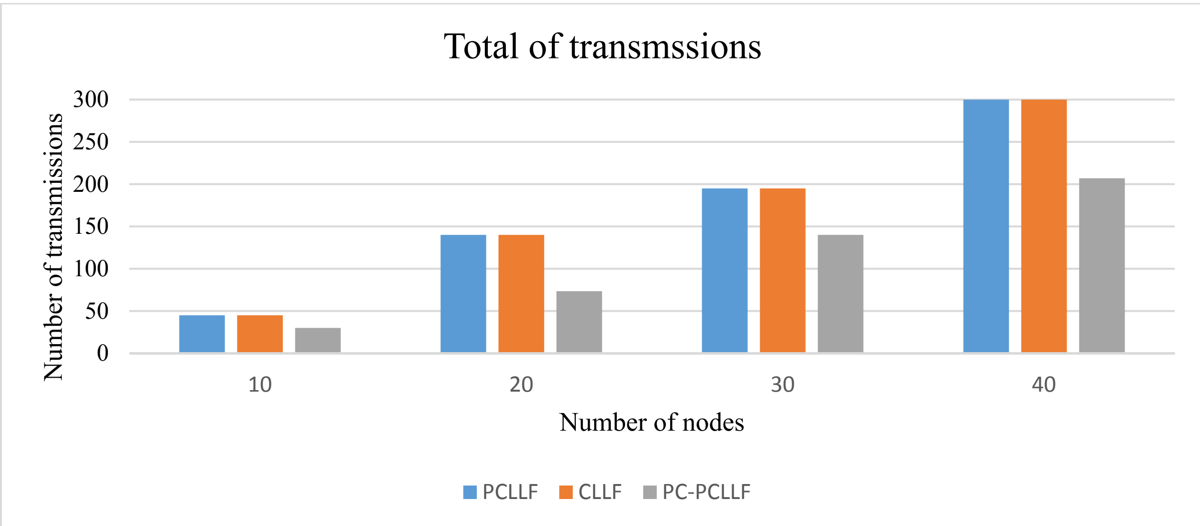

Figure 16: Total of transmission with loose period

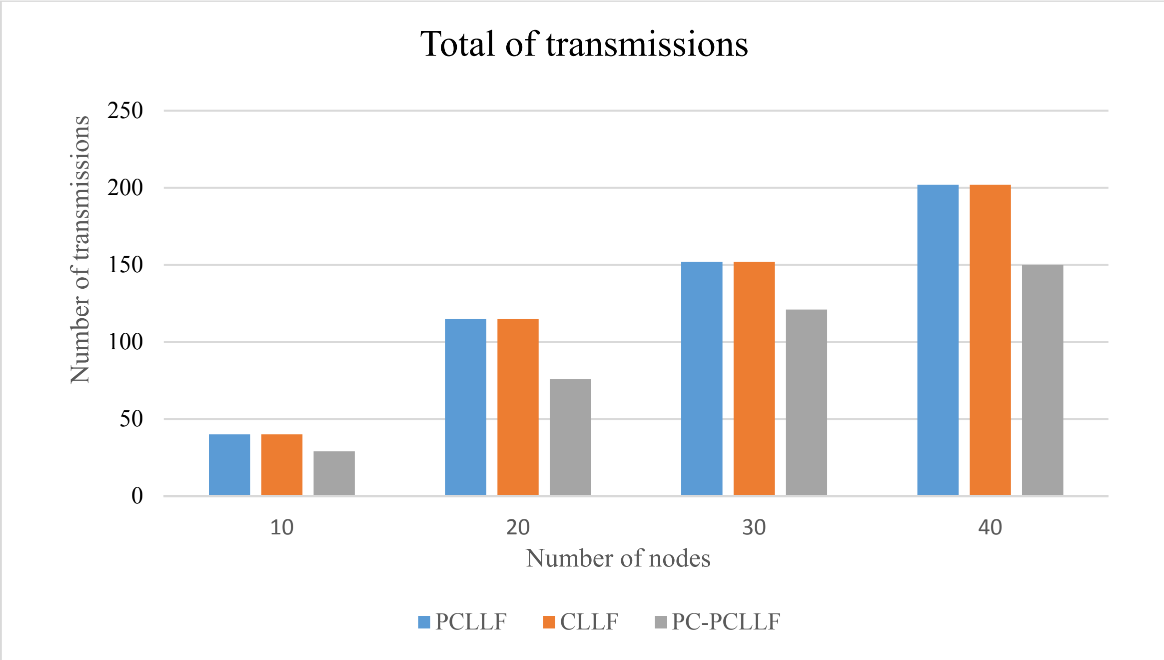

Figure 17: Total of transmission with tight period

Fig.16 and Fig.17 shows average of number of transmissions with loose and tight period, PC-PLLF outperforms CLLF and PCLLF in all experiments. Because our approach optimize packet length so it optimize number of transmission. When compare with PCLLF and CLLF, the number transmission reduces up to 34% with 20 nodes and tight period Because of this, our approach reduces energy consumption.

Chapter 5: CONCLUSION

In this study, we proposed heuristic scheduling algorithm that's based on data-aggregation and prioritizes each packet transmission dynamically based on its laxity (i.e., the remaining time before the end-to-end deadline) with different period apply to TSCH network to increase schedulability, reduce number of data transmissions, E2E delay, high reliability.

- We proposed data-aggregation mechanism to reduce number of packet transmissions. This mechanism is based on period of node and total data size in the path. Because, in TSCH network, maximum of packet size is 127 bytes and MAC header size is 21 bytes so maximum of payload size is 106 bytes. Period of all node can be different which is POW of two, so we choose nodes in the path which period of parent node is not less than child node, packet from parent will be combined with other node to increase schedulability.

According to the evaluation results, the proposed mechanisms shows highly schedulability and reduces number of transmissions that's mean reduce data traffic when comparing with some approaches P-CLLF, CLLF. Therefore, we conclude that our proposal is highly suitable for monitoring and control systems in industrial wireless sensor networks especially with low latency in real-time application.

REFERENCES

[1] W. J. Fleming, “New Automotive Sensors—A Review,” *IEEE Sens. J.*, vol. 8, no. 11, pp. 1900–1921, Nov. 2008.

[2] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, “Intra-Vehicle Networks: A Review,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 534–545, Apr. 2015.

[3] M. Srbinovska, C. Gavrovski, V. Dimcev, A. Krkoleva, and V. Borozan, “Environmental parameters monitoring in precision agriculture using wireless sensor networks,” *J. Clean. Prod.*, vol. 88, pp. 297–307, 2015.

[4] M. Pule, A. Yahya, and J. Chuma, “Wireless sensor networks: A survey on monitoring water quality,” *J. Appl. Res. Technol.*, vol. 15, no. 6, pp. 562–570, 2017.

[5] J. Yang, J. Zhou, Z. Lv, W. Wei, and H. Song, “A real-time monitoring system of industry carbon monoxide based on wireless sensor networks,” *Sensors*, vol. 15, no. 11, pp. 29535–29546, 2015.

[6] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry, “Distributed control applications within sensor networks,” *Proc. IEEE*, vol. 91, no. 8, pp. 1235–1246, Aug. 2003.

[7] A. Lavric, “LoRa (Long-Range) High-Density Sensors for Internet of Things,” *J. Sensors*, vol. 2019, p. 3502987, 2019.

[8] P. Thubert, T. Watteyne, R. Struik, and M. Richardson, “An Architecture for IPv6 over the TSCH mode of IEEE 802.15. 4. draft-ietf-6tisch-architecture-10,” *IETF Draft. March*, 2015.

[9] P. Thubert, T. Watteyne, M. R. Palattella, X. Vilajosana, and Q. Wang, “Ietf 6tsch: Combining ipv6 connectivity with industrial performance,” in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013, pp. 541–546.

[10] A. Darbandi and M. K. Kim, “Path collision-aware real-time link scheduling for TSCH wireless networks,” *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 9, pp. 4429–4445, 2019.

[11] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, “Real-time scheduling for WirelessHART networks,” in *2010 31st IEEE Real-Time Systems Symposium*, 2010, pp. 150–159.

[12] Y. Gao, X. Li, J. Li, and Y. Gao, “Distributed and Efficient Minimum-Latency Data Aggregation Scheduling for Multichannel Wireless Sensor Networks,” *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8482–8495, 2019.

[13] Q. Chen, H. Gao, Z. Cai, L. Cheng, and J. Li, “Distributed low-latency data aggregation for duty-cycle wireless sensor networks,” *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2347–2360, 2018.

[14] T. T. Nguyen and H. Oh, “SCSMA: A Smart CSMA/CA Using Blind Learning for Wireless Sensor Networks,” *IEEE Trans. Ind. Electron.*, p. 1, 2019.

[15] T. T. Nguyen and H. Oh, “Constructing an optimally balanced tree to maximize data throughput with multiple channels,” *Wirel. Networks*, vol. 24, no. 3, pp. 993–1005, Apr. 2018.

[16] D. Cong Tan and M.-K. Kim, “Reliable Multi-Hop Linear Network Based on LoRa,” *Int. J. Control Autom.*, vol. 11, pp. 143–154, 2018.

[17] A. Zourmand, A. L. Kun Hing, C. Wai Hung, and M. AbdulRehman, “Internet of Things (IoT) using LoRa technology,” in *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 2019, pp. 324–330.

[18] C. Pan and H. Zhang, “A time efficient aggregation convergecast scheduling algorithm for wireless sensor networks,” *Wirel. Networks*, vol. 22, no. 7, pp. 2469–2483, 2016.

[19] S. Randhawa and S. Jain, “DAHDA: Dynamic Adaptive Hierarchical Data Aggregation for Clustered Wireless Sensor Networks,” *Wirel. Pers. Commun.*, vol. 97, no. 4, pp. 6369–6399, 2017.

[20] M. Arghavani, M. Esmaeili, M. Esmaeili, F. Mohseni, and A. Arghavani, “Optimal energy aware clustering in circular wireless sensor networks,” *Ad Hoc Networks*, vol. 65, pp. 91–98, 2017.

[21] S. Saginbekov and A. Jhumka, “Many-to-many data aggregation scheduling in wireless sensor networks with two sinks,” *Comput. Networks*, vol. 123, pp. 184–199, 2017.

[22] V. K. Singh, S. Verma, and M. Kumar, “Privacy Preserving In-network Aggregation in Wireless Sensor Networks.,” in *FNC/MobiSPC*, 2016, pp. 216–223.

[23] Mehak and P. Khandnor, “Structure and structure-free data aggregation protocols in wireless sensor networks-a review,” in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pp. 2136–2140.

[24] C. Li, J. Bai, J. Gu, X. Yan, and Y. Luo, “Clustering routing based on mixed integer programming for heterogeneous wireless sensor networks,” *Ad Hoc Networks*, vol. 72, pp. 81–90, 2018.