Packet Scheduling Based on Set Covering Problem Using Greedy Algorithm

Chatchanan Jandaeng

**A Thesis Submitted in Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Engineering
Prince of Songkla University
2012**

Thesis Title Packet Scheduling Based on Set Covering Problem Using Greedy Algorithm
Author Mr.Chatchanan Jandaeng
Major Program Computer Engineering

Major Advisor:

....
(Asst. Prof. Dr. Wannarat Suntiamorntut)

Co-advisor:

....
(Asst. Prof. Dr. Nittida Elz)

Examining Committee:

....Chairperson
(Asst. Prof. Dr. Ladda Prechaveerakul)

....
(Asst. Prof. Dr. Wannarat Suntiamorntut)

....
(Asst. Prof. Dr. Nittida Elz)

....
(Dr. Sakuna Charoenpanyasak)

....
(Dr. Kamol Kaemarungsri)

The Graduate School, Prince of Songkla University, has approved this thesis as fulfillment of the requirements for the Doctor of Philosophy in Computer Engineering.

....
(Prof. Dr. Amornrat Phongdara)
Dean of Graduate School

Thesis Title	Packet Scheduling Based on Set Covering Problem Using Greedy Algorithm
Author	Mr.Chatchanan Jandaeng
Major Program	Computer Engineering
Academic Year	2012

ABSTRACT

All sensor node shares a single channel using a multiple access protocols named CSMA/CA mechanism. When a sensor node receives more than one packet at the same time, these packets are termed collided, even when they coincide only partially. The collision between packets is likely to occur in the network with a higher density of neighbor nodes than the low density of neighbor nodes due to high competition for access to the communication channel between neighbor nodes. All collided packets are discarded. A solutions is scheduling the packets before sending.

The scheduling algorithm named SCGA (Set Covering Problem with Greedy Algorithm) is therefore introduced to solve this problem. Some scheduling algorithms can also influence and delay the data transmitting in the real-time wireless sensor networks. This thesis presents SCGA in order to reduce the number of packet in a MAC layer leading to reduce the overall of packet collision in the system. The SCGA is proved that it is set covering problem. And it can be solved by a greedy approximation method. The network topology is represented by undirected graph and transformed to a scheduling matrix. After that the number of frame length is minimized by a frame length minimization algorithm and throughput is increased by a throughput maximization algorithm.

The SCGA is compared to the existing works by mathematical method and network simulation method. In the mathematical method, the average delay of all algorithm is not significantly different. Although SCGA, which use frame length more than others, does not visually affect the average delay but SCGA provide the better slots allocation (refers to throughput in mathematical method) up to 10-30%. Moreover, channel utilization of SCGA has the best result, when this algorithm is running on 30 nodes benchmark. And result in the second place, when running on 40 nodes benchmark.

The NS-2 is used for SCGA evaluation in network simulation method. SCGA is implemented as extension module of NS-2 and applied to wireless sensor networks. Three performance metrics such as packet collision rate, throughput and end-to-end delay, are compared with simple CSMA/CA mechanism on standard benchmarks. Even though, throughput from both mechanism are not difference significantly when they are compared with statistical method, but SCGA decreases packet collision rate and clearly decrease end-to-end delay better than simple CSMA/CA mechanism.

Acknowledgements

This thesis is completely success because of the support by many people. I am appreciated and thankful to all:

Asst. Prof. Dr. Wannarat Suntiamorntut, the major advisor, suggested and consulted in academic issues, and proved the final report. Asst. Prof. Dr. Nittida Elz, the co-advisor, is more than just a teacher. But consult to all problems, not only academical issue but also other personal issue, occurred within this past six years.

All examining committee for comments, advices and approved for this completed thesis in difficult and complicate situation.

Assoc. Prof. Dr. Kreakchai Thongnoo, whom help in prepare the thesis proposal. All lecturers who had given all knowledge to all of us in order to create ways to greatness, in Department of Computer Science and Department of Computer Engineering.

The Teacher Development Scholarship Program for the Ph.D., by Walailak University, Thailand which supported the scholarship between 2007 until 2009. Information Communication and Technology Program, Prince of Songkla University, supported the Teacher Assistance Scholarship Program during study for Ph.D. period.

All master and bachelor students in Center of Excellence for Wireless Sensor Network (we called as UbiNES), they are both friends and brothers/sisters, especially all tennis team; Jim, Ja, Ju, Moo, and Nart.

Finally The Jandaeng family such as father, mother, sister and my fiance; Ms. Nattapicha Chayankiat, whom has a major support behind this success. Especially, when I am tired and given up, all these people are the power source to achieve all my goals.

Chatchanan Jandaeng

Contents

List of Figures

List of Tables

Abbreviations

ADC	Analog to Digital Converter
BSP	Broadcast Scheduling Problem
CAP	Contention Access Period
CBK	Call Back
CBR	Constant Bit Rate
CCA	Clear Channel Assessment
CFP	Contention Free Period
COL	Packet Collision
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
DUP	Duplicated Packet
FFD	Full Function Device
FSM	Finite State Machine
GTS	Guaranteed Time Slots
HNN-GA	Hopfield Neural Network and Genetic Algorithm
IFQ	Fully Interface Queue
LLC	Logical Link Control
LQI	Link Quality Indication
MAC	Media Access Control
MFA	Mean Field Anneal
NCNN	Noisy Chaotic Neural Network
NRTE	No Route
PAN	Personal Area Network
PHY	Physical Layer
RFD	Reduce Function Device
SCGA	Set Covering with Greedy Algorithm
SNR	Signal to Noise Ratio
SSCS	Service Specific Convergence Sublayer
SVC	Sequential Vertex Coloring
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network

CHAPTER 1
INTRODUCTIONS

The recent advances in micro-electro-mechanical (MEM) technology can yield the constraints of sensor technology. The sensor becomes tiny, low cost, and low power. This made sensors embedded everywhere around us. Each Sensor communicates over the network wirelessly. A wireless sensor network (WSN) is a self-configured network containing numerous small sensor nodes that are deployed in sensor field as shown in Figure 1.1.

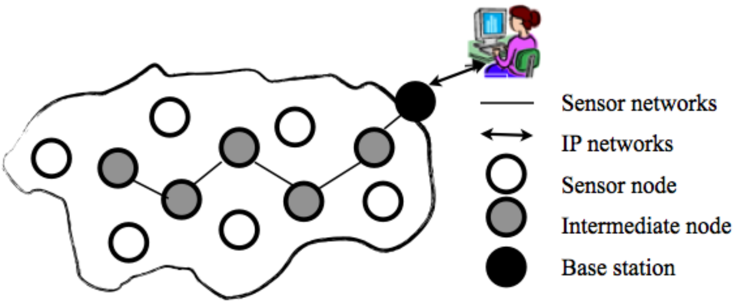


Figure 1.1: Wireless sensor nodes in sensor field

The wireless sensor nodes are categorized into three difference types: sensor node (source node), intermediate node, and base station (sink node). A sensor node or may be called source node or end-device should be implemented as a smallest node and attached the sensing module in order to collect the physical data in the field. After that the raw data may be preprocess on the sensor node or forwarded to the base station. Normally, the sensor nodes are fat from the base station and have to connect to the base station via the intermediate mode. The intermediate nodes act as a repeater or a router in the networks. In a large network, the intermediate node will act as a cluster head collecting the data from its member and forward to the base station. Thus, the intermediate node is higher performance than sensor node. The base station is a bridge between two networks: wireless sensor networks and IP networks. The sensor network has a limit of energy usage leading to communicate within a short range in contrast with the IP network. Base station can be either small device and attached to the PC or an embedded board. All collected data is stored in a data base at this point.

The components of sensor node as shown in Figure 1.2 are difference depend on function of node. Each node consists of sensing modules connected via ADC, microcontroller, external memory, radio transmitter and power sources [1].

The wireless sensor networks technology have been deployed in several applications such as health care monitoring system, and environment monitoring system [2]. In health care monitoring system, WSNs carry the promise of improving and enhancing the quality of life. The

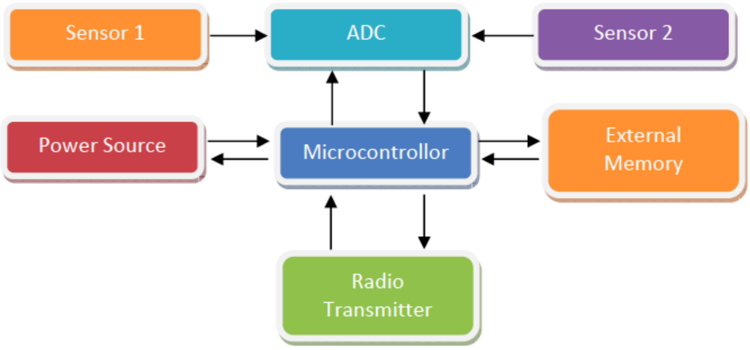


Figure 1.2: The components of sensor node

system will monitor and assist the elderly people in the independent-living resident. There are many research works that demonstrated their prototypes successfully.

The examples of the environment applications is tracking the movements of birds, small animals, and insects; monitoring environmental conditions that affect crops and livestock; irrigation; precision agriculture; forest fire detection; and disaster warning systems.

According to the constraints of WSNs, node has to consume low power dissipation. The main source of sensor node is battery, thus the energy is the lifetime indicator of sensor node. The 95% of energy consumption in wireless sensor node come from RF communication. There are many solutions in order to extend the lifetime of sensor node such as a data aggression to reduce the network usage. Moreover, several routing protocols bases on IEEE 802.15.4 standard have been proposal is to reduce the number of transmitting packages in order to save the energy.

According to the RF module based on IEEE 802.15.4 standard, the low data rate and unreliable network result in the packet corruption bring to retransmit the packet. Normally, the topology is frequently changed and prone to failures because the transmission power of the RF module in a sensor node is very low. Therefore, the packet delivery ratio is decreased. In the large or high density networks, there is a high possibility to have a large amount of packet collision. Moreover, each node has to transmit the data to the base station or sink node. Thus the collision is found around the sing node. These situations cause a lot of waste energy, and impede network performance.

1.1 Motivations

All sensor nodes share a single channel using a multiple access protocols. When a node receives more than one packet at the same time, these packets are termed collided. The collision is likely to occur in a high density of sensor node, especially when each node located closed to each other. All collided packets are discarded. Although some packets can be recovered, the retransmission causes the excessive energy waste. Thus, the packet collision degrades throughputs and increase delay/latency and energy consumption [3].

The collision has a high affect in wireless sensor network because the sensor nodes communicate using a low reliable media and energy is limited. Most applications in WSN require both high throughput and long network lifetime. However, the 95% of energy consumption in the sensor node is used by the communications module with a high potential of packet collision [4]. Thus there are many research works proposed to reduce this waste energy.

The one solution is a new MAC protocol proposed. This MAC protocol is able to avoid the packet collision. They called Spatial TDMA traffic-adaptive medium access protocol (TRAMA) [5]. While Sensor MAC (SMAC) [6] and Timeout MAC (TMAC) [7] were not flexible and only depend on the applications. The IEEE 802.15.4 MAC protocol [8] is the general purpose MAC protocol that be developed for low-power communication and applied the CSMA/CA to reserve and access the channel. This standard uses a super frame structure to manage radio access. The super frame is divided into sub-frames. The first and last frames have been reserved as the beacon slot. The others are in contention access period. Node will listen on the channel and will send the packet in the transmit queue, when the channel is found to be idle. On the other hand, node will wait the end of current transmission and stats the contention when channel is busy. Any devices wish to communicate during the contention access period between two beacons shall compete with other devices using a slotted CSMA/CA mechanism.

The random back off in CSMA/CA mechanism is activated when the channel is busy. Unfortunately, this technique is not efficient when using in a large scale of wireless sensor networks. When reserved nodes want to access the channel simultaneously, the network latency will be high. As a result, the throughput of the network is decreased. The packet collision problem is divided into two categories: directed collision and hidden collision. The detail of these two collisions will be described in the next chapter. The packet collision avoidance in CSMA/CA is effective only in the directed collision. Thus, the hidden collision is still existed.

To avoid the collision, All packets in network and higher layers should be scheduled before passing to MAC layer. The broadcast scheduling problem (BSP) [9] is defined as the scheduling of the transmissions using a minimum number of time slots and having a collision free. The time slot called frame has been assigned to each sensor node. The frame length essentially determines the packet average delay while the number of node that authorized in each frame dominated the throughput. For a fixed frame length, the channel utilization is determined by the number of simultaneous transmission of noninterferring nodes.

The BSP is NP-complete. We cannot get the optimal solution within the polynomial time. However we can get a optimal solution with an approximation method. Many approaches such as Mean Field Anneal (MFA) [10], Sequential Vertex Coloring (SVC) [11], Hopfield Neural Network and Genetic Algorithm (HNN-GA) [12], and Back tracking Sequential Coloring and Noisy Chaotic Neural Network (BSC-NCNN) [13] have been proposed. The concept of maximal compatibles and incompatibles [14] is used to find a schedule that will minimize the frame length and maximize the slot utilization in an integrated fashion. A optimal feasible broadcast schedule for TDMA networks [15] is solved by heuristic algorithm. A time slot per frame was increased to give a better throughput.

All above researched were proposed for an ad hoc network. The energy is not considered. Thus, they are not suitable for wireless sensor network that have limited energy and low processing power. In addition, sensor nodes are densely deployed and prone to failures. Sensor nodes mainly use a broadcast communication paradigm and local gossip whereas most ad hoc networks are based on point-to-point communications. The topology of sensor network are changed very often in terms of mobility and unreliable media.

The algorithm proposed here should be performed in the layer higher than MAC layer in order to decrease the packet congestion and avoid the packet collision. In addition, the algorithm has to operate under the constraints of wireless sensor networks. Therefore, it has to be simple, flexible, and low power.

1.2 Research Contributions

The goal of this research is to propose the packet scheduling algorithm in order to reduce packet congestion and packet collision for wireless sensor network. We propose the new packet scheduling algorithm called *(SCGA)*. The SCGA is approximation algorithm based on set covering problem and greedy algorithm. The algorithm is simple and easy to be implemented in sensor nodes. We compare the SCGA with the previous works using the network benchmarks with the mathematical and network simulation method. The SCGA can reduce packet collision and end-to-end delay. Moreover, the packet delivery ratio is increased.

According to the several works proposed to reduce the packet collision in wireless sensor networks, the performance prediction model in the thesis in order to report the performance when the parameters are changes. These models predict the packet collision rate, end-to-end delay, and packet delivery ratio by applying the multiple regression method. After that we analyze and compare the performance between the simple CSMACA algorithm and this algorithm improving with the SCGA. During the network implementation, the performance prediction model is able to help the developers to make a decision whether our algorithm is suitable to their application or not.

1.3 Research Organization

The rest of this dissertation is organized as follows.

- Due to the various subjects investigated in this research work, we review the theory and its related works in chapter 2. We survey and summary the wireless sensor networks in issues of applications, communication the concern with our works, the disadvantages of packet collision in wireless sensor networks. We review the broadcast scheduling problem and existed research. Finally, we survey the network simulator implemented the wireless sensor modules and present the experimental results of IEEE 802.15.4 Standard on NS2.

- In chapter 3, we propose the packet scheduling algorithm named SCGA. Our problem is also proved to be a set covering problem which is the one of approximation method. The complexity of the algorithm has been demonstrated in term of time complexity by big-O notation, code size and memory usage by implementation on Tmote Sky. SCGA implemented for Tmot Sky is simulated by MSPSim.

- We evaluate the performance of the SCGA in chapter 4. The mathematical results are proposed and compared to existed works with the three performance metrics. In the end of this chapter, the SCGA is simulated and compared with the existed works using network benchmarks.
- The performance of our packet scheduling algorithm has been concluded in this chapter. The limitation and future direction of our research work is also discussed here.

CHAPTER 2
THEORY AND RELATED WORKS

This chapter summarized the theory and related works. Firstly, the wireless sensor networks are introduced in topics of applications and IEEE 802.15.4 standard. Secondly, the broadcast scheduling problem is described using the system model. We explain and compare the characteristics of the existing network simulators. The structure of network simulator (NS-2) is also described here.

2.1 Wireless Sensor Networks

Wireless sensor networks have been applied in various applications such as military, environmental monitoring and health monitoring. Each sensor node sends a message wirelessly based on IEEE 802.15.4 standard. In this section, the applications of wireless sensor networks have been explained briefly. After that physical and MAC layer of the IEEE 802.15.4 standard are also described. Finally, we show the experimental results of IEEE 802.15.4 standard on NS-2.

2.1.1 Applications

Yick and et al [2] classified the wireless sensor network applications into two categories: monitoring and tracking. The example of monitoring applications are health care monitoring, power quality monitoring and environmental monitoring while the tracking application is concerned about the objects, animals, human and vehicles tracking.

In environmental applications, a large number of sensor nodes are deployed in order to collect the sensing data and monitor the impacts of urban and agricultural such as soil, water and sediments. All data has been processed for further analysis or prediction. MasiliNET is the example of the multi-model environmental monitoring system built for microclimate and pest monitoring in the olive groves [16]. ZigBee technology is used to transmit the sensing data such as humidity and temperature of the vegetable greenhouse [17]. Corke and et al [18] researched in the application of wireless sensor networks technology to prolong the network lifetime in a large scale environmental monitoring system. Their framework is applied for cattle monitoring, ground and lake water quality monitoring, virtual fencing, rainforest monitoring and so on.

Wireless sensor networks are also widely used in healthcare applications. The networks are constructed to monitor the patient physiological signals and health related information both in clinical and home environments. Wireless Body Area Network (WBAN) is designed to monitor the vital signals both inside and outside of a human body. Cao and et al [19] surveyed the pioneer WBAN research projects and enabling technologies. They explored the application scenarios, sensor devices, radio systems and interconnection of WBAN to provide the network coverage and energy efficiency. Whereas, Caldeira and et al[20] focused on a handover mechanism of a mobile network. They also proposed the handover mechanism for an infirmary hospital. A distributed telemonitoring system using wireless sensor networks was presented by Corchado and et al [21] in order to assist the people living independently. The system was a service-oriented architecture based platform which allows the heterogeneous wireless sensor networks to communicate in a distributed way. This approach enhanced the capability of the system in order to recover from the errors and achieve a better flexibility when the behaviors were changed at the execution time.

The last example of wireless sensor networks application is home automation. Wireless sensor networks technology delivers the exciting solution for building energy saving or home safer or more comfortable by using a wireless control and monitoring. The hundreds of nodes have to be scaled and deployed in homes. Thus the system has to be reconfigured easily. Finally, the power efficiency is required in some cases where node may not access to AC power.

2.1.2 IEEE 802.15.4 Standard

The IEEE 802.15.4 standard specification is published in 2003 and summarized in technical report by Ergen [22]. This standard contains the physical layer (PHY) and medium access control (MAC) sublayer specifications for low data rate wireless. This wireless network is used for data transmission between two simple devices that consume nominal power. The simple topology is the star topology in a short distance. The multi hop network is established when the distance between source and destination more that 10 meters. Moreover, this network is a self configuration. Each node has 64-bit IEEE address or 16 bit short address as identity. Wireless links under 802.15.4 can operate in three Industrial Scientific Medical (ISM) frequency bands. These accommodate over the air at the data rates of 250 kbps in the 2.4 GHz band, 40 kbps in the 915 MHz band, and 20 kbps in the 868 MHz. Total 27 channels are allocated in IEEE 802.15.4, with 16 channels in the 2.4 GHz band, 10 channels in the 915 MHz band, and 1 channel in the 868 MHz band.

The PHY Sublayer

The PHY layer provides an interface between the MAC sublayer and the physical radio channel. It provides two services, accessed through two service access points (SAPs). These are the PHY data service and the PHY management service. The PHY layer is responsible for the following tasks:

Activation and Deactivation of the radio transceiver: Turn the radio transceiver into one of the three states: transmitting, receiving, or off (sleeping) according to the request from MAC sublayer. The turnaround time from transmitting to receiving, or vice versa, should be no more than 12 symbol periods.

Link Quality Indication (LQI) for received packets:: Link quality indication measurement is performed for each received packet. The PHY layer uses the receiver energy detection, a signal-to-noise ratio, or a combination of these to measure the strength and/or quality of a link. However, the use of LQI value by the network or application layers is not specified in the standard.

Clear Channel Assessment (CCA) for Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA): The PHY layer is required to perform CCA using energy detection, carrier sense, or a combination of these. In energy detection mode, the medium will be busy if the energy value is greater than a predefined energy threshold. In carrier sense mode, the medium will be busy if a signal with the modulation and spreading characteristics of IEEE 802.15.4 is detected. And in the combined mode, both conditions aforementioned need to be met in order to conclude that the medium is busy.

Channel frequency selection: Wireless links under 802.15.4 can operate in 27 different channels (but a specific network can choose to support part of the channels). Hence the PHY layer should be able to tune its transceiver into a certain channel upon receiving the request from MAC sublayer.

Data transmission and reception: This is the essential task of the PHY layer. Modulation and spreading techniques are used in this part. The 2.4 GHz PHY employs a 16-ary quasi-orthogonal modulation technique, in which each four information bits are mapped into a 32-chip pseudo-random noise (PN) sequence. The PN sequences for successive data symbols are then concatenated and modulated onto the carrier using offset quadrature phase shift keying (O-QPSK). The 868/915 MHz PHY employs direct sequence spread spectrum (DSSS) with binary phase shift keying (BPSK) used for chip modulation and differential encoding used for data symbol encoding. Each data symbol is mapped into a 15-chip PN sequence and the concatenated PN sequences are then modulated onto the carrier using BPSK with raised cosine pulse shaping.

The MAC Sublayer

The MAC sublayer provides an interface between the Service Specific Convergence Sublayer (SSCS) and the PHY layer. Like the PHY layer, the MAC sublayer also provides two services, namely, the MAC data service and the MAC management service. The MAC sublayer is responsible for the following tasks: *Generating network beacons if the device is a coordinator*: A coordinator can determine whether to work in a beacon enabled mode, in which a superframe structure is used. The superframe is bounded by network beacons and divided into $aNumSuperframeSlots$ (default value 16) equally sized slots. A coordinator sends out beacons periodically to synchronize the attached devices and for other purposes.

Synchronizing to the beacons: A device attached to a coordinator operating in a beacon enabled mode can track the beacons to synchronize with the coordinator. This synchronization is important for data polling, energy saving, and detection of orphaning.

Supporting Personal Area Network (PAN) association and disassociation: To support self configuration, IEEE 802.15.4 is embedded the association and disassociation functions in its MAC sublayer. This does not only enable a star to be setup automatically, but also allows for the creation of a self configuring and peer-to-peer network.

Employing the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism for channel access: Like most other protocols designed for wireless networks, IEEE 802.15.4 standard uses CSMA/CA mechanism in order to access the multiple channel. However, the new standard does not include the Request-To-Send (RTS) and Clear-To-Send (CTS) mechanism as IEEE 802.11, in consideration of the low data rate used in LR-WPANs.

Handling and maintaining the Guaranteed Time Slot (GTS) mechanism: When working in a beacon enabled mode, a coordinator can allocate portions of the active superframe to a device. These portions are called GTSs, and comprise the Contention Free Period (CFP) of the superframe.

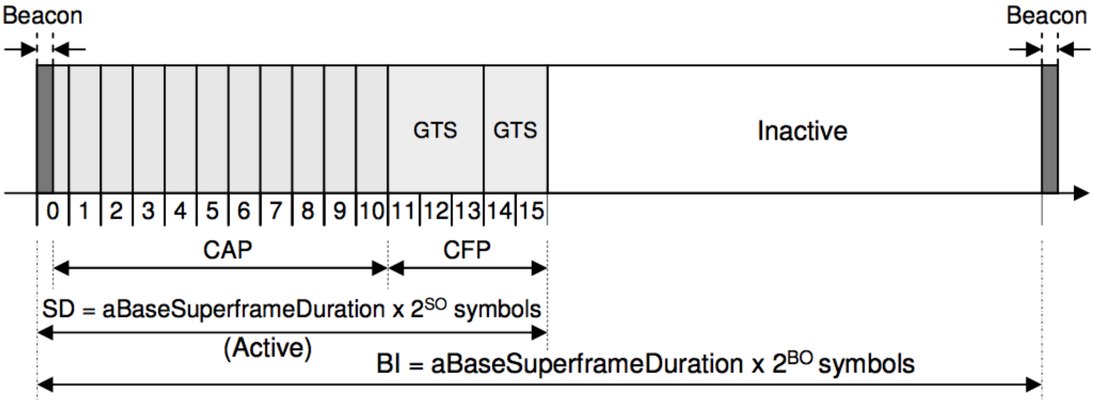

Beacon
Beacon
GTS
GTS
Inactive
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
CAP
CFP
SD = aBaseSuperframeDuration x 2^SO symbols
(Active)
BI = aBaseSuperframeDuration x 2^BO symbols

Figure 2.1: Superframe

Providing a reliable link between two peer MAC entities: The MAC sublayer employs various mechanisms to enhance the reliability of the link between two peers. The frames of the link are the acknowledgment frame and retransmission frame with the data verification by using a 16-bit CRC, as well as CSMA/CA.

General Functions

The standard gives detailed specifications of the following items: type of device, frame structure, superframe structure, data transfer model, robustness, power consumption considerations, and security.

Two different types of devices are defined in an 802.15.4 network, a full function device (FFD) and a reduced function device (RFD). An FFD can talk to RFDs and other FFDs, and operate in three modes serving either as a PAN coordinator, a coordinator or a device. An RFD can only talk to an FFD and is intended for extremely simple applications.

The standard allows the optional use of a superframe structure. The format of the superframe is defined by the coordinator. Figure 2.1 show that the superframe comprises an active part and an optional inactive part, and is bounded by network beacons. The length of the superframe called beacon interval (BI) and the length of its active part called superframe duration (SD) are defined as Equation 2.1 and Equation 2.2, respectively. Let $aBaseSuperframeDuration$ is 960 symbols whereas BO and SO are beacon order and superframe order, respectively.

$$BI = aBaseSuperframeDurationx2^{BO} \tag{2.1}$$

$$SD = aBaseSuperframeDurationx2^{SO} \tag{2.2}$$

The standard allows the optional use of a superframe structure. The format of the superframe is defined by the coordinator as shown inFigure 2.1. The superframe contains an active part and an inactive part. Superframe is bounded by network beacons.

The values of BO and SO are determined by the coordinator. The active part of the superframe is divided into $aNumSuperframeSlots$ (default value 16) equally sized slots and the beacon frame is transmitted in the first slot of each superframe. The active part can be further broken down into two periods, a contention access period (CAP) and an optional contention free period (CFP). The optional CFP may accommodate up to seven so-called guaranteed time slots (GTSs), and a GTS may occupy more than one slot period. However, a sufficient portion of the CAP shall remain for contention based access of other networked devices or new devices wishing to join the network. A slotted CSMA-CA mechanism is used for channel access during the CAP. All contention based transactions shall be complete before the CFP begins. Also all transactions using GTSs shall be done before the time of the next GTS or the end of the CFP.

Data transfer can happen in three different ways: (1) from a device to a coordinator; (2) from a coordinator to a device; and (3) from one peer to another in a peer-to-peer multi-hop network. The data transfer are classified into the following three types:

Direct data transmission: This applies to all data transfers, either from a device to a coordinator, from a coordinator to a device, or between two peers. The unslotted CSMA-CA or slotted CSMA-CA is used for data transmission, depending whether non-beacon enabled mode or beacon enabled mode is used.

Indirect data transmission: This only applies to data transfer from a coordinator to its devices. In this mode, a data frame is kept in a transaction list by the coordinator, waiting for extraction by the corresponding device. A device can find out if it has a packet pending in the transaction list by checking the beacon frames received from its coordinator. Occasionally, indirect data transmission can also happen in non-beacon enabled mode. For example, during an association procedure, the coordinator keeps the association response frame in its transaction list and the device polls and extracts the association response frame. Unslotted CSMA-CA or slotted CSMA-CA is used in the data extraction procedure.

GTS data transmission: This only applies to data transfer between a device and its coordinator, either from the device to the coordinator or from the coordinator to the device. No CSMA-CA is needed in GTS data transmission.

2.1.3 CSMA/CA Algorithm

IEEE 802.15.4 standard supports both slotted and unslotted CSMA-CA. In both cases, the algorithm uses the unit of time called *back-off* periods, which is equal to $aUnitBackoffPeriod$. The slotted CSMA/CA is used when superframe structure is used in the personal area network (PAN).

The boundary of the next back-off period is located when a device transmits a data frame during the CAP. Meanwhile, a sensor node in unslotted CSMA/CA does not use the beacon at the beginning. The back-off periods of one device do not need to be synchronized to the back-off periods of another device.

Each device has three variables to allocate channel such as BE;NB; and CW. The BE stands for the back-off exponent. The $macMinBE$ is between 0 and 3 and the default value is 3. If NB is greater than $maxMacCSMABackoffs$, the CSMA/CA algorithm will be terminated. The CW is initialized to 2 before transmitting. CW will resets to 2 when the channel is busy. The slotted CSMA/CA algorithm is shown in Figure 2.2.

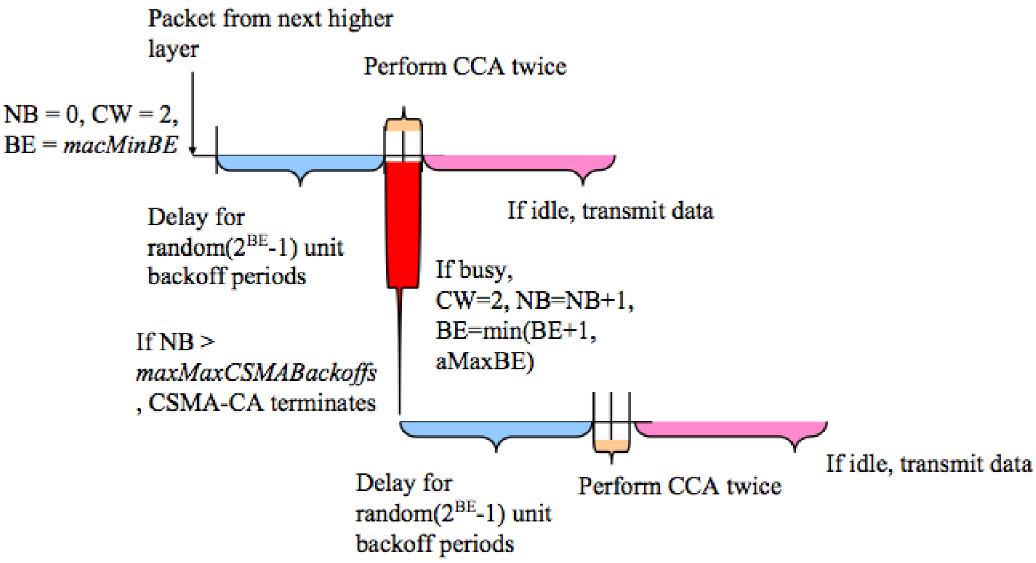

Packet from next higher layer
NB = 0, CW = 2, BE = macMinBE
Perform CCA twice
Delay for random(2^BE-1) unit backoff periods
If idle, transmit data
If busy, CW=2, NB=NB+1, BE=min(BE+1, aMaxBE)
If NB > maxMaxCSMABackoffs , CSMA-CA terminates
Delay for random(2^BE-1) unit backoff periods
Perform CCA twice
If idle, transmit data

Figure 2.2: Slotted CSMA/CA algortihm

Figure 2.2 shows that when source node sends a packet, the NB, CW, and BE are initialized to 0, 2, and 3, respectively. After that the sender delays for random $(2^{BE}-1)$ periods. The CCA method are start for two times to check channel. If channel is idle, the data are transmitted. On other hand, the NB is increased and BE is assigned to the minimal value between $BE+1$ and $aMaxBE$. The sender is delay again for random $(2^{BE}-1)$ periods, start CCA method for two times, and send data if channel idle. CCA method is active twice and the data is sent when the channel is idle. This process will be repeated for *maxMacCSMABackoffs* time. The transmission process will be terminated if NB is greater than maxMacCSMABackoffs.

2.2 Broadcast Scheduling Problem

All sensor nodes share a single channel using a multiple access protocols. When a node receives more than one packet at the same time, these packets are termed collided, even when they coincide only partially. The collision between packets is likely to occur in the network with a higher density of neighbor nodes than the low density of neighbor nodes due to high competition for access to the communication channel between neighbor nodes. All collided packets are discarded. A solutions is to schedule the packets before sending.

Ephremides and Truong defined broadcast scheduling problem (BSP) is the scheduling of the transmissions of all the station in a minimal number of time slots such as no collision among packed occur. Frame is the final arrangement of the station transmission into their assigned time slots. The frame structure is directly related to the network performance. First, the frame length is essentially determined the packet average delay. Second, the channel utilization is determined by the simultaneous transmission of noninterfering station for a fixed frame length.T Therefore, they will refer this number as a throughput. Fianlly, the BSP is to find the minimal frame length for a maximum throughput in wireless sensor network.

2.2.1 System Model

The wireless sensor networks is represented with undirected graph. After that it is transformed to scheduling matrix to avoid packet collision. At the end, the scheduling matrix is optimized in order to decrease delay and increase channel utilization.

A wireless sensor network can be represented by a undirected graph, $G = (V, E)$, where $V = \{1, 2, 3, ..., N\}$ represents the sensor nodes, whereas the set of undirected edge E characterizes the set of transmission links in the network. Note that, N is the total number of the sensor nodes. As a result, there exists an undirected edge $e = (i, j) \in E$ if two nodes are within the range of each other, which is also known as one-hop apart. If $(i, j) \notin E$, but there is an intermediate node k such that $(i,k) \in E$ and $(k, j) \in E$, then node i and j are two-hop apart.

Definition 2.1. The wireless sensor network can be described by an *NxN* symmetric connectivity matrix $\mathcal{T}$, which is defined as

$$t_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{otherwise.} \end{cases} \tag{2.3}$$

Definition 2.2. The corresponding compatibility matrix $\mathcal{C}$ can be obtained from matrix $\mathcal{T}$, and is defined as

$$c_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \vee \exists k_{k\in V}[(i,k) \in E \wedge (k,j) \in E] \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

Definition 2.3. For the broadcast scheduling problem, it requires a conflict-free and constraints-satisfied TDMA frame for packet transmission and this frame is repeated over time. Thus we assume that there are L time slots in each frame and use scheduling matrix $\mathcal{S}$ sized *LxN* to denote a TDMA frame, where the element is represented as

$$s_{li} = \begin{cases} 1, & \text{if } l^{th} \text{ time slot to be assigned to node } j \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

Follow this equation 4.1.1, the slot utilization of the whole network, η, is given by

$$\eta = \frac{1}{LN} \sum_{l=1}^{L} \sum_{i=1}^{N} s_{li} \tag{2.6}$$

The objective is to get an optimum TDMA cycle that has the minimum frame length L and the maximum slot utilization index, η, which is referred to as optimum broadcast scheduling problem in the following. More precisely, the BSP can be stated below :

Minimize L and maximize η,

subjects to

$$\sum_{l=1}^{L} s_{li} \geq 1, \forall i \tag{2.7}$$

$$\sum_{l=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{N} s_{li}s_{lj}c_{ij} = 0 \tag{2.8}$$

where L denotes frame length while η is channel utilization. Time slot s_{li} is the member in scheduling matrix sized N while c_{ij} is the corresponding matrix.

Equation 2.7 reflects the non-transmission constraint, which guarantee that every sensor node have to be assigned at least one time slot. Equation 2.8 which characterized the conflict-free constraint means that the two sensor nodes with one-hop or two hop part must be scheduled to transmit in the different time slots.

The minimum TDMA frame length depends on the actual topology, and generally is computationally intractable owing to its NP-complete. However, a tight lower bound for a frame length can be found easily and be estimated the minimum frame length. By defining the degree of a vertex i as the number of edges incident to it and denoting this as $deg(i)$, we use the following Lemma 2.9 [10].

Lemma 2.4. *The frame length L satisfies*

$$L \geq \triangle(G) + 1$$
$$where \triangle(G) = \max_{i \in V} \deg(i)$$

Proof. This lemma is proved by Wang and et. al. [10].

2.2.2 Reviews of Literature

The wireless sensor network is represented with graph theory and transformed to a scheduling matrix. The packet scheduling algorithm minimizes frame length of scheduling matrix leading to have a shorter waiting time for the next sending turn. The channel utilization is maximized by increasing the allocated time slots in scheduling algorithm. The BSP has been proved to be NP-complete combinatorial optimization problem as demonstrated in[10–13, 15].

The algorithms [10–13, 15] aim at finding both minimal frame length. After that throughput is maximized. Usually, two stages are adopted to tackle the two objectives in a separate fasions: frame length minimization and throughput maximization.

Wang and *et al* proposed the approximation method called Mean Field Anneal (MFA) to optimize the schedule matrix which divided into two phases: minimize frame length and maximize throughput. The main purpose of MFA is frame length minimization.

Yeo and *et al* [11] applied the Sequence Vertex Coloring (SVC) method to optimize packet scheduling matrix. The SVC was applied in wireless sensor network. All adjacent node is assigned to the different colors. The SVC problem is to find the minimum numbers of color. When the SVC is applied to BSP, the number of color will be represent the frame length. If the nodes are assigned with same color, they will be assigned to the same time slot. From their experimental results, SVC could reduce the average delay and increase the channel utilization better than MFA.

S. Salcedo-Sanz and *et al* [12] minimized the frame length with a Hopfield Neural Network (HNN) method, whereas, throughput is maximized with the combination of HNN and genetic algorithm. The HNN for the frame length minimization based on a random scheme and the state of neuron function was repeated until a feasible solution or user decided to halt the procedure. This algorithm was tested using the examples which it obtained the optimum fame lengths more than MFA. Unfortunately, authors did not discuss their algorithms in term of throughput and channel utilization.

Shi and *et al* [13] proposed a hybrid algorithm, which combined the Backtracking Sequential Coloring (BSC) and Noisy Chaotic Neural Network (NCNN) in order to solve the broadcast scheduling problem in IEEE 802.11 standard. They used two stages of optimization that different method according to the two objectives of the BSP. They also used the Backtracking Sequential Coloring to find the minimal TDMA frame length and Noisy Chaotic Neural Network to maximize channel utilization. This hybrid algorithm was evaluated using three benchmark examples and one large instance. The hybrid method gave better solutions than MFA, and HNN-GA in both frame length minimization and throughput maximization.

Ahmad and *et al* [15] proposed a heuristic algorithm in order to find an optimal feasible broadcast schedule for ad hoc TDMA network. The network is modeled with Finite State Machine (FSM) after that the problem is solved by the concept of maximal compatible. However, they had to find a schedule that would minimize the frame length and maximize the slot utilization in an integrated fasion. The concepts of the FSM is set which a single node are generated. After that, their sets were combined as lattice structure under the condition that all element in each set do not conflict the BSP. Finally, all sensor nodes were grouped. All sensor node in each group able to send packet in the same time without packet collision. The number of set indicates frame length while the total of elements in all sets is throughput. This heuristic algorithm minimized frame length and utilized channel better than the previous works mentioned in the paper. However, this approach has the big limitation for wireless sensor network. It requires a high computation power and consume the high memory. Moreover, it take the long computation time.

We can conclude that two performance parameters, average time delay (τ) and throughput (σ), are mostly used in the existing research works. The average time delay and number of slots in a TDMA frame of FSM are compared with the respective values of [10] - [15] which is presented in Table 2.1.

Table 2.1: Performance comparison

	15 nodes		30 nodes		45 nodes	
algorithm	τ	σ	τ	σ	τ	σ
MFA [10]	7.20	18	10.67	38	6.99	71
SVC [11]	7.20	18	9.99	37	6.76	60
HNN-GA [12]	7.00	20	9.30	35	6.30	77
BSC-NCNN [13]	6.80	-	9.20	-	5.80	-
FSM [15]	6.84	20	9.20	35	6.00	64

The main idea of FSM is to generate different time slot to each node. After that, the combination is used to reduce the unused time slot. Nodes which do not caused any collision will be combined. Thus the minimal time slots are generated and given the new schedule matrix which having both the minimal frame length and maximal throughput. The algorithm can perform effectively in every benchmark.

However, this algorithm may consumed a high computation power because of matching time slot functions. Moreover, a large memory is used when the time slot is generated. These are the main constraints in wireless sensor network. Therefore, the new algorithm must generate schedules within tight lower bound in a negligible time and require the low resources such as computation power and memory.

2.3 Network Simulation

Recently there has been growing the interest in providing a fine-grained metering and control of living environments using low power devices. Wireless sensor networks, which consist of spatially distributed self-configurable sensors, perfectly meet the requirements. Since running real experiments is costly and time consuming, the simulation is essential to study the protocols or algorithms in wireless sensor networks.

This section concludes the network simulator that supporting the wireless sensor network simulation such as NS-2 [23], TOSSIM [24], EmStar [25], OMNeT++ [26], J-Sim [27], ATEMU [28], and Avrora [29]. We also analyze and compare, shown in Table 4.3.

Table 2.2: Network simulator supporting wireless sensor networks

Name	Types	Simulation method	GUI	Commercial	Designed for WSN
TOSSIM	Emulator	Discrete-Event	Yes	No	Yes
EmStar	Emulator	Trace-Driven	Yes	No	Yes
OMNeT++	Simulator	Discrete-Event	Yes	No	No
J-Sim	Simulator	Discrete-Event	Yes	Yes	No
ATEMU	Emulator	Discrete-Event	Yes	Yes	No
Avrora	Simulator	Discrete-Event	No	Yes	No
NS-2	Simulator	Discrete-Event	No	No	No

Simulator is universally used to develop and test protocols of WSNs, especially in the beginning stage of these designs. The cost of simulating thousands of nodes networks is very low, and the simulation can be finished within very short execution time. Both general and specialized simulators are available for uses to simulate WSNs.

The tool, which is using firmware as well as hardware to perform the simulation, is called *emulator*. Emulation can combine both software and hardware implementation. Emulator implements in real nodes, thus it may provide more precision performance. Usually emulator has highly scalability, which can emulate numerous sensor nodes at the same time.

Discrete-event simulation is widely used in WSNs, because it can easily simulate lots of jobs running on different sensor nodes. Discrete-event simulation includes some of components. This simulation can list pending events, which can be simulated by routines. The global variables, which describe the system state, can represent the simulation time, which allow the scheduler to predict this time in advance. This simulation includes input routines, output routines, initial routines, and trace routines. In addition, this simulation provides dynamic memory management, which can add new entities and drop old entities in the model. Debugger breakpoints are provided in discrete-event simulation, thus users can check the code step by step without disrupting the program operation.

However, *Trace-Driven Simulation* provides different services. This kind of simulation is commonly used in real system. The simulation results have more credibility. It provides more accurate workload; these detail information allow users to deeply study the simulation model. Usually, input values in this simulation constant unchanged. However, this simulation also contains some drawbacks. For example, the high-level detail information increases the complexity of the simulation; workloads may change, and thus the representativeness of the simulation needs to be suspicious.

TOSSIM can support thousands of nodes simulation and can emulate radio models and code executions. Moreover, the power consumption can be simulated with PowerTOSSIM. However, TOSSIM only emulates homogeneous applications.

EmStar can not support large number of sensors simulation and only run in real time simulation. It is only applied to iPAQ-class sensor nodes and MICA2 motes.

OMNet++ supports MAC protocols and some localized protocols in WSN. More over it supports power consumptions simulation and channel controls. But it has limited available protocols.

J-Sim can simulate large number of sensor nodes, around 500 nodes and can simulate radio channels and power consumptions. The disadvantage is the execution time is much longer.

ATEMU can emulate different sensor nodes in homogeneous networks or heterogeneous networks and it can emulate power consumptions or radio channels. The simulation time is much longer like J-Sim.

Avrora support thousands of nodes simulation and can save much more execution time.

Finally, NS-2 can not simulate more than 100 nodes and can not simulate problems of the bandwidth or the power consumption in WSNs.

2.3.1 IEEE 802.15.4 Standard

The NS-2 is selected in this research because it is the open source simulator that be used in many researches especially designing network protocol. The main idea of this simulation is evaluation and comparison network performance between IEEE 80.15.4 standard, previous works, and our approach. The IEEE 802.15.4 module has been proposed by Jianliang Zhen [30] and included to be standard module in Network Simulator (NS2) to research in wireless sensor network protocol.

Figure 2.3 outlines the function modules in the simulator, and a brief description is given below for each of the modules. *Wireless Scenario Definition* selects the routing protocol; defines the network topology; and schedules events such as initializations of PAN coordinator, coordinators and devices, and starting (stopping) applications. It defines radio-propagation model, antenna

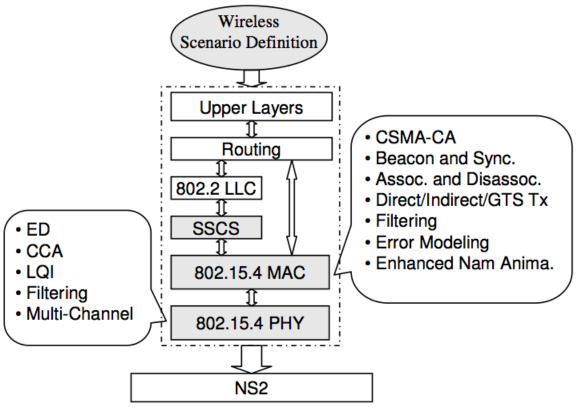


Figure 2.3: NS2 Simulator for IEEE 802.15.4

model, interface queue, traffic pat- tern, link error model, link and node failures, superframe structure in beacon enabled mode, radio transmission range, and animation configuration. *Service Specific Convergence Sublayer (SSCS)* is the interface between 802.15.4 MAC and upper layers. It provides a way to access all the MAC primitives, but it can also serve as a wrapper of those primitives for convenient operations. It is an implementation specific module and its function should be tailored to the requirements of specific applications. *802.15.4 PHY* implements 14 PHY primitives such as energy detection, clear channel assessment, and link quality indication *802.15.4 MAC* is the main module which implements 35 MAC sublayer primitives i.e. as CSMA/CA, beacon and synchronization services and association and disassociation services.

2.3.2 Experimental Results

To study the nature of IEEE 802.15.4 standard, the wireless sensor network is simulated by NS-2 to study packet flow and packet drop. The 15, 30, and 40 nodes are randomly placed in the area of 40x40 m^2. The transmission range of each node is 10 meters. An error model has not been considered in the study. Every node will send the first packet at 5 seconds and continue along Constant Bit Rate. The experimental results are classified and analyzed by the flow id to study the packet flow and dropped packet.

Packet Flow

A CBR packet is firstly passed from the application layer (called AGT in NS-2) to MAC sublayer as shown in Figure 2.4. The CBR packet has to know its route before it is forwarded to lower layer. Source node finds the destination node using a routing protocol (AODV protocol is used in the study). When it has route for packet, the CBR packet is forwarded to MAC layer. If there are many source node want to access the channel in the same time, the CSMA/CA mechanism is used for the communication channel access. The CBR packet is sent via RF channel. However, some packets cannot send to the destination node directly. They are forwarded by route node or intermediate node At the destination node, the CBR packet is forwarded to AGT in order to process in the next step.

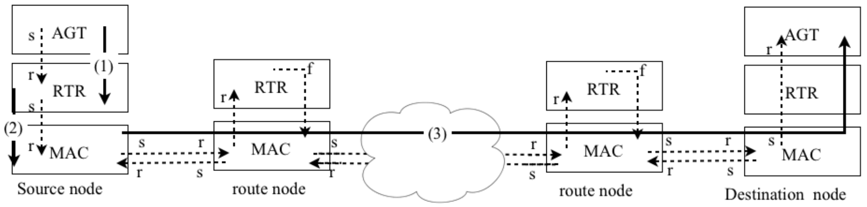
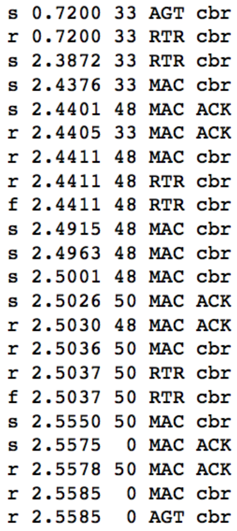


Figure 2.4: Packet flow and drop of IEEE 802.15.4 in NS-2

The simulation results are generated and saved to trace file. The necessary columns have been filtered as shown in Figure 2.5. There are long line format in trace file and take the huge storage. Therefore, we filter the necessary columns as show in Figure 2.5.



```
s 0.7200 33 AGT cbr
r 0.7200 33 RTR cbr
s 2.3872 33 RTR cbr
s 2.4376 33 MAC cbr
s 2.4401 48 MAC ACK
r 2.4405 33 MAC ACK
r 2.4411 48 MAC cbr
r 2.4411 48 RTR cbr
f 2.4411 48 RTR cbr
s 2.4915 48 MAC cbr
s 2.4963 48 MAC cbr
s 2.5001 48 MAC cbr
s 2.5026 50 MAC ACK
r 2.5030 48 MAC ACK
r 2.5036 50 MAC cbr
r 2.5037 50 RTR cbr
f 2.5037 50 RTR cbr
s 2.5550 50 MAC cbr
s 2.5575  0 MAC ACK
r 2.5578 50 MAC ACK
r 2.5585  0 MAC cbr
r 2.5585  0 AGT cbr
```

Figure 2.5: Flow id 0 of node ID 33

Figure 2.5 shows the flow id 0 of source node number 33^{th} that consists of fives columns: 1) packet types (send (s), receive(r), forward(f), and drop (D)), 2) time in second unit, 3) the event occurs in node id, 4) protocol name and 5) packet type (CBR or ACK). As show in Figure 2.5, node 33^{th} starts to send a CBR packet from AGT. Then the CBR packet is forwarded via RTR and MAC layer. Finally, the CBR packet is sent to the neighbor node (node 48^{th}). The packet is forwarded from node 48^{th} to 50^{th} and stoped at the destination node (node 0^{th}).

Packet Drop

Dropped packet in wireless sensor networks occurs frequently in every layers of OSI model. The reasons of dropped packets may be caused by packet collision, queue full, link failure, or duplicated packet and so on. The retransmission mechanism is active as soon as the network found dropped packets. This effect will increase the end-to-end delay and decrease the throughput. The packet is dropped in the data link layer because of three reasons: link quality indication (LQI), packet collision (COL), and duplication packet (DUP). Meanwhile, the reasons of dropped packet in the network layer are no route, call back error from data link, and route loop.

In the data link layer, link quality indication or LQI is measured from the signal to noise ratio (SNR) of wireless channel. When the SNR is less than the acceptable limits and preferably greater than or equal to the capture threshold, the packet will be accepted. Otherwise, the packet is dropped with the indicated drop. There are two significant reasons for these drops: directed collision and hidden collision. The direct collision occurs when two or more nodes want to allocate channel in the same time and can detect the presence of other node. The hidden collision likely occurs direct collision but two or more node cannot detect the presence of other nodes and transmit unaware of transmissions of other nodes. Whereas DUP indicates that a duplicate packet has been received. The duplicate packet is then dropped with this error message.

In network layer, No Route Error (NTTE) indicates no route has been discovered to the indicated node, during the routing request phase such as packet jam in network layer that result in RREQ and RREP message of AODV cannot be sent. Moreover, the position of sensor node is random which bring to some node is not connected to other nodes in sensor field.

Call back from MAC or CBK indicates the MAC layer do not able to transmit the packet. Hence MAC layer informs the upper layer about the transmission failure. The possible reasons for the transmission fail are: failure in accessing the channel, following the CSMA-CA mechanism, cannot receive acknowledgement for the transmission, or transaction expired.

Moreover, the importance reason of packet dropping in network layer is fully interface queue called IFQ. IFQ indicates the queue is full due to the excessive transmission rate. Note that, the queue management is FIFO. The first packet is added to the queue. If the packet arrives after the queue is full, the packet will be dropped.

The LQI and COL are the main reasons of dropping in data link layer whereas NRTE is the reason of dropped packet in network layer. Moreover the interface queue size is the other point of dropping packet in simulation.

Figure 2.4 shows the packet flow in the IEEE 802.15.4 standard. In addition, the figure also shows the drop points of packet flow. The packet is dropped in three phases: 1) AGT-to-RTR, 2) RTR-to-MAC, and 3) MAC to next node. The LQI and COL are the main reasons of dropping in data link layer whereas NRTE is the reason of dropped packet in network layer. Moreover the interface queue size is the other point of dropping packet in simulation. Figure 2.4 shows packet flow of IEEE 802.15.4 standard. In addition, the figure show the drop points of packet flow. This figure shows that packet drop occurs in three phases: 1) AGT-to-RTR, 2) RTR-to-MAC, and 3) MAC to next node.

The objective of source node is to send CBR packet from the agent level or application level to the destination node. In the first phase, the packet is forwarding from the top of stack to the network layer. The source node has to route the path of destination before sending. The AODV routing protocol is selected in this study. The RREQ and RREP messages of AODV are broadcasted to find the destination node. At this point, the routing packet are dropped when there are many packets in the channel. The retransmission is enabled. Finally, some node cannot find the destination and stop sending. We note that 45% of CBR packet is are dropped because of NRTE.

In the second phase, the CBR packet is forwarded from RTR layer to MAC layer via logical link layer (LL) and interface queue (IFq). The CBR packet and AODV packet are dropped because of the full queue. Moreover, many packets are not performed within the simulation time. Their packets are discarded and marks as END in trace file. Although we increase the interface queue size, the IFQ drop still occurs. From the study, we found that the queue sized 450 slots is the maximum queue length. The performance of network does not change eventhough the queue size is more that 450 slots.

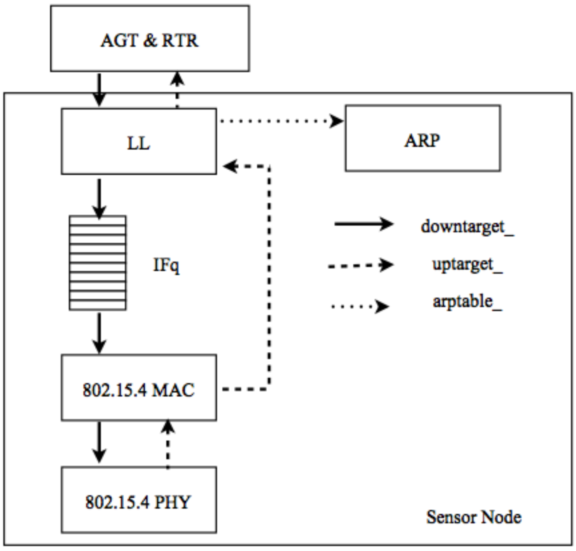


Figure 2.6: Box diagram of sensor node in NS-2

Link quality indication causes the dropped packet in the third phase. Both CBR and AODV packets are sent from the source node to the destination via intermediate node. Packets are forward to network layer to establish the route path. If the path is in the routing table of intermediate node, the packet is forwarded. On the other hand, the route discovery mechanism is begun if there is no route to the destination. All mechanisms increase the packets in the channel, especially AODV packets. Moreover, the control packets such as AODV packets are dropped in this phase causing the dropped packet in network layer such as NRTE and CBK etc.

2.3.3 Performance of IEEE 802.15.4 Standard

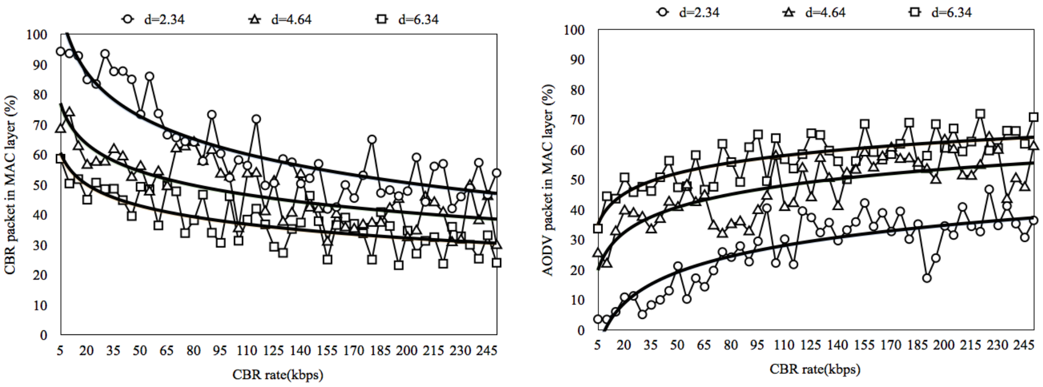
To study the performance of IEEE 802.15.4 standard, the simulation is set up with NS-2. The performance metrics consists of in packet collision, end-to-end delay, and throughput. More information about the parameters is described in the next chapter.

We have addressed only two domains: CBR rate and density of node called dense. The packet in MAC layer is directly proportional to CBR rate. Whereas the dense is the averaged number of neighbor node. The number of node is varied from 15, 30, and 40 and randomly placed in the sensor field 40x40 m^2. The averaged neighbor node is 2.34, 4.64, and 6.34 nodes of the overall

15, 30, and 40 nodes respectively.

The Number of Packet in MAC Layer

There are two important reasons of dropped packet in network layer: NRTE and IFQ. The dropping problem in MAC layer comes from NRTE in network layer whereas IFQ is the resource limitation. The low link quality affects the routing mechanism. And no route problem affects directly to the throughput because of dropped packet in network layer. Moreover, fully interface queue affects both throughput and control packet such as AODV.



(a) CBR packet

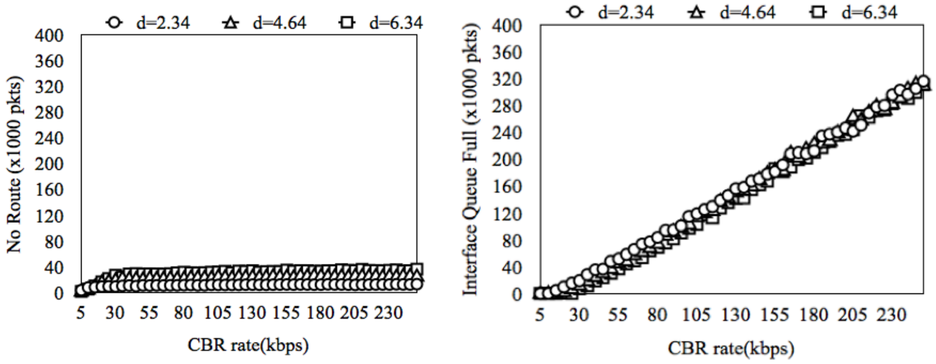
(b) AODV packet

Figure 2.7: Packet in MAC layer

Figure 2.7 shows that the CBR rate does not affect to NRTE. But the CBR rate affects to IRQ. NRTE happens when the packet is dropped in MAC layer. The AODV protocol cannot find the path to the destination node. Thus the density of node will effect to NRTE. NRTE is double increased when the density of node is doubling. While the CBR rate is increasing, the growth of IFQ is linear function. The number of packet forwarded to network layer is high, when the CBR rate in application layer is high. The comparison between NRTE and IFQ shows that IFQ directly varies the CBR rate while the NRTE does not affect to the CBR rate. Moreover, we found that only CBR packets are dropped in network layer whereas the AODV packets are kept in the queue until the simulation is stop and they are not processed.

Dropped Packet in Network Layer

There are two important reasons of dropped packet in network layer: NRTE and IFQ. The dropping problem in MAC layer comes from NRTE in network layer whereas IFQ is the resource limitation. The low link quality affects the routing mechanism. And no route problem affects directly to the throughput because of dropped packet in network layer. Moreover, fully interface queue affects both throughput and control packet such as AODV.



(a) no route

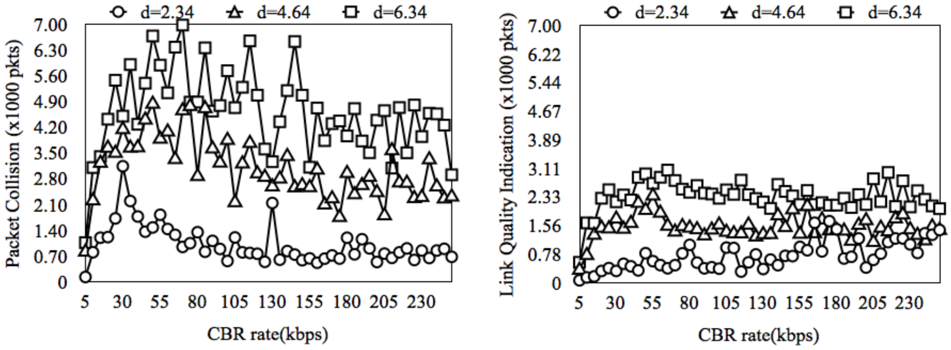
(b) fully interface queue

Figure 2.8: Packet drop in network layer

Figure 2.8 shows that the CBR rate does not affect to NRTE. But the CBR rate affects to IRQ. NRTE happens when the packet is dropped in MAC layer. The AODV protocol cannot find the path to the destination node. Thus the density of node will effect to NRTE. NRTE is double increased when the density of node is doubling. While the CBR rate is increasing, the growth of IFQ is linear. The number of packet forwarded to network layer is high, when the CBR rate in application layer is high as shown in Figure 2.8 b. The comparison between NRTE and IFQ shows that IFQ directly varies the CBR rate while the NRTE does not affect to the CBR rate. Moreover, we found that only CBR packets are dropped in network layer whereas the AODV packets are kept in the queue until the simulation is stop and they are not processed.

Dropped Packet in MAC Layer

The main reason of no route is the interruption of communication in data link layer. There are two reasons: packet collision (COL) and low link quality indicator (LQI) in NS-2. The experimental results are shown in Figure 2.9 and Figure 2.10.

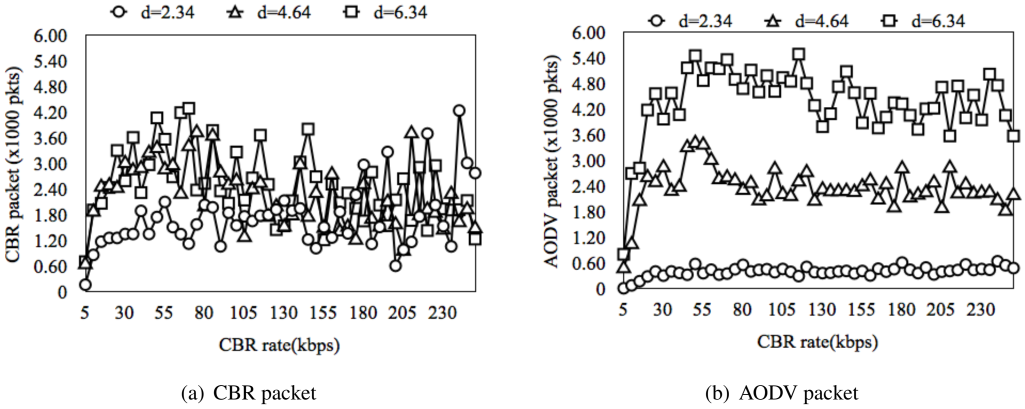


(a) packet collision

(b) low link quality

Figure 2.9: Packet drop in MAC layer classified by causes

Figure 2.9 is the dropped packet in MAC layer which classified by causes. This figure shows that the density of node is the main reason of packet collision. The packet collisions are increasing when the dense is increasing. The CBR rate does not significant affect to the packet collision and low link quality since the CBR packets are filled in network layer before passing to MAC layer. However, the number of packet collision is doubly greater than LQI.



(a) CBR packet

(b) AODV packet

Figure 2.10: Packet drop in MAC layer classified by type

Figure 2.10 is the classification of dropped packet based on the packet types: CBR packet and AODV packet. Both CBR and AODV packet are dropped up to 95% of dropped packet in MAC layer. The CBR packet is dropped at 70-80% while AODV are dropped at 10-20% for the low density of node. When the density of node is high, the number of dropped AODV packet is increased, in contrast with the CBR packet.

2.4 Summary

Wireless sensor networks have many limitation even they are deployed in various applications. Moreover, there is a scheduling problem because of the nature of wireless sensor networks. We summarized the system model of BSP and the previous works. Moreover, the network simulator (NS-2) is selected in this thesis.

The study of IEEE 802.15.4 characteristic using NS-2 has been explained in the last section. From the study, we found that the CBR rate and density of node affect directly to the number of packet in MAC and network layer. The density of node directly affects the number packet in MAC layer more than the CBR packet. The CBR packet does not directly affects the dropped packet in MAC layer. But it affects the dropped packet in network layer. On the other hand, the density of node is the main reason of packet collision and low link quality.

In the next chapter, the proposed packet scheduling algorithm named SCGA will be introduced. Our algorithm is a set covering problem and greedy algorithm (SGCA) which is able to solve the BSP.

CHAPTER 3
THE PACKET SCHEDULING BASED ON SET COVERING WITH GREEDY ALGORITHM

The proposed packet scheduling algorithm called packet scheduling algorithm using Set Covering and Greedy Algorithm (SCGA) is the approximation method as set covering problem. The network topology is represented with an undirected graph and transformed to the scheduling matrix with *PACKET-COLLISION-FREE* function. After that, the number of time slot is minimized with *FRAME-LENGTH-MIN* function and increased the throughput with *THROUGHPUT-MAX function*. This chapter will explain those three functions and analyze the complexity of the SCGA using Big-O and the memory usage of the real-implementation on the sensor node, named TmoteSky.

3.1 Packet Collision Free

Packet collision free phase transforms the network topology, that represented by graph theory, to be scheduling matrix under the packet collision free. The columns of matrix are sensor node, while the rows of matrix are time slots. More over, the members in matrix are represented with colored status: black, gray and white. The black status denoted the allocate slot for node. On the other hand, the white status denoted the unauthorized node for sending packet in this time slot. While gray nodes are in the matrix at the first step and then changed to be black or white status depends on frame length minimization phase and throughput maximization phase.

Definition 3.1. The wireless sensor network topology is represented with undirected graph, $G = (V,E)$ where V denotes set of sensor nodes and E denotes set of edge. Let $\mathcal{U} = \{u_1, u_2, u_3, ..., u_N\}$ is set of time slots sized N, where N denotes the number of sensor node. The time slot, u_i, is the initial time slot that allocated for node i and follows Equation 3.1, whereas u_{ij} denote status of node j in time slot i.

$$\forall j_{j \in V} u_{ij} = \begin{cases} BLACK, & \text{if } i = j \\ WHITE, & \text{if } (i,j) \in E \\ WHITE, & \text{if } (i,j) \notin E \wedge \exists k_{k \in V} [(i,k) \in E \wedge (k,j) \in E] \\ GRAY, & \text{otherwise.} \end{cases} \tag{3.1}$$

All node in each time slot are labeled with different color as defined in Equation 3.1. Black node able to send any packet with guarantee of no collision. In the initial stage, time slot j is reserved for node j. Therefore, u_{ij} is black when i and j are equal. The second line in Equation 3.1 prevent directed collision whereas the third line prevents the hidden collision. Gray node is defined in the initial status with no guarantee packet collision. However, all gray node will change to be white or black with combine operation.

Definition 3.2. $\mathcal{D}$ is set of status code in node status. Let $\mathcal{D} = \{00, 10, 11\}$, where 00, 10 and 11 denote WHITE, GRAY, and BLACK status, respectively.

Definition 3.3. $v(x)$ is function for return the status code. Let $v(x) = x \,\&\, 0x01$ where x denotes status of node in time slot as defined in Definition 3.2. This function will be used in SCGA as $s_{ij} = v(u_{ij})$.

Lemma 3.4. *The scheduling matrix, $\mathcal{U}$, relied on Definition 3.1, which able to prevent both directed and hidden packet collision problem in wireless sensor network.*

Proof. We proved that the scheduling matrix, $\mathcal{U}$, which generated by Definition 3.1 must rely on TDMA scheduling matrix in Equation 2.7 and Equation 2.8.

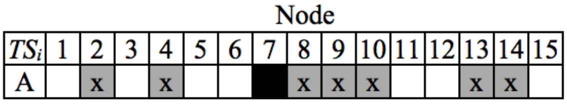
Let matrix $\mathcal{U}$ is generated by Definition 3.1 while scheduling matrix $\mathcal{S}$ is generated from $s_{ij} = v(u_{ij})$.

Firstly, we prove that $\sum_{l=1}^{L} s_{li} \geq 1, \forall i_{i \in V}$ when $\forall l_{l \in \mathcal{U}} \exists i_{i \in V} [u_{li} = BLACK$ and $u_{li} = BLACK \rightarrow s_{li} = 1]$. First, because of $1 \leq l \leq L$, $1 \leq i \leq N$ and $L = N$, thus there have at least one pair of i and l are equal that resulting in u_{li} is set to black. Second, from Definition 3.1, node j in time slot l is black mean that time slot l is grant permission for node i that result in s_{li} is 1. The first reason claim that every node $i \in V$ is set with black at least one time when i is equal l and the second reason support that the black node is set to 1. Therefore, we can conclude that $\sum_{l=1}^{L} s_{li} \geq 1, \forall i_{i \in V}$.

Secondly, we prove that $\sum_{l=1}^{L} \sum_{i=1}^{N} \sum_{j=1}^{N} s_{li} s_{lj} c_{ij} = 0$. Let s_{li} or s_{lj} is 0 when u_{li} or u_{li} is white, respectively. This equation is more than zero if only if the three parameters: s_{li}, s_{lj}, and c_{ij}, are one in the same time. However, both s_{li} and s_{lj} are not be one simultaneously when c_{ij} is one. The c_{ij} is one mean that node i and j are one hop or two hop nodes. Then, this time slot are not be allocated for both s_{li} and s_{lj} in the same time. As explain above, we conclude that the second assumption is truth.

From the two reasons above, we conclude that the Definition 3.1 allocate the free time slot to all sensor node m that prevent the directed collision and hidden collision □

Figure 3.1 shows that the time slot consist of 15 nodes, which allocated to node 7 (see network topology in Figure 3.2) in order to assure that all node should be assigned at least one time slot. We define all together in total three sets: directed collision set ($\{3, 12\}$), hidden collision set ($\{1, 5, 6, 11, 15\}$) and unknown set ($\{2, 4, 8, 9, 10, 13, 14\}$). All node in directed and hidden collision sets are assigned with white color, which means that all node in both set do not allow to send the packet in this time slot. When at least 2 nodes in unknown set send packet simultaneously, packet collision between these nodes may occur. Thus, all node in unknown set are set to gray color in the beginning step.



Node

TS_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A		x		x			■	x	x	x			x	x	

Figure 3.1: A time slot defined in Packet Scheduling Algorithm

Algorithm Description

Initially, the network topology is represented by the undirected graph $G = (V,E)$ when V denotes the set of sensor node, and E denotes the set of edge. The graph G is the input of PACKET-COLLISION-FREE() function. Whereas the scheduling matrix, $\mathcal{U}$, is the result of PACKT-COLLISION-FREE() function.

Algorithm 1 is algorithm of PACKET-COLLISION-FREE() function. The square scheduling matrix , $\mathcal{U}$, consists of rows and columns sized N. In each row has a list of time slots, u_l. While the u_{li} is the status of node i in time slot l, which assigned by a color status as explained in Definition 3.1. The scheduling matrix has frame length sized N slots.

Algorithm 1 PACKET-COLLISION-FREE()

```
for $l \in V$ do
    set GRAY to all member of list, $u_l$
    for $i \in V$ do
        if $(l,i) \in E$ then
            $s_{li} = WHITE$
            for $k \in V$ do
                if $(k,i) \in E$ then
                    $u_{lk} = WHITE$
                end if
            end for
        end if
    end for
    $u_{ll} = BLACK$
    $\mathcal{U} = \mathcal{U} \bigcup \{u_l\}$
end for
```

In Algorithm 1, all node in time slot, u_l, has set to unknown status with gray color. Then all adjacency node, $(l,i) \in E$, is set to white color in order to prevent the direct collision. After that all adjacency node, $(k,i) \in E$, will be set to white color in order to prevent the hidden collision. Finally the node, u_{ll}, is set to black color which means that this time slot is allocated for node l. In result time slot, u_l, will be add to the schedule matrix, $\mathcal{U}$. This algorithm will be repeated for all sensor node in V. At the end of this algorithm, the results is the scheduling matrix, $\mathcal{U}$, sized N.

Time Complexity

The time complexity analysis of algorithm is presented by big-O notation. The simple method to generate big-O notation is counting the primitive instruction in algorithm which called one time unit.

For Algorithm 1, let L_i denotes timing for one instruction or one time unit and T_i denotes timing for many instructions. Considering the inner loop of Algorithm 1, founds that there are three inner loops which analyzed as follow:

- The code in the loop from line 6^{th} to 10^{th} has two instructions and they are looped for N times. Thus the operation time is:

$$\begin{aligned} T_7 &= L_7 + L_8 && (3.2)\\ T_6 &= N * T_7 && (3.3)\\ &= N * (L_7 + L_8) && (3.4) \end{aligned}$$

- The instruction from line 3^{rd} to 12^{th} consists of simple instructions in line 4^{th} and 5^{th}, and loop instruction in line 6^{th}. Thus the operation time is:

$$\begin{aligned} T_4 &= L_4 + L_5 + T_6 && (3.5)\\ T_3 &= N * T_4 && (3.6)\\ &= N * (L_4 + L_5 + T_6) && (3.7)\\ &= N * (L_4 + L_5 + N * (L_7 + L_8)) && (3.8) \end{aligned}$$

- The instruction from 1^{st} to 15^{th} consists of instruction in line 2^{nd} and loop instruction in line 3^{rd} The instruction in line 2^{nd} is not the simple instruction. This instruction has more complexity. However, we selected the bitwise operation in order to reduce complexity that may occur. Finally this instruction is the simple command. The operation time of this loop is:

$$\begin{aligned} T_2 &= L_2 + T_3 && (3.9)\\ T_1 &= N * T_2 && (3.10)\\ &= N * (L_2 + T_3) && (3.11)\\ &= N * (L_2 + N * (L_4 + L_5 + N * (L_7 + L_8))) && (3.12) \end{aligned}$$

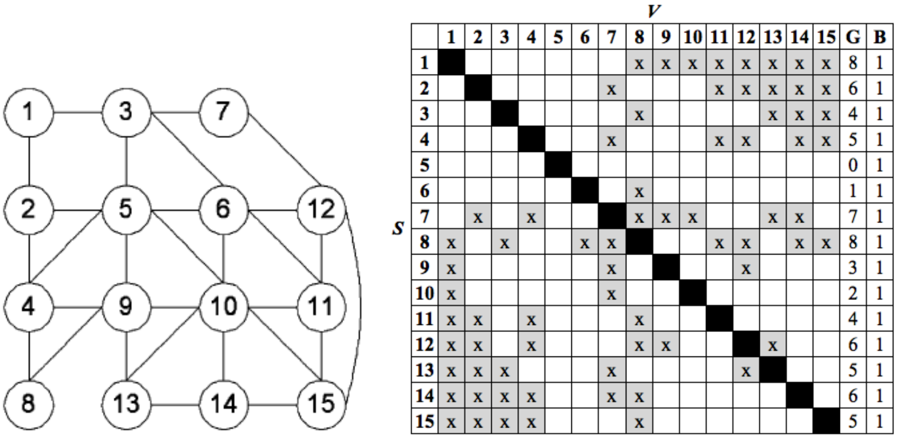
The overall timing of Algorithm 1 is:

$$\begin{aligned} T_1 &= N(L_2 + N(L_4 + L_5 + N(L_7 + L_8))) && (3.13)\\ &= N(1 + (N(2 + (2N)))) && (3.14)\\ &= N + N(N(2 + (2N))) && (3.15)\\ &= N + N(2N + 2N^2) && (3.16)\\ &= N + 2N^2 + 2N^3 && (3.17) \end{aligned}$$

We conclude that the big-O notation of Algorithm 1 is $O(N^3)$.

Example

When we want to find the scheduling matrix of network topology in Figure 3.2a which consists of 15 nodes. While the scheduling matrix of network show in Figure 3.2b. This scheduling matrix sized 15x15 consists of 15 black slots. Each node is granted as one time slot to send packet. Moreover, there are 70 gray time slots which can be changed in the next algorithm. The frame length is 15 frame and can be optimized when Algorithm 2 is applied. Whereas the gray time slots are changed to black or white color using Algorithm 4.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	G	B
1	■							x	x	x	x	x	x	x	x	8	1
2		■					x				x	x	x	x	x	6	1
3			■					x					x	x	x	4	1
4				■			x				x	x		x	x	5	1
5					■											0	1
6						■		x								1	1
7		x		x			■	x	x	x			x	x		7	1
8	x		x			x	x	■			x	x		x	x	8	1
9	x						x		■			x				3	1
10	x						x			■						2	1
11	x	x		x				x			■					4	1
12	x	x		x				x	x			■	x			6	1
13	x	x	x				x					x	■			5	1
14	x	x	x	x			x	x						■		6	1
15	x	x	x	x				x							■	5	1

(a) The topology for 15 nodes

(b) Scheduling matrix initiation

Figure 3.2: Throughput and channel utilization

3.2 Frame Length Minimization

Wang *et al* [10] proved that finding the optimal solution of broadcast packet scheduling problem is a NP-complete problem. Many approximation methods are proposed to find an optimized scheduling matrix. In this section, we proved that frame length minimization phase is the set covering problem. This problem can be solved by greedy algorithm which also one of approximation methods.

Regarding to Wang *et al* has proposed, let $\mathcal{R}$ is subset of scheduling matrix $\mathcal{U}$. Therefore, $|\mathcal{R}| \leq |\mathcal{U}|$. The main objective of SCGA is to find the minimum $\mathcal{R}$ following lemma 2.4 as show in Equation 3.18:

$$\triangle(G) + 1 \leq M \leq |\mathcal{R}| \leq |\mathcal{U}| \tag{3.18}$$

where $\triangle(G) = \max_{i \in V} \deg(i)$ and M is the minimum frame length.

The main idea of the algorithm is to find the minimum subset of U which under condition as follow: all member in each subset will only valid in the Equation 2.7, Equation 2.8, and Equation 3.18. The above process is here by called as set covering problem. The frame length minimization in broadcast scheduling problem is also represented as *set covering problem.* In this algorithm, we examine a simple greedy heuristic method with a logarithm ratio bound. We found that the size of the approximate solution may grow relatively to the size of an optimal solution.

In the beginning of this section, we proved that the frame length minimization problem is the set covering problem. Then we propose the greedy approximation algorithm and its greedy choice function. Finally, we will give the minimum frame length of scheduling matrix in Figure 3.2b as an example.

Lemma 3.5. *The frame length minimization problem in BSP can be solved with set covering approximation method.*

Proof. An arbitrary instance $(\mathcal{U}, \mathcal{F})$ of the set covering problem consists of a finite set $\mathcal{U}$ and a family of $\mathcal{F}$ of subsets of $\mathcal{U}$, such that every elements of $\mathcal{U}$ belongs to at least one subset in $\mathcal{F}$:

$$\mathcal{U} = \bigcup_{F \in \mathcal{F}} F$$

The problem is to find the minimum size subset $\mathcal{S}_{opt} \in \mathcal{F}$ whose members cover all of $\mathcal{U}$.

$$\mathcal{U} = \bigcup_{S \in \mathcal{S}_{opt}} S$$

We note that the $\mathcal{S}_{opt}$ is the minimal frame length scheduling matrix.

Let $\mathcal{F} \subseteq \mathcal{P}(\mathcal{U}) - \varepsilon$ or $\mathcal{F} = \{F_1, F_2, F_3, ..., F_p\}$, where $p \leq 2^{|\mathcal{P}(\mathcal{U})|} - 1$. All time slot, $F_p = \{u_i \in \mathcal{U} | \sum_{i=1}^{|F_p|} \sum_{j=1}^{|F_p|} s_{pi} s_{pj} c_{ij} = 0\}$, able to combine in order to reduce frame length under the packet collision free condition. Therefore, we found a minimum size of subset $\mathcal{S}_{apx} \in \mathcal{F}$ which their member cover all of $\mathcal{U}$ under the packet collision free condition or does not conflict with Equation 2.7 and Equation 2.8.

As explained above, we conclude that the frame length minimization problem can be solved by set covering approximation method and following Equation 3.19.

$$\mathcal{U} = \bigcup_{S \in \mathcal{S}_{apx}} S \tag{3.19}$$

under two conditions as follow:

$$L_{min} = \min |\mathcal{S}_{apx}|$$

$$W_{max} = \sum_{S \in \mathcal{S}_{apx}} gray(S)$$

where W_{max} is the maximum summation of gray node in a optimal scheduling matrix, while $gray()$ function will return the number of gray color in time slot □

A Greedy Approximation Algorithm

The greedy approximation method consists of core function called greedy choice function in order to select the valid subset from $\mathcal{U}$ and store in S. Then subset S will be added into an optimal scheduling matrix S_{apx}. This process is going to be repeated over the unselected time slot in T. This algorithm will terminate if only if T is an empty set. In the end of Algorithm 2, an optimal scheduling matrix is returned from algorithm as S_{apx}.

Algorithm 2 FRAME-LENGTH-MIN()

```
$\mathcal{T} = \mathcal{U}$
$\mathcal{S}_{apx} = \phi$
while $T \neq \phi$ do
    $S = GREEDY\_DECISION(T)$
    $\mathcal{S}_{apx} = \mathcal{S}_{apx} \bigcup S$
    $\mathcal{T} = \mathcal{T} - S$
end while
return $\mathcal{S}_{apx}$
```

For the greedy choice will be described in next section. Algorithm 2 base on approximation method with logarithm lower bound function, which we have already proved in Lemma 3.6.

Lemma 3.6. *FRAME-LENGTH-MIN($\mathcal{U}$) is the* $\ln|\mathcal{U}|$*-approximation algorithm.*

Proof. We have already show that the FRAME-LENGTH-MIN() provides suboptimal and gives solution more than the optimal solution for $ln(|\mathcal{U}|)$ times.

Let $\mathcal{S}_{opt}$ denotes the minimal set that cover $\mathcal{U}$, while S_{apx} denotes a minimal set from the approximation algorithm and let $n = |\mathcal{U}|$. We prove that $|\mathcal{S}_{apx}| = |\mathcal{S}_{opt}| \ln n$.

After the round k^{th} of Algorithm 2, where n_k denotes $|\mathcal{U}|$ after k^{th} round and n_0 is n slots. We will select the covered set at least $n/|\mathcal{S}_{opt}|$ time slots, which the number of time slot $|S|$ of each round in the algorithm must have at least $n_k/|\mathcal{S}_{opt}|$. For example, there are 15 nodes ,n_k, in this round and the minimum time slot ,$\mathcal{S}_{opt}$, is 8 slots. Therefore, the number of time slots, S_{max}, must be $15/8 = 1.875 \approx 2$ time slots.

After $(k+1)^{th}$ round, the remaining time slots will be less than $n_k - \frac{n_k}{|\mathcal{S}_{opt}|}$ or $n_k(1 - \frac{1}{|\mathcal{S}_{opt}|})$ time slots.

$$\begin{aligned} n_k &\leq n_{k-1}(1 - \tfrac{1}{|\mathcal{S}_{opt}|}) \\ &\leq n_{k-2}(1 - \tfrac{1}{|\mathcal{S}_{opt}|})^2 \\ &\;\;\vdots \\ &\leq n_0(1 - \tfrac{1}{|\mathcal{S}_{opt}|})^k \end{aligned} \tag{3.20}$$

The n_k are decreased for k rounds that bring to $n_k = 0$ or $n_k < 1$. Moreover, $n_0 = n$ and $1 - x \leq e^{-x}$. Therefore, we can rewrite the Equation 3.20 and show in Equation 3.21 as below:

$$\begin{aligned} n(e^{-1/|\mathcal{S}_{opt}|})^k &< 1 \\ ne^{-k/|\mathcal{S}_{opt}|} &< 1 \\ \frac{1}{e^{k/|\mathcal{S}_{opt}|}} &< \tfrac{1}{n} \\ e^{k/|\mathcal{S}_{opt}|} &\geq n \\ k/|\mathcal{S}_{opt}| &\geq \ln n \end{aligned} \tag{3.21}$$

where k round indicates the $|\mathcal{S}_{apx}|$ of approximation method. Therefore we can conclude that $|\mathcal{S}_{apx}| = \ln|\mathcal{U}|.|\mathcal{S}_{opt}|$ □

The Greedy Decision

GREEDY_DECISION() function in Algorithm 2 is the core function which indicate the performance of scheduling algorithm: frame length and channel utilization. This function returns the set of time slots, which validated and obtained the maximum gray node in cover set. The validated time slot means that all time slot does not conflict with the combine operation which defined in Definition 3.7.

Definition 3.7. Let time slot A and B rely on packet collision free that defined in Lemma 3.4 and d_{ai} and d_{bi} are the node i^{th} in time slot A and B, respectively, where $d_{ai} \in \mathcal{D}$ and $d_{bi} \in \mathcal{D}$. $R = A \uplus B$ means that time slot A and B are combined and assigned to time slot R as defined in Equation 3.22

$$R = \{d_{ri} | i \in V, d_{ai} \uplus d_{bi}\} \tag{3.22}$$

The combine operation $\uplus$ is used for time slot combination in Equation 3.22. The expression, $d_{ai} \uplus d_{bi}$, is able to get result from Table 3.1.

Table 3.1: The combine operation

$\uplus$	00	10	11
00	00	00	00
10	00	10	11
11	00	11	11

All time slot in cover set able to be combined with combine operation. The combination process must be under two conditions: 1) the combined of time slot must still prevent packet collision and 2) the number of black node in cover set must be more than the summation of black node before combination process. Thus, Lemma 3.8 proves that the result from combination process still prevent packet collision as defined in Equation 2.7 and Equation 2.8. Whereas Lemma 3.9 proves that the number of black node in cover set is not decreased.

Lemma 3.8. *$R = A \uplus B$ represents the combination process between time slot A and B which assigned to time slot R under condition that time slot R still prevent directed and hidden collision.*

Proof. Let two time slots A and B rely on the packet collision free as shown in Lemma 3.4. Therefore, time slot A and B are $\sum_{i=1}^{N}\sum_{j=1}^{N} v(u_{ai})v(u_{aj})v(c_{ij}) = 0$ and $\sum_{i=1}^{N}\sum_{j=1}^{N} v(u_{bi})v(u_{bj})v(c_{ij}) = 0$, respectively. We prove that

$$\sum_{i=1}^{N}\sum_{j=1}^{N} v(u_{ri})v(u_{rj})v(c_{ij}) = 0 \tag{3.23}$$

There are three parameters:$v(u_{ri})$, $v(u_{rj})$ and $v(c_{ij})$ for each round in Equation 3.23. The summation in this equation is more than zero if only if multiplication of three parameters is more than zero at least one time. In addition, the multiplication is more than zero if only if all parameter are one at the same time. Moreover, both $v(u_{ri})$ and $v(u_{rj})$ cannot be one simultaneously when $v(c_{ij})$ is one. Because $v(c_{ij})$ is one means that node i and j are one hop or two hop nodes. Therefore, the time slot is not allocated to both $v(u_{ri})$ and $v(u_{rj})$ at the same time. And it brings to the summation of multiplication to zero, which means that the combination process still prevent directed and hidden collision □

Lemma 3.9. *The combination process obtains the number of gray node more than the number of gray node before combination.*

Proof. Let T_A and T_B denote the number of black node in time slot A and B, respectively which $T_A = \sum_{i=1}^{N} v(u_{ai}) \geq 1$ and $T_B = \sum_{i=1}^{N} v(u_{bi}) \geq 1$. We prove that

$$T_R > max(T_A, T_B) \tag{3.24}$$

where, $R = A \uplus B$ and $T_R = \sum_{i=1}^{N} v(u_{ri})$

If the combination process occurs under condition that the packet collision free (as explain in Equation 2.8 and $\sum_{i=1}^{N}\sum_{j=1}^{N} v(u_{ri})v(u_{rj})v(c_{ij}) = 0$) and rely on Lemma 3.8, then the number of black node will be increased only. Because the black node is changed to be white node in combination process mean that the node unable to send packet in that time slot which conflict the condition that all node has to allocate the time slot at least one time slot. Therefore, $T_R = T_A + T_B > \max(T_A, T_B)$.

Algorithm Description

The $GREEDY\_DECISION()$ function selected the time slot, which contains the maximum gray node. Therefore, all time slot is sorted with any sorting algorithm at the first step. After that the $GREEDY\_DECISION()$ function will selecte the second time slot, which contained the maximum gray node. The selected time slot will be able to combine with the first time slot. The number of gray color of the new time slot must be more than gray color of time slot before combination process. If the second time slot is not valid, the third time slot will be selected, combined and tested as the previous time slot. These steps are repeated until there are no more time slots. If the number of cover time slot is less than the ratio between number of uncover time slot and the optimal frame length, then these selected cover set is ignored. After that the greedy selection function is restarted, the second, third, ..., time slot is selected relatively.

Let $\mathcal{T}$ denotes list of row in scheduling matrix $\mathcal{U}$, while S denotes subset of $\mathcal{T}$. All members in S that are selected time slots, which validated. And it will be returned to called function. Algorithm 3 show greedy choice function whereas it is described in Lemma 3.8 and Lemma 3.9.

Algorithm 3 GREEDY_DECISION($\mathcal{T}$)

```
1:  $sort(\mathcal{T}, gray)$
2:  $L = \phi$
3:  loop
4:      $S = getFirst(\mathcal{T} - L)$
5:      $L = L \bigcup \{S\}$
6:      $K = \{S\}$
7:      loop
8:          $B = getFirst(\mathcal{T} - K)$
9:          if $B = NULL$ then goto 15
10:         $K = K \bigcup \{B\}$
11:         $R = S \biguplus B$
12:         if $T_R \leq \max(T_S, T_B)$ then goto 8
13:         $S = R$
14:     end loop
15:     if $|S| < |\mathcal{T}|/M$ then goto 3
16:     return S
17: end loop
```

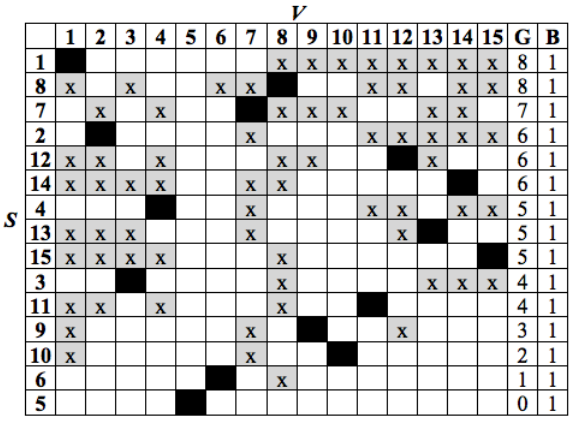
The unselected slots in $\mathcal{T}$ are sorted by number of gray node. The first time slot is selected from $\mathcal{T} - L$ by $getFirst()$ function and then assign to S. After that it will be added to L and K in order to prevent the time slot re-selection. The second slot is selected by $getFirst()$ function from $\mathcal{T} - K$ which both assigned to B and buffered with K. Let time slot R is the result of combination process between S and B. The time slot R is valid if only if the T_R is more than $\max(T_S, T_B)$ when T_R represents the number of black color in R. On the other hand, the second time slot in line 8^{th} is re-selected, combined, and tested by line 12^{th}. If $getFirst()$ function can not get any time slot with valid conditions and size of S is more than $|\mathcal{T}|/M$, the covered set is claimed that this selected set follows set covering problem as explained in Lemma 3.5. In the end of algorithm, the covered time slot in S will be returned to called function.

Example

The scheduling matrix $\mathcal{U}$ in Figure 3.3 is the result of *PACKET-COLLISION-FREE()*. The time slot in scheduling matrix are sorted by number of gray node. The column, which labeled G, is the number of gray node in each slot while the column, which labeled B, is the number black node. The maximum degree of this network is 7. Therefore, the lower bound of the optimal solution M is 8.

The time slot 1^{st} in Figure 3.3 is the first time slot which has been selected and assigned to S, L, and K in the first round, while the time slot 8^{th} is the second time slot from T and it has been selected and assigned to B. The time slot S and B are combined and assigned to time slot R as shown in Figure 3.4.

Because the T_R in Figure 3.4 is 2 and it is more than $max(T_S, T_B)$, thus the combination between slot 1 and 8 is valid. After that, the next time slot will be selected in the next round.

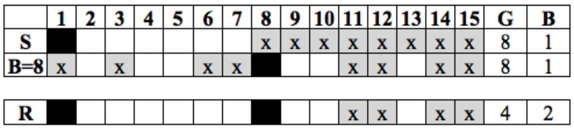


V

S

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	G	B
1	■							x	x	x	x	x	x	x	x	8	1
8	x		x			x	x	■			x	x		x	x	8	1
7		x		x			■	x	x	x			x	x		7	1
2		■					x				x	x	x	x	x	6	1
12	x	x		x				x	x			■	x			6	1
14	x	x	x	x			x	x						■		6	1
4				■			x				x	x		x	x	5	1
13	x	x	x				x					x	■			5	1
15	x	x	x	x				x							■	5	1
3			■					x					x	x	x	4	1
11	x	x		x				x			■					4	1
9	x						x		■			x				3	1
10	x						x			■						2	1
6						■		x								1	1
5					■											0	1

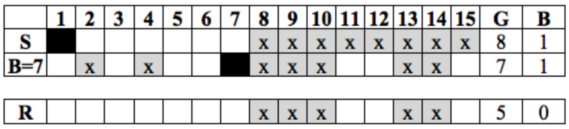
Figure 3.3: Sorted scheduling matrix



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	G	B
S	■							x	x	x	x	x	x	x	x	8	1
B=8	x		x			x	x	■			x	x		x	x	8	1
R	■							■			x	x		x	x	4	2

Figure 3.4: Valid 2 time slots combination

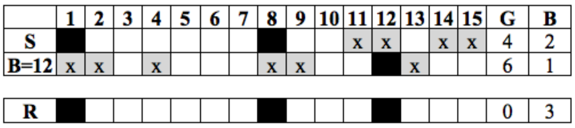
The time slot 7^{th} is not selected although it has the maximum gray node because T_R is 2, which conflict with the line 12^{th} in Algorithm 3 as shown in Figure 3.5. The number of black node in combined time slot must be more than the previous time slots in order to guarantee that all sensor node granted at least one time slot for packet transmission. The time slot is not selected when the number of black node in new time slot is not more than the maximum black node of S or B as show the example in Figure 3.5. The time slot 1^{th} and 7^{th} are not combined because the number of black node from new time slot is zero. This situation indicates that this time slot does not allow any node to send any packet.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	G	B
S	■							x	x	x	x	x	x	x	x	8	1
B=7		x		x			■	x	x	x			x	x		7	1
R								x	x	x			x	x		5	0

Figure 3.5: Invalid 2 time slots combination

As the reason above, the time slot 12^{th} is selected instead. Figure 3.6 shows the combination of slot 1, 8, and 12. Time slot S is result of combination process from previous step, while time slot 12^{th} is selected in this round. The combination process is valid, which the result in frame length is eliminated up to 2 slot and black node is generated up to 3 node.

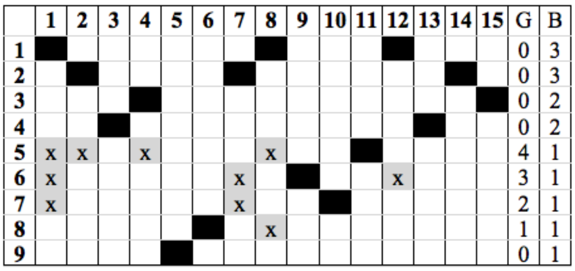


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	G	B
S	■							■			x	x		x	x	4	2
B=12	x	x		x				x	x			■	x			6	1

R	■							■				■				0	3

Figure 3.6: Valid 3 time slots combination

The line 14^{th} is an importance condition. When algorithm can not select any time slot that follow the condition, the time slot S is validated as shown in Lemma 3.6. If $|S|$ is less than $|\mathcal{U}|/M$, then the algorithm is restarted in Algorithm 3.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	G	B
1	■							■				■				0	3
2		■					■							■		0	3
3				■											■	0	2
4			■										■			0	2
5	x	x		x				x			■					4	1
6	x						x		■			x				3	1
7	x						x			■						2	1
8						■		x								1	1
9					■											0	1

Figure 3.7: Frame length minimization

The result of this algorithm is subset of time slot, which is covered and validated the condition. The *GREEDY_DECISION()* function is repeatedly called by *FRAME-LEGTH-MIN()* function in Algorithm 2.

Time Complexity

The time complexity of Algorithm 2 is very easy to evaluation because it has only one loop sized S_{apx}. However, the $GREEDY\_DECSION()$ function is called in this function that proved in Lemma 3.6. The time complexity of $GREEDY\_DECSION()$ function is $O(ln|\mathcal{U}|)$ where $\mathcal{U}$ denote scheduling matrix. While the time complexity of Algorithm 2 is $O(S_{apx})$. Therefore, we conclude that the time complexity of frame length minimization phase is $O(|S_{apx}|\ln|\mathcal{U}|)$.

3.3 Throughput Maximization

The scheduling matrix from frame length minimization phase still contains gray color in time slot. Throughput maximization is performed to increase the number of black node in scheduling matrix by replacing gray node with black or white color in order to increase the throughput. However, the node replacement in this phase must follow Definition 3.7. Their gray node can be changed to black status in order to increase throughput.

Algorithm Description

The input of this phase is the scheduling matrix, S_{apx}, which is the result from frame length minimization phase , as show in figure 3.7. The idea of the this algorithm is the gray node elimination by node replacement. The gray node is able to changed to other color under the condition of packet collision free as the previous phase. At the end of this phase, the new scheduling matrix, S, will be composed entirely of black and white slots, as shown in Algorithm 4.

Initially, the scheduling matrix, S_{apx}, is sorted by number of gray node. The scheduling matrix is column order traversal regarding to find the gray color. If node j of time slot i is gray node, the $\mathcal{U}_i$ is combined with time slot S_j under the condition in Lemma 3.8 and Lemma 3.9. This combination process result in the gray node, s_{ji}, is changed to black or white node. This algorithm will be repeated until there are no gray node in scheduling matrix.

Algorithm 4 THROUGHPUT-MAX()

```
 1: $S = sort(S_{apx}, gray)$
 2: $i = 0$
 3: while $i < |V|$ do
 4:    $j = 0$
 5:    while $j < |S_{apx}|$ do
 6:       if $S_{ji} = GRAY$ then
 7:          $R = S_j \uplus \mathcal{U}_i$
 8:          if $T_R > \max(T_{S_j}, T_{\mathcal{U}_i})$ then
 9:             $S_j = R$
10:          end if
11:       end if
12:       $j = j + 1$
13:    end while
14:    $i = i + 1$
15: end while
16: return S
```

Time Complexity

When we consider the inner loop of Algorithm 4, there are two inner loops. Algorithm 4 can be analyzed as following:

- The first line is the algorithm for sorting time slots by number of gray node. The big-O notation of this line is $O(|S_{apx}|ln(|S_{apx}|))$.
- The number of instruction in the loop from line 5^{rd} to 13^{th} are presented with T_1. These instructions are executed for $|S_{apx}|$ round. Therefore, the time unit of these instruction is shown below:

$$T_6 = L_6 + L_7 + L_8 + L_9 + L_{12} \tag{3.25}$$

$$T_5 = |S_{apx}|.T_6 = 5|S_{apx}| \tag{3.26}$$

- From line 3^{rd} to 15^{th} consist of three instructions that all instruction is in the loop. Therefore, the time unit are shown below:

$$T_4 = L_4 + T_5 + L_{14} \tag{3.27}$$

$$T_3 = N.T_4 \tag{3.28}$$

The big-O notation of Algorithm 4 is calculated from:

$$
\begin{aligned}
T_1 &= L_1 + L_2 + T_3 && (3.29)\\
&= |S_{apx}|.ln(|S_{apx}|) + 1 + N.T_4 && (3.30)\\
&= |S_{apx}|.ln(|S_{apx}|) + 1 + N(2 + 5.|S_{apx}|) && (3.31)\\
&= |S_{apx}|.ln(|S_{apx}|) + 1 + 2N + 10.N.|S_{apx}| && (3.32)
\end{aligned}
$$

We conclude that the big-O notation of Algorithm 4 is $O(N.|S_{apx}|)$.

Example

Figure 3.8a is the example of input in Algorithm 4. There are 9 time slots in scheduling matrix and it contains total 10 gray nodes. Only time slot 5, 6, 7 and 8 contain the gray nodes. All time slot is sorted by number of gray node. Then, the scheduling matrix is using column ordered transversal in order to find and change gray node to black node.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	G	B
5	x	x		x				x			■					4	1
6	x						x		■			x				3	1
7	x						x			■						2	1
8						■		x								1	1
1	■							■				■				0	3
2		■					■							■		0	3
3				■											■	0	2
4			■										■			0	2
9					■											0	1

(a) input matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	G	B
1	■							■				■				0	3
2		■					■							■		0	3
3				■											■	0	2
4			■										■			0	2
5	■							■			■					0	3
6	■								■			■				0	3
7	■									■						0	2
8						■		■								0	2
9					■											0	1

(b) output matrix

Figure 3.8: Example of throughput maximization

For example, the node 1, 2, 4 and 8 in time slot 5^{th} are gray nodes. Node S_5 is selected and combined with original time slot that allocated in node 1^{st}, $\mathcal{U}_1$. If the combination between S_5 and $\mathcal{U}_1$ validates with condition in line 8^{th} in Algorithm 4, the gray node will be changed to black color. On the other hand, if the combination is not valid, the gray node will be changed to white color. All gray node with valid the conditions are changed to be black node while others are change to be white node. Because of reason above, the node 1 and 8 are combined with S_5. whereas node 2 and 4 are not combined with S_5. This process will be repeated until there no gray node in scheduling matrix. Finally, an optimal scheduling matrix is enhanced by throughput maximization, as shown in figure 3.8b. At the end, the throughput of the scheduling matrix is 21 nodes, which id increased up to 40%

3.4 The Memory Usage of SCGA in Tmote Sky

Our SCGA algorithm has been developed on Contiki [31] operating system which is a famous tiny operating system on TmoteSky. In this thesis, C programming is used and the MSPSim (the virtual Tmote Sky [32]) has been deployed to measure the memory usage of process.

Contiki is a well-known operating system for wireless sensor networks as this operating system is very small, which consume 2 Kbytes of RAM and 40 Kbytes of ROM. We can use the Contiki on MSPSim directly after it is compiled. Whereas MSPSim is a microcontroller simulator based on MSP430 family. Moreover MSPSim performs together with CooJa [33] which is a network simulator.

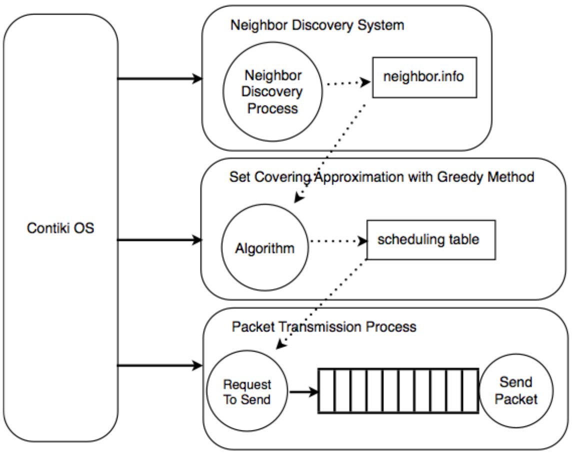

Contiki OS
Neighbor Discovery System
Neighbor Discovery Process
neighbor.info
Set Covering Approximation with Greedy Method
Algorithm
scheduling table
Packet Transmission Process
Request To Send
Send Packet

Figure 3.9: SCGA module on Contiki Operating System

Figure 3.9 shows box diagram of SCGA that implemented on Contiki operating system. The Neighbor Discover System (NDS), SCGA and packet transmission process are setup for this experiment.

Code Size

The Contiki will activate NDS in order to get the neighbor information within the range of 2-hop. This information will be used to form the scheduling table. After that, Contiki will enable SCGA and start the scheduling process. The output of this process is the scheduling table. When a node requires to send the packet, it will check the sequence with the scheduling table. Then it waits for its time slot.

The size of the SCGA code on Tmote Sky is about 200 Kbytes. In order to measure how big of the SCGA code is, the *hello_world* coding is built and the size of *hello_world* code is about 190 Kbytes. Thus, our SCGA code is only 11 Kbytes (6%) extra compared to the basic *hello_world* code.

Memory Usage

Apart from the code size of SCGA, the memory usage of our SCGA also has to be reported. Normally, Contiki use two types of memory in the process, data segment and stack segment. The global variable in C programming is kept in the data segment, while the local variable is stored in the stack segment. Thus every variables are declared carefully in order to use the memory efficiently. The data segment of SCGA is show in Table 3.2.

Table 3.2: Data Segment Memory

Variable	Function	size (bytes)
SLOT_INFO	store data for a slot	$(\frac{node}{4}+1).sizeof(uint8\_t)$
schedule	scheduling matrix	$node.sizeof(SLOT\_INFO)$
ischedule	initial scheduling matrix	$node.sizeof(SLOT\_INFO)$
stack	data structure: stack	$stack\_size.sizeof(uint8\_t)$
neighbor.info	information of neighbor node	$node.sizeof(uint32\_t)$

The size of data segment is able to calculate from Equation 3.33.

$$data\_seg = \frac{2.node.mnode + 4mnode + 16.node}{4} + sizeof(stack) + 2 \tag{3.33}$$

The size of stack segment consume memory with worst case that shown in Table 3.3 and Equation 3.34.

Table 3.3: Stack Segment Memory

Function	Variable	Description	size (bytes)
generate_schedule	i,j,k	loop variable	$3.sizeof(uint16\_t)$
	cmatrix	temporary buffer	$sizeof(uint32\_t)$
	filter	fileter variable	$sizeof(uint32\_t)$
	adj	flag for adjacency node	$sizeof(uint8\_t)$
	cmatrix2	temporary buffer	$sizeof(uint32\_t)$
	filter2	fileter variable	$sizeof(uint32\_t)$
	adj2	flag for adjacency node	$sizeof(uint8\_t)$
min_frame	ms	selected time slot	$mnode.sizeof(SLOT\_INFO)$
	s	combined time slot	$(\frac{mnode}{4}+1).sizeof(uint8\_t)$
	i,j,found	temporary buffer	$3.sizeof(uint16\_t)$
	S, R, B	temporary slot	$3*(\frac{mnode}{4}+1).sizeof(uint8\_t)$
max_slot	i,j	temporary buffer	$2.sizeof(uint16\_t)$
	gray	the number of gray node	$sizeof(uint8\_t)$
	s,b,bound	the number of black node	$3.sizeof(uint8\_t)$
	B, R, C	selected time slot	$3.sizeof(SLOT\_INFO)$

$$stack\_seg = \frac{mnode^2 + 9.mnode + 48}{4} + \frac{3.mnode + 36}{4} + 26 \tag{3.34}$$

The data segment and stack segment consume the memory from Equation 3.33 and 3.34, respectively. Finally, the total memory which used by SCGA is concluded in Equation 3.35.

$$MEM\_SIZE = \frac{mnode^2}{4} + \frac{mnode.(node+2)}{2} + 4.node + sizeof(stack) + 30 \tag{3.35}$$

where *mnode* is the maximal node in the network, while *node* is the number of node concerned with the algorithm, and $sizeof(stack)$ is size of allocated stack.

For example, the SCGA is implemented on Contiki. The maximum node is 32 nodes and size of stack is 256 slots. Therefore, the memory size is $20.node + 689$ bytes. If there are 32 sensor nodes in network, SCGA consume memory 1.3 kB only.

3.5 Summary

The chapter describes the packet scheduling algorithm named SCGA and proved that our algorithm can solve the packet collision. SCGA consists of three parts: packet collision free, frame length minimization, and throughput maximization. The time complexity of three parts are $O(N^3)$, $O(S_{apx}.ln(N))$, and $O(N.S_{apx})$, respectively. Our algorithm is able to transmute the maximum throughput. while it surprisingly minimize the delay under low complexity of algorithm. In the next chapter, our SCGA will be evaluated and compared with the existed works by using the mathematical method and network simulator method .

CHAPTER 4
PERFORMANCE EVALUATION

The Packet Scheduling based on Set Covering Problem by using Greedy Algorithm (SCGA), which has been introduced and described in the previous chapter. As it has been demonstrated in the previous chapter, the broadcast scheduling problem (BSP) is a set covering problem and it is able to be solved by greedy algorithm. The main reason for using greedy technique is because of its simplicity and incomplexity. Therefore, it is suitable for implementation in sensor nodes. In this chapter, we will show the SCGA evaluation by 3-03-3 using both mathematical method and network simulator. The performance in term of throughput, averaged delay and channel utilization are analyzed and discussed in this chapter. The Constant Bit Rate (CBR) and density of node are focused because of their affectations in packet collision

4.1 Mathematic Evaluation

Packet collision minimization is required in the broadcast scheduling problem (BSP). Three performance metrics such as, throughput, averaged delay, and channel utilization are necessary to evaluate when packet scheduling algorithm is applied. Thus these three metrics will be used to assess the SCGA and also compared with the other algorithms using the network benchmarks. Firstly, we will explain how to calculate three performance metrics. Then later, the comparison of the existed works is discussed.

4.1.1 Performance Metrics

Three performance metrics such as throughput, average delay, and channel utilization are expressed as the following:

- Throughput (σ: slots) It is the number of the reserved time slots or black slots, that are able to be assigned to any sensor nodes. The throughput is calculated using the Equation 4.1. The schedule matrix, S, is $|V|x|S|$. $|V|$ denotes the number of nodes, $|S|$ denotes the frame length and s_{ij} is the status of node in each time slot.

$$\sigma = \sum_{i=1}^{|V|} \sum_{j=1}^{|S|} s_{ij} \tag{4.1}$$

where

$$s_{ij} = \begin{cases} 1 & \text{node status is black} \\ 0 & otherwise. \end{cases}$$

- Channel Utilization (η :%) The throughput and average frame delay have been traded-off. More frames mean more available time. On the other hand, the averaged frame delay depends on the frame length. While the channel utilization is independent. Therefore, the channel utilization is the best metric to measure the performance of the algorithm. The channel utilization is calculated from the Equation 4.2.

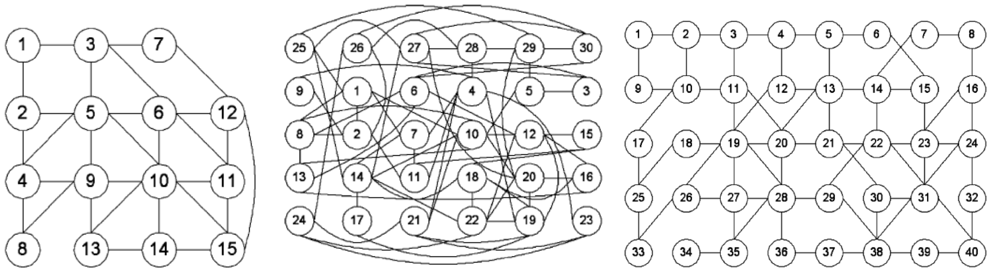
$$\eta = \frac{\sigma}{|V|x|S|}x100 \tag{4.2}$$

- Average delay (τ: frames) This indicates the waiting time of transmission. The average delay is calculated by Equation 4.3. This metric depends on the frame length and the number of black slots per node. If any algorithm can reduce the frame length with the same throughput, the average delay will be different. The distribution of black slots can determine the average delay. A high distribution gives a low average delay.

$$\tau = \frac{|S|}{|V|}\sum_{i=1}^{|V|}\left(\frac{1}{\sum_{j=1}^{|S|} s_{ij}}\right) \tag{4.3}$$

4.1.2 Comparison with The Existed Works

In order to make a standard comparison, the well-known network topology benchmarks have been introduced and used to compare between the SCGA and the existed works. These benchmarks have been introduced by Wang and et al [10] and usually used for the broadcast scheduling problem in wireless sensor networks and ad hoc networks. The benchmarks consist of three topologies called BM15, BM30, and BM40, based on 15, 30, and 40 nodes, respectively. These three network topologies are shown in Figure 4.1.

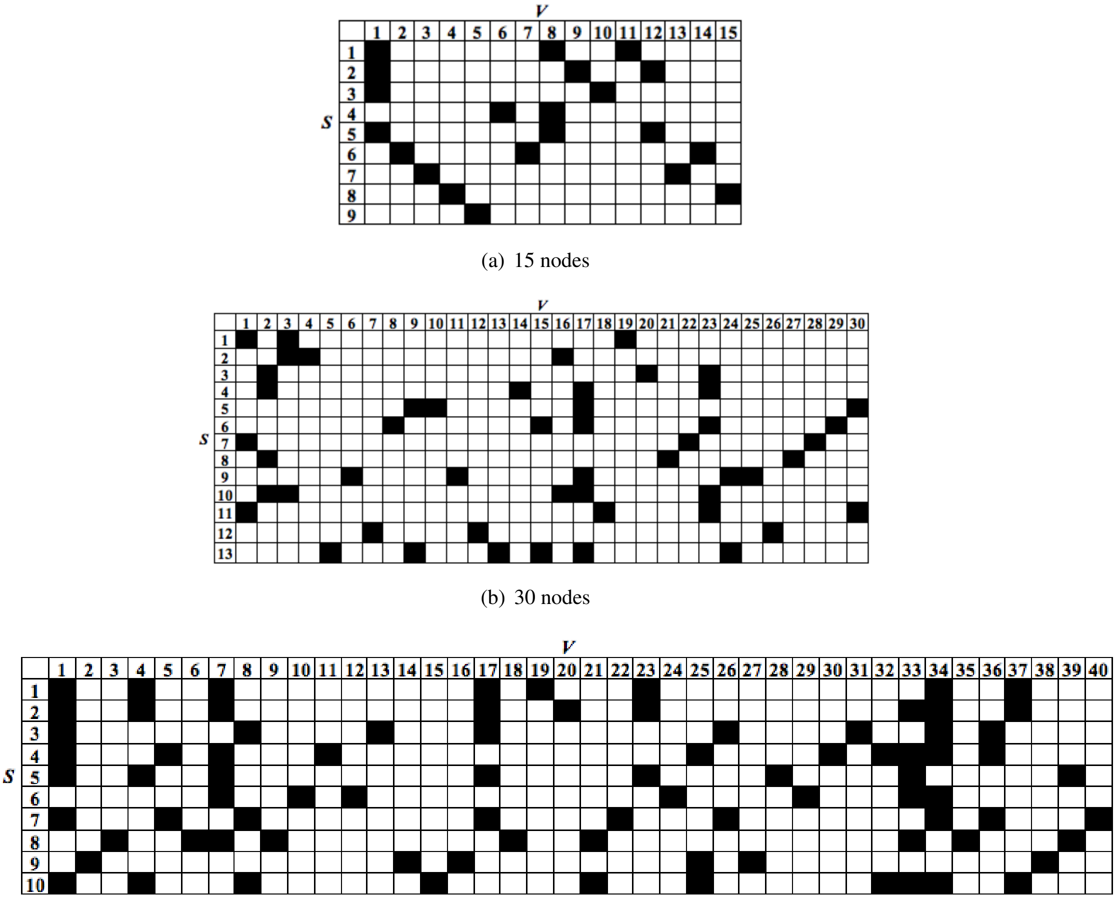


(a) BM15: 15 nodes (b) BM30: 30 nodes (c) BM40: 40 nodes

Figure 4.1: Network benchmarks

Table 4.1: Summary of benchmarks and results

	MinNB	Average *NB*	*MaxNB*	S_{opt}	S	σ	τ	η
BM15	2	3.87	7	8	9	21	7.85	0.1556
BM30	2	4.53	7	8	13	51	10.31	0.1308
BM40	2	3.20	7	8	10	84	7.31	0.2100



(a) 15 nodes

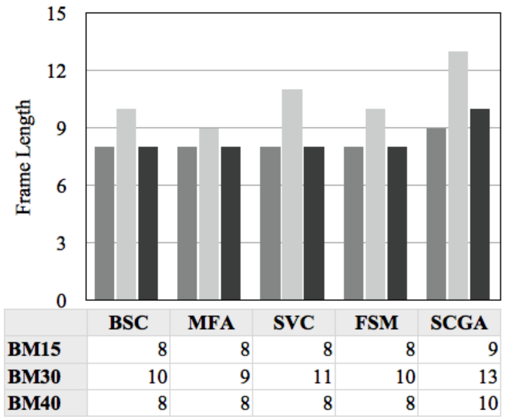
(b) 30 nodes

(c) 40 nodes

Figure 4.2: scheduling matrix

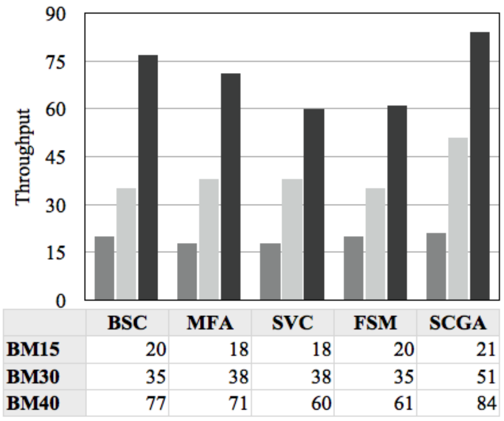
The scheduling matrixes of SCGA as shown in Figure 4.2 have been applied to the benchmarks. These three benchmarks have the optimal frame length at 8, the minimal neighbor node at 2 and the maximal neighbor node at 7. However, our SCGA gives the optimal frame length to 9, 13 and 10 when running on the three benchmarks, BM15, BM30 and BM40, respectively. The result is shown in Table 4.1. We found that the SCGA gives the best throughput when running on BM40 benchmark Although the BM40 has 40 nodes and 10 frames length, the channel utilization and average delay are better than the others. Each frame in Figure 4.2 consists of time slots filled with black or white color. Node j of frame i filled with black color means that node j sends a packet of frame i with no collision. For example, the first frame in Figure 4.2a is reserved for node 1, 8, and 11. Therefore, these nodes are granted the permission to send a packet in this frame while the other nodes are blocked.

After we have run our algorithm on three benchmarks, the other algorithms such as BSC [13] , MFA[10], SVC [11] and FSM[15] are also evaluated for the frame length comparison. The frame length comparison is shown in Figure 4.3(a). Our algorithm cannot achieve the optimal frame length and have the frame length greater than the others. The proposed algorithms minimizes the frame length to 9 frames, which is a optimal frame length of the benchmarks. However, the proposed algorithm, SCGA achieve the best throughput by taking the advantage of getting not optimal frame length. We can notice that the algorithm having the best optimal frame length will not gain the best throughput. On the other hand, we can conclude that the frame length is inversely proportional to the throughput but the frame length is directly proportional to the memory usage in a node of wireless sensor networks. Therefore, we have to trade off between the frame length and throughput.



	BSC	MFA	SVC	FSM	SCGA
BM15	8	8	8	8	9
BM30	10	9	11	10	13
BM40	8	8	8	8	10

(a) frame length

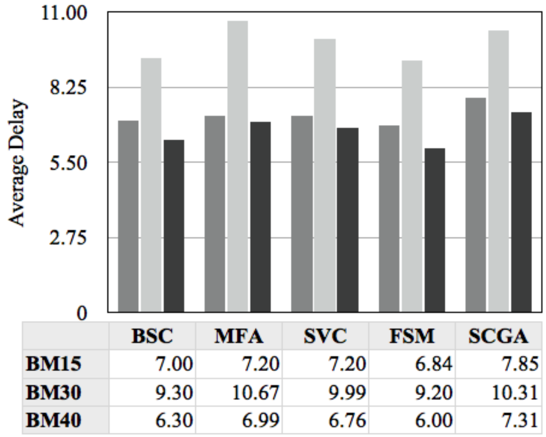


	BSC	MFA	SVC	FSM	SCGA
BM15	20	18	18	20	21
BM30	35	38	38	35	51
BM40	77	71	60	61	84

(b) throughput

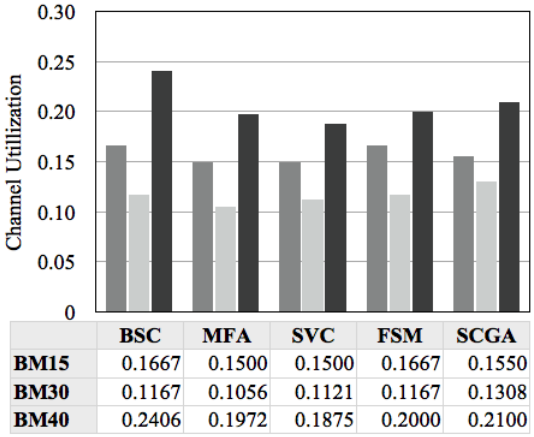
Figure 4.3: Performance comparison: frame length and throughput

The performances of all algorithms are shown in Figure 4.3 The performance calculated using the equation 4.1, 4.2 and 4.3 have been used to evaluate all algorithms. The SCGA gives the best throughput because our algorithm is not addressed to optimal the frame length as same as the others. However, with our frame length, the memory space of sensor node is enough for implementation as discussed in the complexity of algorithm in Chapter3. Therefore, they presented only the throughput or the average delay while the other approaches determined both the average delay and the throughput concurrently.



	BSC	MFA	SVC	FSM	SCGA
BM15	7.00	7.20	7.20	6.84	7.85
BM30	9.30	10.67	9.99	9.20	10.31
BM40	6.30	6.99	6.76	6.00	7.31

(a) average delay



	BSC	MFA	SVC	FSM	SCGA
BM15	0.1667	0.1500	0.1500	0.1667	0.1550
BM30	0.1167	0.1056	0.1121	0.1167	0.1308
BM40	0.2406	0.1972	0.1875	0.2000	0.2100

(b) channel utilization

Figure 4.4: Performance comparison: average delay and channel utilization

The average delay (τ) and channel utilization (η) of the existing algorithms calculated from the Equation 4.3 and 4.2 are shown in Figure 4.4. The average delay varies directly with the frame length. The channel utilization also varies directly with the throughput.

The average delays of BSC, MFA, SVC, FSM and SCGA are 6.30, 6.99, 6.76, 6.00 and 7.31, respectively. The average delay of each algorithm is not significantly different. Thus our algorithm used greater frame length than the others will not affect to the average delay too much. By the way, the SCGA gains a better throughput at about 10-30%. The channel utilization of SCGA gives the best result when running on BM30 benchmark and gives the second place when running on BM40 benchmark.

After we can get an average delay, frame length and channel utilization, we have listed the ranging of performance metrics for BSC, MFA, SVC, FSM and SCGA and shown in Table 4.2. Then we give the 3 marks for the first range and 1 mark for the second range. SCGA provide the best throughput whereas BSC has the high channel utilization. FSM has the lowest delay.

Because SCGA have the highest frame length for node allocation. Therefore, there are many space time slot for sensor node and it bring to be the highest score in throughput metric. SCGA take the frame length more than other algorithms that result in the channel utilization less than BSC. The separation of time slot effects average delay metric. As shown in Figure 4.2(a), node 1^{st} grants for 4 slots to send packet while node 2^{nd} grants only 1 slot. This situation bring to the high averaged delay because of Equation 4.3.

Table 4.2: The first and second algorithm ordered by performance metrics

	σ	η	τ
1^{st}	**SCGA**	BSC	FSM
2^{nd}	BSC	**SCGA**	BSC
3^{rd}	SVC, MFA, FSM	FSM	SVC, MFA,**SCGA**

The mathematical results are shown and analyzed in this section. We conclude that the SCGA generates the best result. After we have SCGA algorithm, we have applied SCGA in the wireless sensor networks and evaluated by network simulator (NS-2). The SCGA algorithm combined with CSMA/CA is compared the result with a simple original CSMA/CA in the next section.

4.2 Network Performance Evaluation

The implementation of SCGA on NS-2 is explained in the first part. The IEEE 802.15.4 standard combined with SCGA is evaluated and compared with the simple CSMA/CA algorithm in the second part. The number of packet collision and throughput are reported to demonstrate how efficient of our SCGA is. Meanwhile, the side effect of scheduling algorithm (average end-to-end delay) is also discussed.

4.2.1 Implementation of SCGA

SCGA has been implemented as a module of NS-2. The architecture of SCGA is shown in Figure 4.5. The architecture consists of three parts: neighbor discovery system, SCGA, and interfacing module.

Neighbor Discovery System

Neighbor discovery system or NDS is the system for discovering the node information. Sensor node has to know the information of neighbor node before starting SCGA in order to use ID of neighbor node for scheduling. This system consists of a neighbor discovery process (NDP) and a neighbor information table (NIT). SCGA will obtain the information of neighbor node form NIT. While, the NDP will be executed depending on a neighbor discovery algorithm used in each routing protocol. The output of NDP is the information of the neighbor node. However, the NDS for the experiments in this thesis is not implemented for real. The information of the neighbor node is taken from the network topology as shown in Figure 4.1. Thus, the information is ready to import to NS-2.

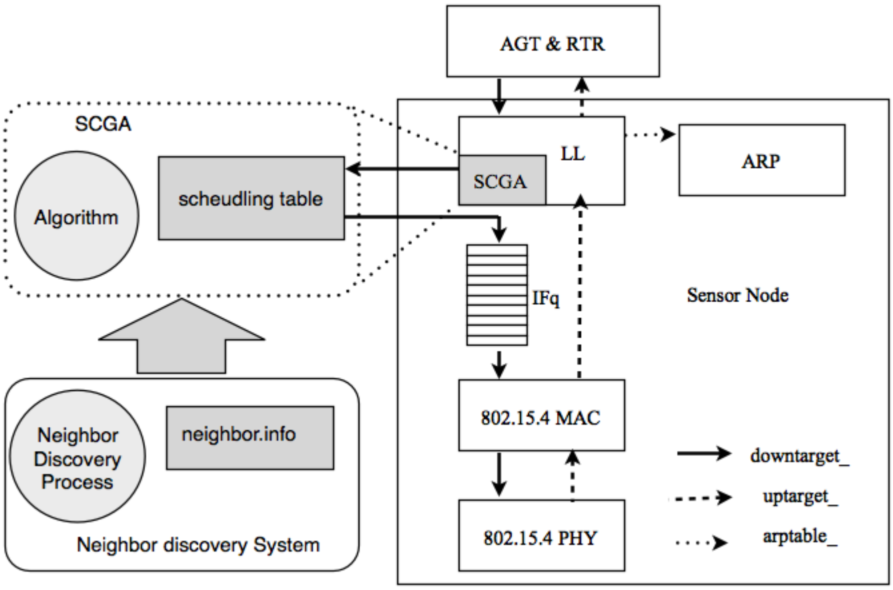


Figure 4.5: Architecture of SCGA

SGCA

SCGA is a core of this experiment. Firstly, SGCA gets the information of neighbor node from NDS module periodically or when the routing table is changed. When the network topology is changed or the neighbor discovery process changes the information in the table, the flag is set. This will inform the SCGA to perform scheduling in the next period. The algorithm consists of three phases as explained in chapter 3: packet collision free, frame length minimization, and throughput maximization. The result of SGCA is scheduling table.

Interfacing Model

Interface model is the interface between SCGA and LL layer in NS-2. When the packet is forwarded from the network layer to the interface queue, the packet is redirect to SCGA model to verify that the packet is granted to access the channel in this time slot. The time slot is marked in the packet and sent to the interface queue (IFq). By this technique, the packet is entered the queue via downtarget_ to IFq.

4.2.2 Scenarios and Parameters

The network simulation is the open source simulator and widely used in many network research works, especially in the network protocol development. The IEEE 802.15.4 MAC and PHY modules have been proposed by Zhen [30] and included to be a standard module in Network Simulator (NS-2). The details of the IEEE 802.15.4 standard are described in the RFC standard while the network simulation modules are explained in [30].

The most traffic behavior applications in wireless sensor network are periodic as Constant Bit Rate (CBR) in NS-2. Sometime, a base station may unicast the packet to control the sensor node. However, these packets are too small if they are compared to the packet from the sensor node. The User Datagram Protocol (UDP) is thus selected in the transport protocol in order to ignore all packet retransmission mechanisms in the transport layer when packets loss.

There are many routing protocols designed for wireless sensor networks. Unfortunately, we cannot find in NS-2. Wireless sensor networks is similar to ad hoc network but sensor nodes are more densely deployed, are prone to fail, and topology changes very frequently. Although sensor nodes are fixed, the network topology is frequently changed due to the transmit power of sensor nodes changed over time. Thus, we decide to deploy a famous ad hoc routing protocol named AODV in NS-2.

The sensor node communication in the IEEE 802.15.4 standard is divided into two categories: beacon and non-beacon mode. The data link layer and physical layer of the simulation rely on the non-beacon IEEE 802.15.4 standard that managed the packet collision with simple CSMA/CA. Because a beacon mode is a centralized paradigm, the PAN coordinator schedules a child node to handle the packet collision.

There are two interesting factors which will affect to the packet collision: CBR rate and density of node. The CBR rate indicates the number of packet forwarding from application layer whereas the density of node is the multiplication of control packet in MAC layer. When CBR packets are passed to the lower layer, the overhead such as routing packet and MAC packet are generated. The number of packet in the lower layer depends on a retransmission in the transport layer, route recovery in the network layer, or neighbor discovery in the data link layer. In order to evaluate the effects of traffic rate and the number of source node in application layer, the CBR rates are varied from 1, 6, 11, ... , 250 kbps that is the maximum bandwidth of IEEE 802.15.4 standard while the density of node is 15, 30 and 40 nodes corresponding to the network topology in the benchmarks. Each the density of node consists of 10 test cases with the random position of node. All sensor nodes send the CBR packet sized 40 bytes to a base station. The density of node increases the packet transmission rate that brings to increase the network traffic. The parameters are concluded in Table 4.3

Table 4.3: Parameters for NS-2

Parameter	**Default value**
Channel	WirelessChannel
Propagation	TwoRayGround
Network Interface	WirelessPhy/802.15.4
Antenna model	OmniAntenna
Transmission range	10 m
Sensor Field Size	40x40 m^2
Number of node	15, 30, and 40 nodes
Node Placement Method	Random
Mac Protocol	802.15.4
Link Layer Type	LL
Interface Queue Yype	Droptail/PriQueue
Interface Queue Length	450
Routing Protocol	AODV
Applications	CBR
CBR length	40 bytes
CBR rate	1,6,11,...,250 kbps

We observe three parameters: packet collision rate, average end-to-end delay and throughput. The packet collision rate is the important parameter to dominate how better of the purpose packet scheduling algorithm. The average end-to-end delay is discussed in term of a side-effect. In addition, the throughput is also reported. All experimental results will be explained in the next part.

4.2.3 Performance Improvement with SCGA

We improve the IEEE 802.15.4 standard by injecting our SCGA in LL layer before passing to MAC layer. The experiment results show that the number of dropped packet in the MAC layer is decreased. Although, the result does not give the significant improvement in a low density of node, SCGA can reduce the number of dropped packet clearly in a high density. The main reason of dropped packet in the MAC layer is the low link quality and packet collision causing no route in the network layer.

Although the packets are scheduled and the number of dropped packet is decreased in the MAC layer, the full interface queue is still caused the dropped packet in the network. Most of CBR packet is dropped by interface queue that resulting in a lower throughput. SCGA can cause a higher number of dropped packets from NRTE reason. Unfortunately, the NRTE cannot affect significantly to the network because the dropped packet from NRTE occurs only 10% of the overall packet drop. The problem of packet drop in the network layer comes from the full interface queue at 90% which is not able to solve with SCGA.

Packet Collision

Packet collision is a measure of the amount of data that is dropped because of the collided packet in MAC layer. All sensor nodes share a single channel using a multiple access protocols. An arbitrary node sends the signal to allocate the channel. If the channel is idle, it is granted to send the packet. On other hand, it resends the signal when the channel is not idle after the back-off time. The high node density has the higher possibility of collision than the low node density because sharing the same communication channel. This experiment is the simulation to compare the packet collision between the original CSMA/CA and SCGA. The CSMA/CA mechanism is a packet collision avoidance that implemented in the IEEE 802.15.4 standard. Our SCGA is built in LL layer to help to improve the CSMA/CA mechanism.

The figures in this section are performance comparison between the simple CSMA/CA and CSMA/CA with SCGA. The x-axis is CBR rate varied from 1 to 250 kbps and the y-axis is the number of packet shown in the thousand packets. In addition, there are three sub figures varied in the density of nodes, 2.34, 4.64, and 6.34. (explain how to get 2.34, 4.64 and 6.34). The density of node is the averaged neighbor node of 15, 30, and 40 nodes. There are two lines in each chart. The line with circle represents the simple CSMA/CA mechanism and the dark line represents the CSMA/CA combining with SCGA.

The results as shown in Figure 4.6 inform us that the packet collision directly varies to the density of node while the CBR rate does not affect to the packet collision. The packet collision is less than 3500 packets in the low density and increased up to 5000 packets for high node density.

This figure shows that SCGA can reduce packet collision clearly in the high dense network. For the spare network, SCGA can reduce packet insignificantly.

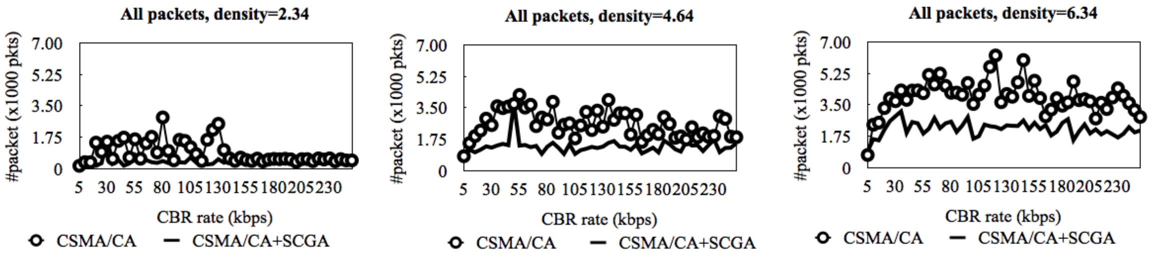


Figure 4.6: Packet collision

The number of overall packet collision has already shown in Figure 4.6. The overall packet collisions are able to be a packet collision in AODV packet or CBR packet. Thus, the experiment for classification is setup and the results are shown in Figure 4.7 and Figure 4.8. We found that the packet collision in AODV packet between the original CSMA/CA and the CSMA/CA with SCGA is not different. In contrast with CBR packet, the packet collision of the original CSMA/CA is higher than the CSMA/CA with SCGA when the density of node is 4.64 and 6.34. Therefore, our SCGA can perform efficiently and can reduce the packet collision in CBR packet better than AOVD packet. Even the AODV packets are scheduled, the packets are stacked in the interface queue and not processed. Because of this reason, the SCGA cannot solve the packet collision of AODV directly when the density of node is low.

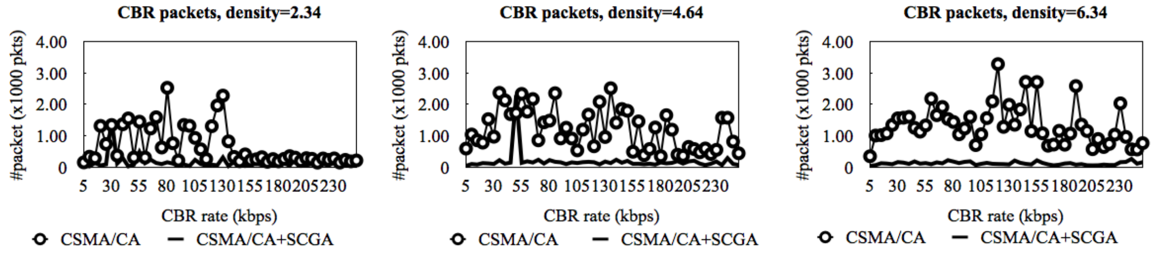


Figure 4.7: Packet collision: CBR packet

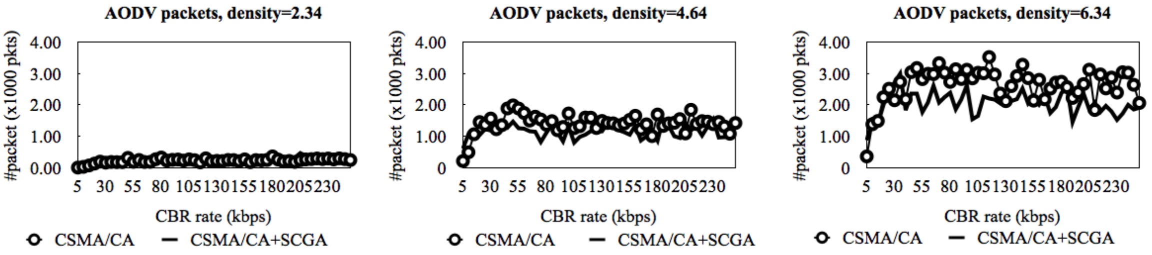


Figure 4.8: Packet collision: AODV packet

Figure 4.7 show packet collision of AODV packet. The packet collision is concerned with the number of packet in MAC layer. Therefore, packet collision of AODV less than CBR in the low dense network. On the other hand, packet collision of CBR more that AODV in the high dense network. Although AODV packets are scheduled by SCGA, their packets are in the interface queue and are not processed. AODV packets dropped because of IFQ is the evidence. Therefore, SCGA can not solve the packet collision of AODV directly. The fully interface queue have to be studied and analyzed. The performance of SCGA in CBR packet opposites the AODV. SCGA solves the packet collision of CBR packet directly as show in Figure 4.8 especially the high dense network.

We can conclude the experimental results that CSMA/CA improving with our SCGA is able to reduce the number of packet collision in MAC layer, especially the packet collision of CBR packet. Although, the CSMA/CA with SCGA cannot reduce the number of AODV packet collision significantly in a low node density, the AODV packet collision is decreased when the density of node is increasing. The CSMA/CA with SCGA can reduce the CBR packet collision significantly.

Averaged End-to-End Delay

The average end-to-end delay, called delay in this thesis, is the side affect of packet scheduling algorithms. The delay is required be aware because it is able to delay the overall networks. The data from the upper layer protocol are buffered in the FIFO queue to wait for the route. If the routing process takes the time longer than the transmission time of upper layer protocol, FIFO queue will be full and resulted in the dropped data. The delay refers to the time taken for a packet to be transmitted across a network from source to destination in application layer. Therefore, the delay is calculated when the transmission is only successful. The comparison of the delay between the simple CSMA/CA and the CSMA/CA with SCGA is shown in Figure 4.9.

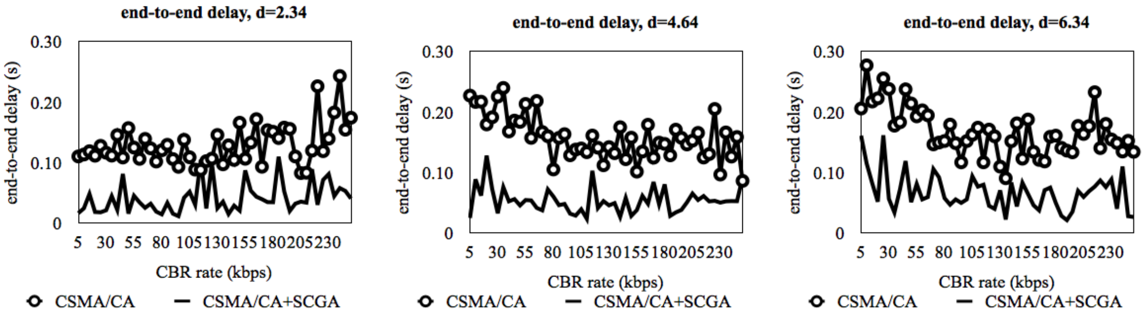


Figure 4.9: Averaged End-to-End Delay

The packets are dropped largely in the IEEE 802.15.4 standard. The dropped packet is directly proportional to the network performance. The dropped CBR packet will affect to the throughput while the dropped AODV packet will affect to the network latency. The delay of CSMA/CA with SCGA is significantly lower than the original CSMA/CA. However, the delay is not affected by the CBR rate as we can see from the graph in Figure 4.9 . Even the CBR rate is increased, the delay is not different. From this experiment, our SCGA does not affect to the delay.

Throughput

The main objective in wireless sensor networks is to collect the data from the sensor field and forward to the base station. The necessary performance of this kind of networks is throughput. Throughput is the amount of successful received packet in the time period. The experiment is setup to evaluate the throughput of SCGA. The CBR packets are dropped before sending in MAC layer more than 50%. Thus we calculate the success ratio from the ratio between CBR packet sent from MAC layer and CBR packet received by base station in this thesis. The results of the throughput and success ratio are shown in Figure 4.10 and Figure 4.11 , respectively.

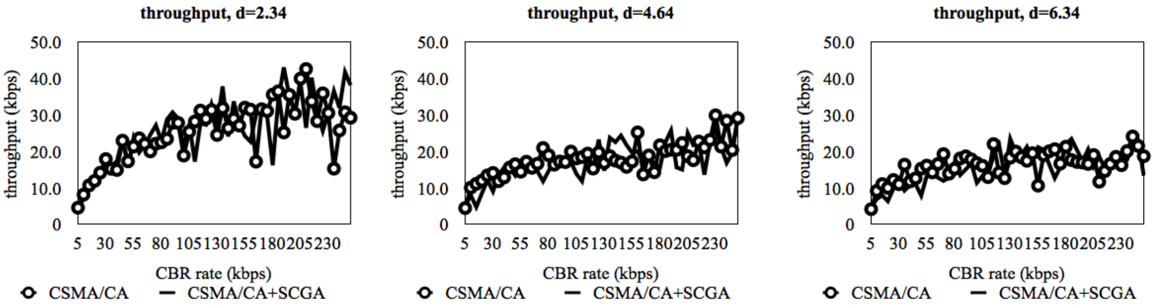


Figure 4.10: Throughput

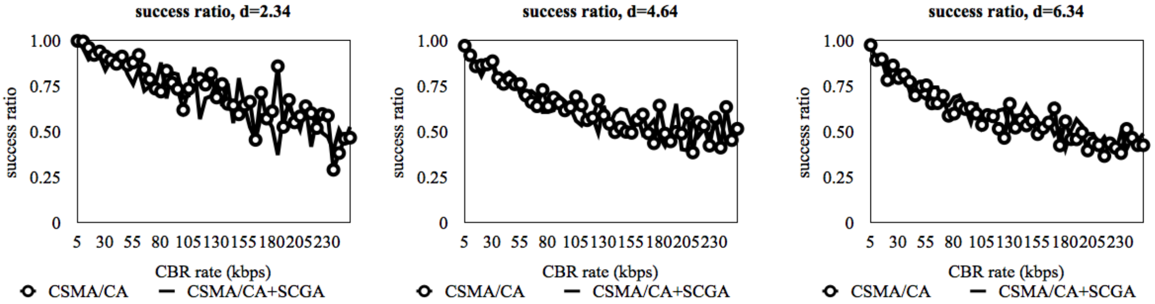


Figure 4.11: Success ratio

The CBR rate can affect to both the throughput and the success ratio. Throughput is increased as a logarithm function and converged to 40, 30, 20 kbps for the density of node at 2.34, 4.64, 6.34, respectively. Although the CBR rates are increased, the throughput is grown with the condition of the interface queue limitation. The success ratio is also grown as a logarithm function. However, the success ratio is not different when the density of node is increased. Even the throughput of the original CSMA/CA and the CSMA/CA with SCGA is similar, the CSMA/CA with SCGA can achieve better the average end-to-end delay and the packet collision than the normal CSMA/CA.

4.3 Summary

We measure the performance of the SCGA with the mathematical method in terms of the frame length, the throughput, the averaged delay, and the channel utilization. The comparison is also compared to the previous works using the network benchmarks. From the mathematical method, we can conclude that the SCGA produces the highest throughput and utilizes the channel better than other algorithms. However, the SCGA cannot gain a better averaged delay. Moreover, the network simulator (NS-2) is used to evaluate our SCGA when it is applied to wireless sensor networks using the same benchmarks. Although, the SCGA cannot achieve the better throughput and success ratio, the CSMA/CA with SCGA can obtain the better collided packet reduction and the delay compared to the CSMA/CA.

CHAPTER 5
CONCLUSIONS

Summary

This thesis has proposed the algorithm for packet scheduling in logical link layer named Set Covering problem with Greedy Algorithm (SCGA). SCG reduces packet collision and packet delay while throughput is increased, which will solve the packet scheduling problem. The broadcast scheduling problem is proved to be set covering problem, which can be solved by greedy algorithm. In addition, the broadcast scheduling problem is modeled with undirected graph and represented with matrix. The columns of matrix are sensor node, while the rows of matrix are time slots. More over, the members in matrix are represented with colored status: black, gray and white. The black status denoted the allocate slot for node. On the other hand, the white status denoted the unauthorized node for sending packet in this time slot. While gray nodes are in the matrix at the first step and then changed to be black or white status depends on frame length minimization phase and throughput maximization phase.

SCGA consists of three phases: packet collision free, frame length minimization and throughput maximization. Packet collision free phase transforms the network topology, that represented by graph theory, to be scheduling matrix under the packet collision free. Frame length minimization phase is the core of our algorithm that combines two validate time slots, in order to reduce number of rows in matrix. The combination of time slots has been proved to be set covering problem. We solved this problem with greedy method and defined the new operator to support greedy decision. While throughput maximization phase changes gray status in matrix with validate time slot, which rely on greedy algorithm.

We implemented the SCGA for Tmote Sky, then simulated on MSPsim in order to show that the proposed algorithm can be implemented and executed under resource constraint device. Moreover, we analyzed time complexity of SCGA and presented the time complexity with big-O notation. Packet collision free phase grows in $O(N^3)$ whereas frame length maximization phase grows in $O(S_{apx}.log(n))$. While throughput maximization phase grows in $O(S_{apx}.N)$.

The SCGA is evaluated by two methods: mathematical method and network simulation method. For the mathematical method, the SCGA are implemented in C programing language and compared with the previous works by network benchmarks. The benchmarks are usually used as the standard for all research in broadcast scheduling problem. The frame length and throughput are main metrics whereas channel utilization and average delay are by-product. The SCGA takes average delay more than previous works only 15.44%, 12.06% and 26.03% for 15, 30 and 40 nodes respectively because SCGA has the highest frame length. SCGA generated throughput better than previous works as result up to 5%, 10% and 34.21% for 15, 30 and 40 nodes. The previous works utilized the channel better than SCGA only 7.13% and 14.57% at 15 and 30 nodes benchmark. In reverse, SCGA utilizes channel better that FSM and BSC to 11.91% at 30 node benchmark.

For network simulation method, SCGA is implemented as extension module in network simulation programe named NS-2. The packet collision problem is solved by CSMA/CA mechanism in IEEE 802.15.4 standard. While the SCGA is the extension module in order to improve performance of CSMA/CA by scheduling packets before CSMA/CA mechanism is executed. We compare the SCGA with CSMA/CA under two domains: CBR rate and density of node in order to evaluate packet collision, end-to-end delay and throughput. SCGA reduces the number of packet in MAC layer result in decreasing of packet collision especially with CBR packet.

On the other hand, SCGA cannot reduce packet collision of AODV protocol directly. Because AODV packet has less amount than CBR packet and these packets have been dropped at interface queue. As result, the packets in interface queue have not been proceed within simulation time.

Importantly, the density of node is main reason of packet collision. The performance of SCGA is directly varied with density of node. Actually, the SCGA is not side effect in term of average end-to-end delay. On the other hand, it decreases end-to-end delay because packet dropped in MAC layer has decreased.

Throughput of SCGA is not different from CSMA/CA. Because the low link quality and fully interface queue cannot be solved by SCGA. By statistical method, they are the evidence to conclude that throughput from SCGA and throughput from CSMA/CA are not different significantly. From the simulation, we found that there are two factors to control throughput: interface queue and link quality. Therefore, SCGA should be improved in order to increase throughput by determine these two factors.

Future Works

The SCGA consists of three phases: packet collision fee, frame length minimization, and throughput maximization. The first and second phases already are proved and shown that SCGA can prevent packet collision under minimal frame length. However, the greedy choice in this phase determines the packet collision only. More over, the greedy choice in throughput maximization phase is simple and is not proved before implementation. Therefore, the SCGA should be improved greedy choice in both two phases.

The weighting function of greedy choice is very important. We determine the packet collision only in SCGA. The other problems do not discuss in this report is fully interface queue and link quality. The throughput is low because of these two factors. Therefore, the weighting function of greedy choice should determine fully interface queue problem and link quality.

We analyze the SCGA in the term of algorithm complexity and network simulation. Because the broadcast scheduling problem concerned with queuing theory. This algorithm should be analyzed with queuing theory. The transmission time, service time, and waiting time of all packet should be analyzed and discuss.

Bibliography

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102 –105, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey," *Computer Networks*, vol. 52, pp. 2292 – 2330, August 2008.

[3] A. Bachir, M. Dohler, T. Watteyne, and K. Leung, "MAC Essentials for Wireless Sensor Networks," *Communications Surveys Tutorials, IEEE*, vol. 12, no. 2, pp. 222 – 248, 2010.

[4] L. Zhaohua and G. Mingjun, "Survey on Network Lifetime Research for Wireless Sensor Networks," in *Broadband Network Multimedia Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference on*, pp. 899 – 902, October 2009.

[5] A. El-Hoiydi, "Spatial TDMA and CSMA with Preamble Sampling for Low Power Ad Hoc Wireless Sensor Networks," in *Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on*, pp. 685 – 692, 2002.

[6] W. Ye, J. Heidemann, and D. Estrin, "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Trans. Networks*, vol. 12, pp. 493–506, June 2004.

[7] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, (New York, NY, USA), pp. 171–180, ACM, 2003.

[8] "IEEE 802.15.4 Standard, Wireless Medium Access Controls (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Person Area Networks (WPANs)," September 2003.

[9] A. Ephremides and T. Truong, "Scheduling Broadcasts in Multihop Radio Networks," *Communications, IEEE Transactions on*, vol. 38, pp. 456 – 460, April 1990.

[10] G. Wang and N. Ansari, "Optimal Broadcast Scheduling in Packet Radio Networks using Mean Field Annealing," *Selected Areas in Communications, IEEE Journal on*, vol. 15, pp. 250 – 260, February 1997.

[11] J. Yeo, H. Lee, and S. Kim, "An Efficient Broadcast Scheduling Algorithm for TDMA Ad-Hoc Networks," *Computers and Operations Research*, vol. 29, no. 13, pp. 1793 – 1806, 2002.

[12] S. Salcedo-Sanz, C. Bousono-Calzon, and A. Figueiras-Vidal, "A Mixed Neural-Genetic Algorithm for the Broadcast Scheduling Problem," *Wireless Communications, IEEE Transactions on*, vol. 2, pp. 277 – 283, March 2003.

[13] H. Shi and L. Wang, "A Hybrid Neural Network for Optimal TDMA Transmission Scheduling in Packet Radio Networks," in *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, vol. 5, pp. 3210 – 3213, July 2005.

[14] I. Ahmad and A. S. Das, “A Heuristic Algorithm for the Minimization of Incompletely Specified Finite State Machines,” *Computers & Electrical Engineering*, vol. 27, no. 2, pp. 159 – 172, 2001.

[15] I. Ahmad, B. Al-Kazemi, and A. S. Das, “An Efficient Algorithm to Find Broadcast Schedule in Ad Hoc TDMA Networks,” *Journal of Computer System, Networks, and Communication*, vol. 2008, pp. 12:1–12:10, January 2008.

[16] V. Jelicic, T. Razov, D. Oletic, M. Kuri, and V. Bilas, “MasliNET: A Wireless Sensor Network based Environmental Monitoring System,” in *MIPRO, 2011 Proceedings of the 34th International Convention*, pp. 150 –155, May 2011.

[17] C. Yu, Y. Cui, L. Zhang, and S. Yang, “Zigbee Wireless Sensor Network in Environmental Monitoring Applications,” in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom ’09. 5th International Conference on*, pp. 1 – 5, September 2009.

[18] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, and D. Moore, “Environmental Wireless Sensor Networks,” *Proceedings of the IEEE*, vol. 98, pp. 1903 – 1917, November 2010.

[19] H. Cao, V. Leung, C. Chow, and H. Chan, “Enabling Technologies for Wireless Body Area Networks: A Survey and Outlook,” *Communications Magazine, IEEE*, vol. 47, pp. 84 – 93, December 2009.

[20] J. Caldeira, J. Rodrigues, and P. Lorenz, “Toward Ubiquitous Mobility Solutions for Body Sensor Networks on Healthcare,” *Communications Magazine, IEEE*, vol. 50, pp. 108 –115, may 2012.

[21] J. Corchado, J. Bajo, D. Tapia, and A. Abraham, “Using Heterogeneous Wireless Sensor Networks in a Telemonitoring System for Healthcare,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, pp. 234 – 240, March 2010.

[22] S. C. Ergen, “ZigBee/IEEE 802.15.4 Summary,” tech. rep., September 2004.

[23] “The network simulator: NS-2.” `http://http://www.isi.edu/nsnam/ns/`, Available on October 2012.

[24] P. Levis and N. Lee, “Tossim: A simulator for tinyos networks.” `http://www.cs.berkeley.edu/~pal/pubs/nido.pdf`, Available on October 2012.

[25] J. Elson, S. Bien, N. Busek, B. Bychkovskiy, A. Cerpa, D. Ganesan, L. Girod, B. Greenstein, T. Schoellhammer, S. Stathopoulos, and D. Estrin, “EmStar: An Environment for Developing Wireless Embedded Systems Software,” tech. rep., University of California, March 2003.

[26] “Omnet++ network simulation framework.” `http://www.omnetpp.org`, Available on October 2012.

[27] “J-Si’m Home Page.” `http://j-sim.cs.uiuc.edu`, Available on October 2012.

[28] “ATEMU- Sensor Network Emulator/Simulator/Debugger.” `http://www.hynet.umd.edu/research/atemu/`, Available on October 2012.

[29] "The AVR Simulation and Analysis Framework." `http://compilers.cs.ucla.edu/avrora`, Available on October 2012.

[30] J. Zheng and M. J. Lee, "A Comprehensive Performance Study of IEEE 802.15.4 Standard," *Sensor Network Operations*, pp. 218 – 237, 2006.

[31] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pp. 455 – 462, November 2004.

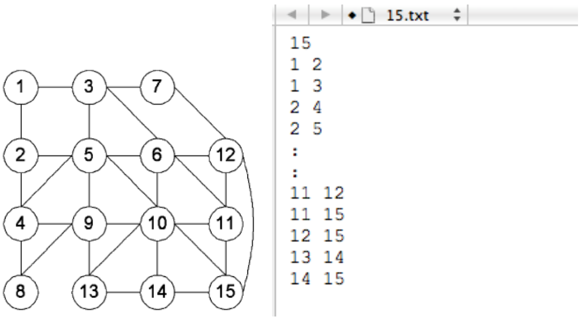
[32] "Tmote sky: Ultra Low Power IEEE 802.15.4 Compliant Wireless Sensor Module." `http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf`, Available on October 2012.

[33] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," in *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pp. 641 – 648, November 2006.

APPENDIX A
SOURCE CODE AND SCRIPTS

A.1 Compilation and Execution

Section 4.1.2 compares the exist works and the SCGA. Network topology is converted into pairs of nodes and stored in text file. For example, a benchmark in Figure A.1a consists of 15 nodes and 28 edges. The text file shown in Figure A.1b consists of number of node in the first line and pairs of nodes that concerned with the topology such as the number 15 in figure denotes 15 sensor nodes whereas *"1 2"* in text file denotes node 1 and 2 are adjacency.



```
15
1 2
1 3
2 4
2 5
:
:
11 12
11 15
12 15
13 14
14 15
```

(a) network topology (b) text file

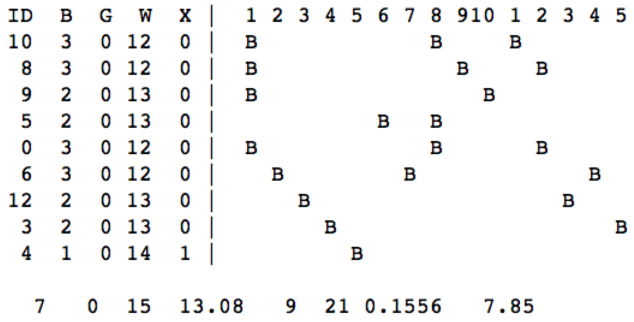
Figure A.1: Input file

All network topologies are stored in text file named, 15.txt, 30.txt, and 40.txt. The *scga.c* is compiled and executed as show below:

```
$ gcc scga.c -o scga -lm -DLD_TXT
$ ./scga 15.txt
```

The *-DLD_TXT* mean that the input of this program is text file. When the *scga* program is executed with argument, the results are generated and shown in Figure A.2:

There are two parts in Figure A.2. Firstly, the scheduling matrix consist of the number of black (B), gray (G), and white (W) node in each time slot. The *B* in each time slot represent that which node is allowed to send packet such as node 1, 8, and 11 are granted to send packet in the same slot. Secondly, the performance metrics are shown in last line of result. This line consist of maximum adjacent node, seed number, number of node, summary of delay, frame length, throughput, utilization, and average delay.



```
ID  B  G  W  X |  1 2 3 4 5 6 7 8 910 1 2 3 4 5
10  3  0 12  0 |  B             B     B
 8  3  0 12  0 |  B               B     B
 9  2  0 13  0 |  B                 B
 5  2  0 13  0 |            B   B
 0  3  0 12  0 |  B             B       B
 6  3  0 12  0 |    B         B             B
12  2  0 13  0 |      B                   B
 3  2  0 13  0 |        B                     B
 4  1  0 14  1 |          B

  7   0  15  13.08   9  21 0.1556   7.85
```

Figure A.2: Result of *psa* program

A.2 Source Code : scga.c

```c
/*
 * The directive and function prototypes are cut
 */

#define MAX_NODE        255
#define STACK_SIZE      200
#define TX_RANGE        10.0
#define WHITE           0x0
#define GRAY            0x2
#define BLACK           0x3
#define DEBUG

#ifndef DIST
#define DIST            10
#endif

int STACK[STACK_SIZE];
#define isFull()        (STACK[0]==(STACK_SIZE-1))
#define isEmpty()       (STACK[0]==0)
#define push(n)         STACK[++STACK[0]]=n;
#define pop()           (STACK[STACK[0]--])
#define SORT()          quicksort()

#ifdef CONN
typedef struct{
        int x;
        int y;
} NB;

NB pair[300];
int np;
#endif

typedef struct {
        char id;
        float x;
        float y;
}NODE;

typedef struct {
        char id;
        int black;
        int gray;
        int white;
        char status[MAX_NODE];
        int     select;
}SLOT_INFO;

NODE node_list[MAX_NODE];
SLOT_INFO schedule[MAX_NODE];
SLOT_INFO ischedule[MAX_NODE];
SLOT_INFO ms[MAX_NODE];


int beacon = 0;
int seed=0;
int XX, YY;

int main(int argc, char **argv){
#ifndef CONN
        int x,y;
        if(argc!=5){
                fprintf(stderr,"usage gen_topo  #n #x #y seed\n");
                exit(1);
        }

        N_NODE = atoi(argv[1]);
        if (N_NODE > MAX_NODE) {
                fprintf(stderr, " number of node is more than %d\n", MAX_NODE);
                exit(2);
        }

        XX = x = atoi(argv[2]);
        YY = y = atoi(argv[3]);
        seed = atoi(argv[4]);

        generate_topo(x, y);

#else
        if(argc!=2){
                fprintf(stderr,"usage gen_topo  filename\n");
                exit(1);
        }
        load_conn(argv[1]);
#endif

        generate_schedule();
#ifdef DEBUG
        print_schedule(schedule);
        printf("\n");
#endif

{
        FRAME_LEN=0;
        while (1) {
                SLOT_INFO s=greedy_decsion();
                if (s.black + s.white + s.gray ==0) {
                        break;
                }
                ms[FRAME_LEN] = s;
                FRAME_LEN++;
        }
        memcpy(&schedule, &ms, sizeof(SLOT_INFO)*FRAME_LEN);

#ifdef DEBUG
        print_schedule(schedule);
        printf("\n");
#endif
}
{
                int i,j;
                bubble_sort(FRAME_LEN);
                for (i=0; i < N_NODE; i++) {
                        for (j=0; j<FRAME_LEN; j++) {
                                if (schedule[j].gray == 0) continue;
                                if (schedule[j].status[i] == GRAY) {
                                        SLOT_INFO B = ischedule[i];
                                        SLOT_INFO R = combine(schedule[j], B);
                                        int bound = (B.black > schedule[j].black)?B.black:schedule[j].black;
                                        if (R.black <= bound) continue;
                                        schedule[j] = R;
                                }
                        }
                }
#ifdef DEBUG
        print_schedule(schedule);
        printf("\n");
#endif
}
        statistic();
        return 0;
}

SLOT_INFO greedy_decsion(){
        int i=0,j=0;
        int found = 0;
        SLOT_INFO S, R;
        int unselect = 0;
        bubble_sort(N_NODE);

        for (i=0; i<N_NODE; i++) {
                if (schedule[i].select == 0) {
                        unselect++;
                }
        }
        while (1) {
                while (j < N_NODE) {
                        if (schedule[j].select == 0)  break;
                        j++;
                }
                schedule[j].select = 1;
                S = schedule[j];
                i = j+1;
                while (1) {
                        SLOT_INFO B;
                        int bound;
                        while (i < N_NODE && !found ) {
                                if (schedule[i].select == 0) {
                                        found = 1;
                                        break;
                                }
                                i++;
                        }
                        if (!found) break;
                        B = schedule[i];
                        R = combine(S, B);
                        bound = (B.black > S.black)?B.black:S.black;
                        if (R.black <= bound){
                                i++;
                                found = 0;
                                continue;
                        }
                        schedule[i].select = 1;
                        S = R;
                }
                if (S.black >= (unselect/max_adj))  return S;
        }
}

void generate_schedule(){
        int i,j, k;
        for (i=0; i<N_NODE; i++) {
                schedule[i].id = i;
                for (j=0; j<N_NODE; j++) setStatus(&schedule[i], j, GRAY);
                setStatus(&schedule[i], i, BLACK);
                for (j=0; j<N_NODE; j++) {
                        float dis;
                        if (i==j) continue;
                        dis = find_distance(i, j);
                        if (dis > TX_RANGE) continue;
                        dense++;
                        push(j);
                        setStatus(&schedule[i], j, WHITE);
                }
                if (STACK[0] > max_adj) max_adj = STACK[0];
                while (STACK[0]!=0) {
                        k=pop();
                        for (j=0; j<N_NODE; j++) {
                                float dis;
                                if (k==j) continue;
                                if (j==i) continue;
                                dis = find_distance(k, j);
                                if (dis > TX_RANGE || schedule[i].status[j]==WHITE) continue;
                                setStatus(&schedule[i], j, WHITE);
                        }
                }
        }
        memcpy(&ischedule, &schedule, sizeof(SLOT_INFO)*N_NODE);
        FRAME_LEN = N_NODE;
}

SLOT_INFO combine(SLOT_INFO A, SLOT_INFO B){
        SLOT_INFO R;
        int i;
        for (i=0; i < N_NODE; i++) {
                R.status[i] = cb(A.status[i], B.status[i]);
        }
        R.id=A.id;
        R.white=0;
        R.gray=0;
        R.black=0;
        R.select=0;
        for (i=0; i < N_NODE; i++) {
                char c = R.status[i];
                if (c == WHITE) {
                        R.white++;
                }else if (c == GRAY ) {
                        R.gray++;
                }else {
                        R.black++;
                }
        }
        return R;
}
void setStatus(SLOT_INFO *s, int node, char color){
        int i;
        s->status[node] = color;
        s->white=0;
        s->gray=0;
        s->black=0;
        for (i=0; i < N_NODE; i++) {
                char c = s->status[i];
                if (c == WHITE) {
                        s->white++;
                }else if (c == GRAY ) {
                        s->gray++;
                }else {
                        s->black++;
                }
        }
}

void statistic(){
        int i, j;
        int sum = 0;
        float delay = 0.0;
        float s_delay = 0.0;
        SLOT_INFO thru;

        for (i=0; i<N_NODE; i++) {
                thru.status[i] = 0;
        }
        for (i=0; i<FRAME_LEN; i++) {
                for(j=0;j<N_NODE;j++){
                        if (schedule[i].status[j] == BLACK) {
                                thru.status[j]++;
                        }
                }
        }
        for (i=0; i<N_NODE; i++) {
                sum+=thru.status[i];
                s_delay += 1.0/thru.status[i];
        }
        delay = 1.0 * FRAME_LEN*s_delay/N_NODE;
        printf("%3d ", max_adj);
        printf("%3d ", seed);
        printf("%3d ", N_NODE);
        printf("%6.2f ", s_delay);
        printf("%3d ", FRAME_LEN);
        printf("%3d ", sum);
        printf("%6.4f ", 1.0 * sum/(FRAME_LEN*N_NODE));
        printf("%6.2f ", delay);
        printf("\n");
}

/* Supporting functions that do not concern with the algorithm are cut.*/
```

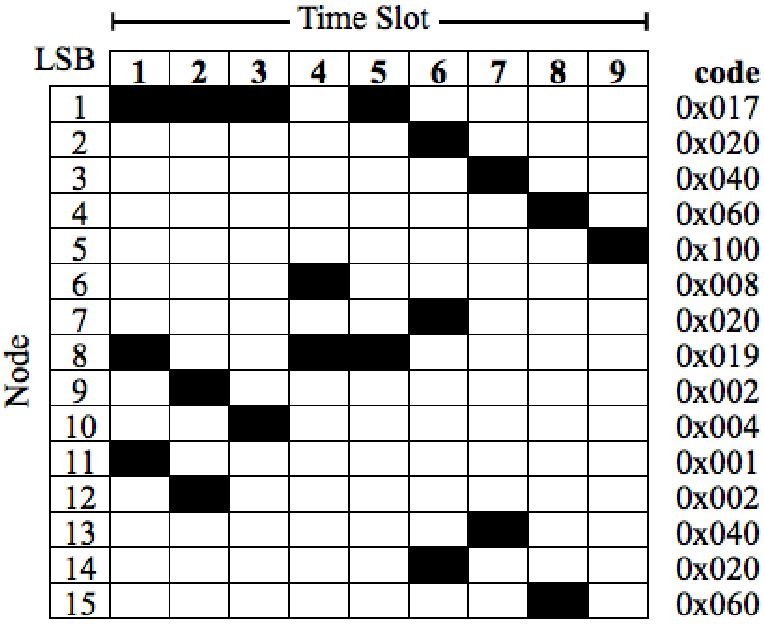
APPENDIX B
NETWORK SIMULATION WITH NS-2

B.1 NS2 modification

We have to modify NS2 for three parts: 1) binding C and TCL parameters, 2) implementing the SCGA , and 3) logging simulation results.

1) binding C and TCL parameters

The packet scheduling matrix from SCGA program are encoded to scheduling code that be used in NS2. For example, node 1 in Figure B.1 can send packet in time slot 1, 2, 3, and 5 whereas frame length is 9 frames. Therefore, the scheduling code of node 1 is 0x017.



LSB	1	2	3	4	5	6	7	8	9	code
1	■	■	■		■					0x017
2						■				0x020
3							■			0x040
4								■		0x060
5									■	0x100
6				■						0x008
7						■				0x020
8	■			■	■					0x019
9		■								0x002
10			■							0x004
11	■									0x001
12		■								0x002
13							■			0x040
14						■				0x020
15								■		0x060

Figure B.1: Scheduling Code

We need to modify mobile node module in NS2 in order to add three parameters: frame length, scheduling code, and time interval. Three parameters are assigned the value by the PSA program and passed them via simulation script.

The first modified file is *NS2_ROOT/common/mobilenode.h*. Three parameters are add in class *MobileNode* as protected member. Moreover, the inline function of parameters are implemented in order to return protected value.

Filename : NS2_ROOT/common/mobilenode.h

```
class MobileNode : public Node {
        friend class PositionHandler;
public:
/* cut */
// Line 148

#ifdef TOY
        inline int SchCode() { return SchCode_; }
        inline int FrameLen() { return FrameLen_; }
        inline double dT() { return dT_; }
#endif

protected:
/* cut */
// Line 187

#ifdef TOY
        int SchCode_;
        int FrameLen_;
        double dT_;
#endif
}
```

The second file is *NS2_ROOT/common/mobilenode.cc*. The parameters are initiated value and assigned with variables from simulation script in *MobileNode* :: *MobileNode*(*void*) line 124 and 148, respectively.

Filename : NS2_ROOT/common/mobilenode.cc

```
/* cut */
MobileNode::MobileNode(void) : pos_handle_(this) {
        X_ = Y_ = Z_ = speed_ = 0.0;
        dX_ = dY_ = dZ_ = 0.0;
        destX_ = destY_ = 0.0;
#ifdef TOY
// line 124
        SchCode_ = 0;
        FrameLen_ = 0;
        dT_ = 0.05;
#endif
/* cut */
// Line 148
#ifdef TOY
        bind("SchCode_", &SchCode_);
        bind("FrameLen_", &FrameLen_);
        bind("dT_", &dT_);
#endif
/* cut */
}
```

The final file for linking parameter between C and TCL file is modification of *tcl/lib/ns − default.tcl*. This file is TCL file that initiate the all parameter.

```
Node/MobileNode set SchCode_ 0
Node/MobileNode set FrameLen_ 1
Node/MobileNode set dT_ 0.05
```

2) Implementing the SCGA

All packets are send from upper layer to MAC layer via $LL :: sendDown()$ in $NS2\_ROOT/mac/ll.cc$. The direction of packet sending is specified in $ch- > direction$. The packet is send down when the $ch- > direction$ equal $DOWN$. At this time, all packet have to be scheduled with PSA via scheduling code. The parameters are initiated from simulation script via the parameters: $SchCode\_$, $FrameLen\_$, and $dT\_$ and assigned to $time\_slot$, n_time_slot, and gr, respectively. All nodes find the next turn of its packet transmission from $SchCode$. Node i can send packet in next turn if only of bit i^{th} of $SchCode$ is set.

3) Logging simulation results

Normally, there are no packet collision tracing for WPAN module in network simulation. However, we can modify WPAN source code in order to show packet collision events and their information such as time, packet type, sensor node etc. The packet collision be detected in physical layer and and drop packet at destination. The modification is shown below:

Filename : NS2_ROOT/wpan/p802_15_4phy.cc

```
void Phy802_15_4::recv(Packet *p, Handler *h){
/* cut */
// Line 609
        wph->colFlag = false;
        if (rxPkt == 0){
                rxPkt = p;
                HDR_LRWPAN(rxPkt)->rxTotPower =
                        rxTotPower[wph->phyCurrentChannel];
        }else{
#ifdef TOY
        fprintf(stdout, "D  %f COL %d %d %s %d %s %d\n",
                CURRENT_TIME,index_, p802_15_4macSA(rxPkt),
                wpan_pName(rxPkt), p802_15_4macSA(p), wpan_pName(p),
                HDR_CMN(rxPkt)->size()+ch->size());
#endif
        wph->colFlag = true;//collision flag is on
/* cut */
}
```

B.2 Network Simulation Process

The simulation generates two trace files: original file and modification from previous section. The original trace file is too big and there are waste disk space. Therefore, we decrease size of file by reducing unused parameters of trace file from network simulation. After that, we concat two file together and process the performance that describe in the next section.

In the network simulation process, we design the scenarios and specific the simulation parameter such as sensor field size, number node, and CBR rate etc. After that the simulation program is started in order to generate log file. The details and programs are listed below:

```
$ gcc scga.c -o /tmp/scga -lm
```

Firstly, the *psa* program is compiled to be binary file. The source code of this file is already explained in previous section.

```
$ /tmp/scga 64 40 40  1.0 0.000100  0 1 >> log-0-0
```

Secondly, we generate network topology and scheduling table by *scga* program. The parameters of *scga* program is composed of number of node, network size, CBR rate in kbps, frame interval in second, sequence of simulation and seed number. If seed number is zero, the topology is regenerated for all simulation time. This program generates two files: topology and schedule. Both files are input of network simulation.

```
$ ns sim.tcl -nn 64 -x 40 -y 40 -topo topo-0-0-0  -app cbr  -rp AODV
-sch std-0-0-0 > trace.tr
```

Thirdly, the scenario is simulated with *ns* program. The parameters consists of the simulation script called *sim.tcl*, number of node (*nn*), sensor field size (*x* and *y*), network topology (*topo*), application (*app*), routing protocol (*rp*), and scheduling file (*sch*). The network simulation generates two log file trace.raw and trace.tr. *trace.raw* is result from network simulation while *trace.tr* comes from modification in last section.

```
$ sh filter.sh  trace.raw > std-1.000-0.fil
$ grep ^D  trace.tr  >> std-1.000-0.fil
```

Finally, we reduce log file size in order to save storage and prepare log file. The final result is save in log file with extension *.fil*.

APPENDIX C
INTERNATIONAL JOURNALS AND CONFERENCES

List of Publication and Proceeding

- Chaiyut Jandaeng, Wannarat Suntiamorntut, and Nittida Elz, PSA : Packet Scheduling Algorithm for Wireless Sensor Networks, International Journal on applications of graph theory in wireless ad hoc networks and sensor networks, vol. 3, issue 3, pp. 1-12, Sep, 2011.
- Chaiyut Jandaeng, Wannarat Suntiamorntut, and Nittida Elz, Performance Prediction Model of Packet Scheduling Algorithm in Wireless Sensor Networks, International Journal of Wireless & Mobile Networks, vol. 3, issue 4, pp. 113-126, Aug, 2011.
- C. Jandaeng, W. Suntiamorntut, and N. Elz. Waste Energy Reduction in Wireless Sensor Networks with Packet Scheduling Algorithm, Embedded System and Intelligent Technology, 2011. ICESIT 10. International Conference on, Feb. 2011., Phuket, Thailand.
- C. Jandaeng, W. Suntiamontut and N. Elz, Packet Scheduling to Minimize Packet Collision and Maximize Throughput for Unslotted IEEE 802.15.4 Networks, The 2010 International Conference on Intelligent Network and Computing, ICINC, pp. 243-247, Nov., 2010., Kuala Lamper, Malaysia.
- C. Jandaeng, W. Suntiamorntut,and N. Elz. Throughput Improvement of Collision Avoidance Architecture in Wireless Sensor Networks, Wireless Communications, Networking and Mobile Computing, 2010. WiCom 10. 6th International Conference on, pages 1-5, Sep. 2010., Chengdu, China.

International journal on applications of graph theory in wireless ad hoc networks and sensor networks
(GRAPH-HOC) Vol.3, No.3, September 2011

PSA: The Packet Scheduling Algorithm for Wireless Sensor Networks

[1]C. Jandaeng,[1]W. Suntiamontut, [2]N. Elz

[1]Centre of Excellence in Wireless Sensor Networks,
Department of Computer Engineering, Prince of Songkla University
cjundang@gmail.com

[2]Computer System and Network Laboratory,
Department of Computer Science, Prince of Songkla University,
Hat Yai, Thailand 90112

Abstract

The main cause of wasted energy consumption in wireless sensor networks is packet collision. The packet scheduling algorithm is therefore introduced to solve this problem. Some packet scheduling algorithms can also influence and delay the data transmitting in the real-time wireless sensor networks. This paper presents the packet scheduling algorithm (PSA) in order to reduce the packet congestion in MAC layer leading to reduce the overall of packet collision in the system The PSA is compared with the simple CSMA/CA and other approaches using network topology benchmarks in mathematical method. The performances of our PSA are better than the standard (CSMA/CA). The PSA produces better throughput than other algorithms. On other hand, the average delay of PSA is higher than previous works. However, the PSA utilizes the channel better than all algorithms.

Keywords

packet collision, packet scheduling algorithm, wireless sensor networks

1. INTRODUCTION

A wireless sensor network is a self-configured network containing numerous small sensor nodes. Each node consists of sensing modules, a processing unit, radio frequency components and power sources [1]. They organize and communicate among themselves in an ad-hoc fashion. The wireless sensor network technology has been deployed in several applications such as health care monitoring systems, home automation and environment monitoring systems [2]. These applications require inexpensive facilities and little manual maintenance. According to the application requirements, each node has been implemented using a low-power microcontroller and radio module. In addition, each node is supplied with a small battery. Energy usage is the indicator of network lifetime [3].

All sensor nodes share a single communications channel using a multiple access protocol. The packet transmission may lead to a time overlap of two or more packet receptions, called collisions. The packet collision problem causes packet loss, packet retransmission, decreasing throughput, increased delay/latency and increased wasted energy consumption. Many research works on the MAC protocol have been proposed to solve the packet collision problem [4] such as Spatial TDMA traffic-adaptive medium access protocol (TRAMA) [5], Sensor MAC (SMAC) [6], and Timeout MAC (TMAC) [7]. A MAC protocol based on IEEE 802.15.4 was developed

DOI : 10.5121/jgraphoc.2011.3301

for low-power communication. The IEEE 802.15.4 MAC protocol uses a random back off in order to reserve and access the channel. A node is authorized to send the packet when the channel is idle. In contrast, random back off is activated when the channel is busy. Unfortunately, this technique will not work properly when used in a large scale wireless sensor network.

Time Division Multiple Access (TDMA) is a solution to reduce the packet collision problem. Total transmission time is divided into frames and each frame is divided into time slots. After that each time slot will be assigned to a sensor node to guarantee that every node is granted permission to send a packet in its time slot guaranteeing collision avoidance. Latency directly varies with frame length. On other hand, throughput inversely varies with frame length. There have been many approaches presented to minimize the frame length and maximize the throughput which are explained in section 2.

All previous works illustrated above are proposed for an ad hoc network. All devices are powerful nodes having unlimited energy. In contrast, sensor nodes are resource constrained having limited energy and low processing power. Therefore, the characteristics of the scheduling algorithm for a sensor network should be simplicity and efficiency. This paper therefore proposes a new algorithm based on the greedy technique that is simple and easy to implement in resource constrained devices. This paper will explain the proposed PSA and describe the evaluated results of the performance using mathematical results.

The remainder of the paper is organized as follows. First we briefly explain the packet collision problem and previous works in section 2. After that, the packet scheduling algorithm is described in detail in section 3. The performance comparisons using the mathematical results are presented in section 4. Finally we give the conclusion about the performance of the proposed packet scheduling algorithm in section 5.

2. PREVIOUS WORKS

Y. Peng *et al* [8] presented the TDMA with a scheduling matrix. The row of the matrix denotes frame length while the column of the matrix denotes nodes. The members of the matrix represent transmission authorization. In [8], they proposed to optimize the number of rows that refers to the frame length with Tabu search and greedy algorithm. This approach can reduce the average latency and produce high throughput in a dense area.

G. Wang and N. Ansari [9] have proved that the scheduling matrix optimization is an NP-complete problem. They also proposed an approximation method, mean field anneal (MFA) to optimize the schedule matrix. The matrix optimization is divided into two phases: minimize frame length and maximize throughput. More recently approximation methods have been proposed. S. Salcedo-Sanz *et al* [10] minimized frame length with a neural network (NN) and maximized throughput with a genetic algorithm (GA), whereas J. Yeo *et al* [11] applied the sequence vertex coloring (SVC) in both phases. S. Haixiang and W. Lipo [12] proposed a hybrid algorithm which combined back tracking sequential coloring (BSC) and noisy chaotic neural network (NCNN) to optimize the scheduling matrix. BSC-NCNN gives the minimal average time delay, while the NN-GA provides higher throughput.

I. Ahmad *et al* [13, 14]. proposed an idea to avoid packet collision. The network topology is represented by a finite state machine (FSM). The set of nodes are grouped with the maximal compatibles and incompatibles concept. This method begins by setting up a number of groups that equals the number of nodes. After that, combine groups together under the condition that no nodes in the same group are neighbor nodes. Finally, all sensor nodes are grouped in many groups and they can send packet in the same time without collision. The number of groups is frame delay while the summation of number of node in all groups is throughput. This idea leads to minimize latency and maximize throughput.

3. THE PACKET SCHEDULING ALGORITHM

The Packet Scheduling Algorithm (PSA) is the algorithm that schedules all packets from application layer and network layer in order to reduce network congestion in the data link layer to avoid the packet collision. When the PSA is implemented, packet collisions will be minimized with increasing of throughput as a by product. A PSA based on a greedy algorithm is a simple algorithm and easily implemented in a sensor node. The basic assumptions of the PSA are defined as follows. All packets communicate via IEEE 802.15.4 standard [15] that avoids packet collision with a simple CSMA/CA mechanism. All sensor nodes must know the information of at least 2-hops neighbor nodes. Finally, time synchronization, neighbor discovery, and routing protocols are not considered in this work.

3.1 Definitions

The node color represents *node status*. Two functions, *combine*() and *match*() are used to reduce the frame length. The node statuses and their functions are defined below:

Definition 1: *Node status* is represented by a node color for each time slot. A black node can send any packet with a guarantee of no collision. If a white node requests to send a packet, its packet may collide. A gray node is in the initial status with no guarantee with regard to packet collision. Finally, a gray node can change status to the other colors with *combine*() and *match*() functions. Figure 1 shows an example of 15 nodes status. The color of each node is set corresponding to its status which could be either black or gray (with "x") or white.

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		x		x			■	x	x	x			x	x	

Figure 1: Node status in PSA algorithm

Definition 2: The function *combine*() is used to reduce the frame length by combining two frames. The two frames must be tested with the *match*() function before the combination. The will be merged if the *match()* function returns valid. In the combination process, the status of a node can be changed to another color as defined below. Let *A, B* and *R* denote the frame and A_i, B_i, and R_i are node status in the i^{th} time slot of *A*, *B*, and *R*; *V* denotes the set of nodes and *R=combine(A,B).*

$$\forall i_{i \in V}, R_i = \begin{cases} A_i & \text{if } B_i \text{ is GRAY} \\ B_i & \text{otherwise.} \end{cases}$$

The above equation also means that:

2.1) the node status of frame *R* can be replaced with the status of A_i if B_i is a gray node.

2.2) Otherwise, it will be replaced with B_i.

2.3) *combine*(*A*, *B*)=*combine*(*B*, *A*) if *mach(A, B).*

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	x	x		x				x	x			■	x	x	
B	■							x	x	x	x	x	x	x	x

R	■							x	x			■	x	x	

Figure 2: A Result of *combine*() function in PSA algorithm

Figure 2 shows a result of *combin*e() function. The outcomes of the definition 2.1 are $R_8 - R_{15}$ and R_1-R_7 come from the definition 2.2. From definition 2.1, the gray nodes can be changed to black or white because the gray node is an unknown status.

Definition 3: The *match*() function is used to validate two frames before combination. Only two matched frames can be combined. The notation *match*(*A*,*B*) means that the frame *A* and *B* are matched before the combination process in definition 2 starts. Frame *A* and *B* are matched only if all nodes in these two frames meet this condition:

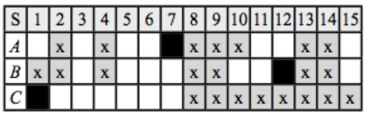
$$\begin{aligned} match(A,B) \leftrightarrow \forall i_{i\in N} A_i &= GRAY \\ \vee (A_i &= BLACK \wedge B_i = GRAY) \\ \vee (A_i &= WHITE \wedge B_i \neq BLACK) \end{aligned}$$

The condition is explained that:

3.1) A_i is gray node while B_i is any status because A_i is unknown status and can be replaced with any status of B_i.

3.2) A_i is black node and B_i is gray node mean that A_i is reserved for node i^{th}. They can be combined because B_i can be changed to any status.

3.3) A_i is white node while B_i is not black node. If any node is blocked in frame *A*, the same node in frame *B* must be blocked or still as unknown status.



S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A		x		x			■	x	x	x			x	x	
B	x	x		x				x	x			■	x	x	
C	■							x	x	x	x	x	x	x	x

Figure 3: Result of *match*() in PSA algorithm
(Frame A and B are matched while Frame A and C are not matched)

The figure 3 is and example of the *match*() function. Frame *B* matches with frame *C* while frame *A* does not match with frame *B*. When we determine slot B_1, B_2, B_4, and C_8-C_{15}, we found that they match because of definition 3.1. Slot B_3 and B_5-B_7 match because all nodes are white nodes in *B* and *C* as shown in definition 3.3. In the same way, slot C_1 and B_{12} also match because of definition 3.2. There are black whereas the other time slots are gray nodes. From *match*(*A*,*B*), we can conclude that they do not mach because A_7 and B_7 conflict with definition 3.2. One of them is black while the other is white. Thus, they could not be combined.

3.2 Algorithm

The wireless sensor network is represented based on a undirected graph *G*=(*V*,*E*) where *V* represents the set of sensor nodes and *E* represents the set of edges. In the case of $(u,v)\in E$, it means that node *u* sends packets directly to node *v*, they are one hop apart. Furthermore, if *u* and *v* are not one hop apart but have an intermediate node *k* such that $(u, k)\in E$ and $(k, v)\in E$, nodes *u* and *v* are said to be two hops apart.

This algorithm consists of three phases. First, the network topology represented in *G*=(*V*,*E*) is transformed to scheduling matrix, *S*, called *scheduling matrix initiation* phase. After that we reduce the frame length of scheduling matrix with *frame length minimization* phase in order to minimize the average delay. The final phase is to maximize the throughput and channel utilization that called *throughput maximization* phase. The details of all phases are explained below:

Phase I) Scheduling matrix initiation

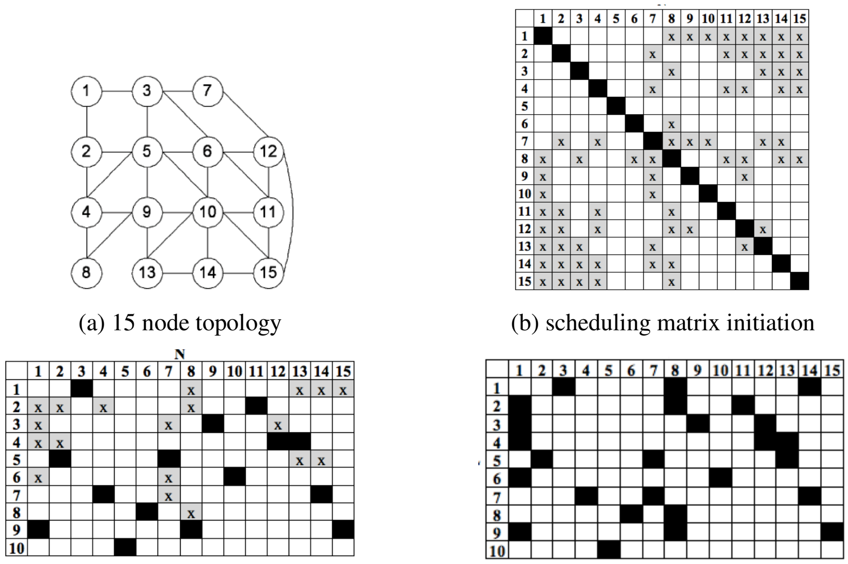
The scheduling matrix initiation is the first phase. The network topology is represented in *V* denotes the set of sensor nodes, and *E* which denotes the set of edges. Both *V* and *E* are the input of algorithm 1 and the scheduling matrix, *S*, is the result of this phase. The square scheduling matrix consists of columns and rows sized |*V*|. Each row is a list of time slots called frame, F_n.

The f_{ni} is the status of node *i* in frame *n* and is represented by a color as explained before. Therefore, the number of rows in the scheduling matrix is called frame length.

Algorithm 1 scheduling matrix initiation

```
for $u \in V$ do
    Set GRAY to all member for list, $F_u$
    $f_{uu} = BLACK$
    for $v \in V$ do
        if $(u,v) \in E$ then
            $f_{uv} = WHITE$
            for $k \in V$ do
                if $(k, v) \in E$ then
                    $f_{uk} = WHITE$
                end if
            end for
        end if
    end for
    $S = S \cup \{F_u\}$
end for
```

Algorithm 1 is explained that all node statuses in frame, F_u, are set to gray. The node, f_{uu}, is set to black mean that this frame is granted for node *u*. All adjacency nodes, $(u, v)\in E$, are set to white in order to prevent direct collision and all adjacency nodes, $(k, v)\in E$, are set to white in order to prevent hidden collision. Finally, frame, F_u, is added to the schedule matrix, *S*. This algorithm will be repeated for every sensor node in *V*. We get the scheduling matrix, *S*, and frame length |*V*| when the first algorithm finishes.



(a) 15 node topology

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	■							x	x	x	x	x	x	x	x
2		■					x				x	x	x	x	x
3			■					x					x	x	x
4				■			x				x	x		x	x
5					■										
6						■		x							
7		x		x			■	x	x	x			x	x	
8	x		x			x	x	■			x	x		x	x
9	x						x		■			x			
10	x						x			■					
11	x	x		x				x			■				
12	x	x		x				x	x			■	x		
13	x	x	x				x					x	■		
14	x	x	x	x			x	x						■	
15	x	x	x	x				x							■

(b) scheduling matrix initiation

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1			■					x					x	x	x
2	x	x		x				x			■				
3	x						x		■			x			
4	x	x										■	■		
5		■					■						x	x	
6	x						x			■					
7				■			x							■	
8						■		x							
9	■							■							■
10					■										

(c) frame length minimization

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1			■					■						■	
2	■							■			■				
3	■								■			■			
4	■											■	■		
5		■					■						■		
6	■									■					
7				■			■							■	
8						■		■							
9	■							■							■
10					■										

(d) throughput maximization

Figure 4: The PSA algorithm

Figure 4a and 4b give an example of algorithm in the first phase. A network topology with 15 nodes is shown in figure 4a. The scheduling matrix in figure ab is the result from phase 1. This matrix sized 15*x*15 consists of 15 black slots that are granted as one slot for each node. Moreover, there are 70 gray slots that can be changed with the next phases. The frame length can be optimized with algorithm 2, while the gray slots are changed to black or white using algorithm 3.

Phase 2) Frame length minimization

The frame length indicates the average waiting time of a sensor node. For example, the node 1 must wait for 14 frames in order to send a packet in its next turn. To minimize the frame length of the schedule matrix, we group all frames with *combine*() and *match*() functions as defined in the previous section based on the greedy algorithm.

The input of this phase is the scheduling matrix, *S,* while the output is the minimized frame length of scheduling matrix. Let F_a, F_b, and *R* denote frames in the scheduling matrix. The algorithm of phase 2 is explained below whereas the *max()* function is defined in algorithm 3 .

Algorithm 2 frame length minimization

```
1: loop
2:    if $F_a = max(S, NULL)$ and $F_b = max(S, F_a)$ and $match(F_a, F_b)$ then
3:        $R=combine(F_a, F_b)$
4:        $S = S - \{F_a, F_b\}$
5:        $S = S \cup \{R\}$
6:   else
7:       return $S$
8:   end if
9: end for
```

The *max()* function finds the frame of S with the maximum number of gray nodes (other than one already chosen frame).

Algorithm 3 *max(S, F)* function

```
Input : $S$ is scheduling matrix and $F$ is frame
R = NULL
g =0
for $r \in S$
    if $r == F$ then
        continue
    if $gray(r) > g$ then
        $g = gray(r)$
        $R = r$
    end if
end for
return R
```

The weighting function is shown in the second line of algorithm 2. The algorithm selects two frames that contain the maximal gray slot because they have a high probability of matching successfully and provide the most gray slot after combination.

The algorithm repeats all statements until there are no matched frames according to the condition in the second line. For each round, it finds two frames from the schedule matrix under two conditions: 1) They are the first and second frames that provided the maximum gray slot and 2) two frames must follow the definition 3. After that, the selected frames are removed and combined to be the new frame, *R*. The new frame, *R*, is added into the schedule matrix. If the condition in the seventh line is true, this phase will stop and return an optimal scheduling matrix, *S*. Finally, we get the new schedule matrix that provides a minimal frame length as shown in figure 4c.

The numbered black slots from this phase are equal to the initial scheduling matrix. However the frame length and number of gray slots are reduced. The next phase replaces gray slots with black slot in order to increase throughput. Phase 3 still relies on *match*() and *combine*().

Phase 3) Throughput maximization

Throughput maximization is the last phase of PSA. This phase increases the number of black nodes by replacing gray with black or white color in order to increase the throughput. However, the node replacement must follow *match*(), *combine*() and algorithm 4. The input of this phase is the scheduling matrix shown in figure 4c. The algorithm eliminates gray slots and replace with black or white. Moreover, the initial scheduling matrix, *iS*, produced by the first phase is used in this phase. At the end of this phase, the new scheduling matrix, *S*, is composed entirely of black and white slots.

Algorithm 4 throughput maximization

```
for $u \in V$ do
    for $F_v \in iS$ do
        If $f_{vu} = GRAY$ and $match(iF_u, F_v)$ then
            $F_v = combine(iF_u, F_v)$
        end if
    end for
end for
replace all gray nodes with white nodes
```

The main idea of this phase is to replace all gray slot that are valid with *match*(). The scheduling matrix is traversed in column order to find a gray slot. Fore example in figure 4c, the first node contains four gray slots and one black slot. The second frame of the first node is a gray slot. That means the first node may transmit the packet without collision. In order to ensure that the first node can send packet in this frame, the frame iS_1 and F_2 are tested with *match*() function. They are merged with *combine* function only if they are matched. After frame combination, the gray slot of the second node in second frame is replaced with white slot because of the *combine*() function. All gray slots in the first column are replaced with black that result in gray slots of the other columns are changed to be white slot. The fourth column is changed to white slot. Therefore, the eighth column will be processed in the next step. Finally, a optimal scheduling matrix is generated and shown in figure 4d.

The packet scheduling algorithm transforms the network topology to be a scheduling matrix. All node members in the matrix are set to black, gray or white color. The PSA combines two frames that tested by *match*() and *combine*() functions in order to reduce the frame length and increase black slots. Both frame length minimization and throughput maximization phases are based on greedy algorithm. A mathematical evaluation by comparing with the previous works in terms of throughput, average delay and channel utilization will be presented in the next section.

4. MATHEMATICAL EVALUATION

Packet collision minimization is the primary goal of the proposed algorithms in the broadcast scheduling problem (BSP). However, the packet scheduling cause effects upon network such as average delay, throughput, and channel utilization. This section explains the three performance metrics that are used to evaluate the proposed algorithm and compare the PSA with the previous algorithms using network benchmarks.

4.1 Performance Metrics

There are three performance metrics for mathematical evaluation of the PSA algorithm which are throughput, average delay, and channel utilization.

Throughput (σ: slots) It is the number of reserved time slots, or black slots, that are assigned to sensor node. The throughput is calculated using the equation below. The schedule matrix, *S*, is of size |*V*|*x*|*S*|. |*V*| denotes the number of nodes and |*S*| denotes the frame length, and s_{ij} is the status of node in each time slot.

$$\sigma = \sum_{i=1}^{N} \sum_{j=1}^{L} s_{ij}$$

when

$$s_{ij} = \begin{cases} 1 & \text{if } \mathrm{s_{ij}} \text{ is black} \\ 0 & \textit{otherwise.} \end{cases}$$

Averaged delay (τ: frames). This indicates the waiting time of a sensor node between opportunities to transmit. The average delay is calculated by the equation below. This metric depends on the frame length and number of black slots per node. If any algorithm can reduce the frame length and generate the same throughput, the average delay will different. The distribution of black slots can determine the average delay. A high distribution gives a lower average delay compared to a low distribution.

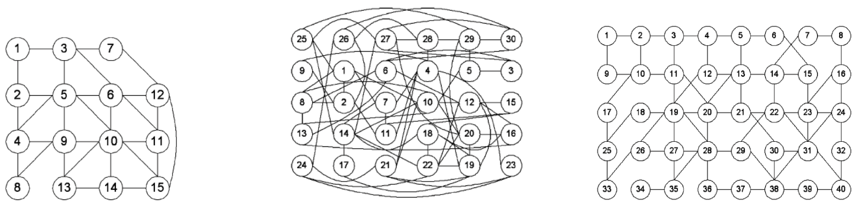
$$\tau = \frac{|S|}{|V|} \sum_{i=1}^{|V|} \left(\frac{1}{\sum_{j=1}^{|S|} s_{ij}} \right)$$

Channel Utilization (η :%): We trade-off between the throughput and the average frame delay. More frames mean more available time. On the other hand, a high frame length can increase the averaged frame delay. Therefore, the channel utilization is the best metric to measure the performance of the algorithm. The channel utilization is calculated from the equation below.

$$\eta = \frac{\sigma}{|S| x |V|} x100$$

4.2 Results and Discussions

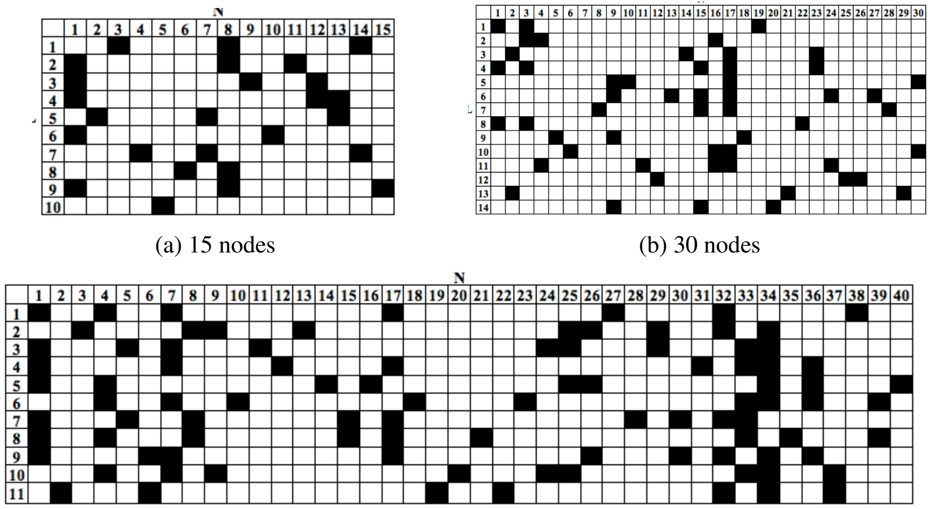
This section compares the PSA with other algorithms. All algorithms are tested with the network topology introduced by [9], which has become the benchmark test cases for the broadcast scheduling problem. The network benchmarks consist of three topologies with 15, 30, and 40 nodes as shown in figure 5. The maximum of neighbor node of all benchmarks are 7 nodes as indicated by the minimal frame length of the scheduling matrix.



(a) I-15 nodes (b) II-30 nodes (c) III-40 nodes

Figure 5: Network benchmarks

All benchmarks are scheduled with the PSA and the other algorithms. The scheduling matrixes of PSA are shown in figure 6. Each matrix consists of frame (row) and sensor node (column). The frame lengths of the three benchmarks are 10, 14, and 11, and throughput (black slots) are 26, 53, 94 slots. Each frame consists of time slots that are filled with black or white color. Node *j* in frame *i* filled with black color means that node *j* sends a packet in frame *i* with no collision. For example the first frame in figure 6a is reserved for node 3, 8, and 14. Thus, node 3, 8, and 14 are granted permission to send a packet in this frame while the other nodes are blocked.



(a) 15 nodes

(b) 30 nodes

(c) 40 nodes

Figure 6: scheduling matrix

The performances of all algorithms are shown in table 1. The performance metrics of three scheduling matrixes are calculated with the equations in section 4.1 and compared with the other algorithms. We compare the PSA with the previous works using the statistical method: one sample t-test. The PSA is compared with the average of old methods for each performance metrics and topology. The hypothesis assumption is the performance of PSA differs from the previous works. We found that the performance metrics are mostly different from the previous works with the confidence level at 95% in contrast with the channel utilization of 40 nodes topology.

Table 1: Performance comparison

benchmark		TABU	HNN	BSC	MFA	SVC	FSM	PSA
15 nodes	$\lvert S \rvert$	-	-	8	8	8	8	10
	σ	20	-	20	18	18	20	26
	τ	-	6.80	7.00	7.20	7.20	6.84	7.63
	η	-	-	16.67	15.00	15.00	16.67	17.33
30 nodes	$\lvert S \rvert$	-	-	10	9	11	10	14
	σ	37	-	35	38	37	35	53
	τ	-	9.20	9.30	10.67	9.99	9.20	10.99
	η	-	-	11.67	10.56	11.21	11.67	12.62
40 nodes	$\lvert S \rvert$	-	-	8	8	8	8	11
	σ	68	-	77	71	60	64	94
	τ	-	5.80	6.30	6.99	6.76	6.00	8.39
	η	-	-	24.06	19.72	18.75	20.00	21.36

The TABU focused on throughput maximization while the HNN focused on average delay minimization. Therefore, they show only throughput or average delay while other approaches determined both average delay and throughput concurrently.

Most algorithms reduce the frame length to 8.0 frames on benchmark I and III. The average frame length of benchmark II is 10.0 ± 1.29 frames (average value ± 95%CI). The PSA reduces the frame length significantly less than other algorithms. The frame length of PSA in the three benchmarks is more than previous works by 25%, 40%, and 37.5% respectively. The average throughputs for all benchmarks are 19.2, 36.4 and 68 slots, respectively. There is 95% confidence to believe that throughput of each algorithm is not different. The PSA generates the free time slot (black node) significantly more than previous works up to 30.00%, 39.47%, and 22.07% on 15, 30, and 40 nodes respectively.

The average delay (τ) and channel utilization (η) are calculated from the equation in section 3.1. The average delay varies directly with frame length and throughput whereas channel utilization also varies directly with throughput but varies indirectly with frame length. The average delays of PSA are more than the other algorithms. The average delays of previous works are 6.96, 9.67, and 6.37 for the three network benchmarks. The delays of each algorithm do not difference significantly but results from PSA are greater than all other algorithms. Because PSA has a frame length longer than the other algorithms, this disadvantage causes an advantage in free slot allocation and leads to throughput increasing. The PSA generates significantly more throughput than other algorithms because there is more free space in the scheduling matrix. Because of the maximal throughput, the channel utilization of PSA is better than most algorithms and most benchmarks except the BSC in 40 nodes topology.

Table 2 shows the first and second algorithms that produce the lowest average delay, the highest throughput, and the highest channel utilization. There are three algorithms that have better performance than other algorithms such as PSA, FSM, and HNN.

Table 2: The first and second algorithm ordered by performance metrics

benchmark		τ	σ	η
(1)	1st	HNN	PSA	PSA
15 nodes	2nd	FSM	TABU, FSM, HNN	BSC,FSM
(2)	1st	FSM, HNN	PSA	PSA
30 nodes	2nd	BSC	MFA	BSA, FSM
(3)	1st	HNN	PSA	BSC
40 nodes	2nd	FSM	BSC	PSA

To compare average delay, HNN is the algorithm that reduces the packet collision under the minimum average delay and FSM is the second. The average delay of PSA is more than other methods because it has the highest frame length. Although the PSA generates the highest throughput, it is not enough to minimize the average delay. Throughput and frame length are not the main factors that affect the average delay. The number of slots per node in the scheduling matrix is the main factor instead. If each node has been allocated fairly, it will result in lower average delay. Figure 6a is the example. The PSA allocates 5 slots for node 1 while most other nodes are allocated only 1 or 2 slots. In contrast, the FSM gives approximately the same number of allocated slots for all nodes. Because of this, the average delays of FSM are less than PSA in spite the throughput of PSA being more than FSM.

The PSA utilizes the channel better than the other algorithms in all the benchmarks. The frame length of PSA is significantly more than all algorithms, up to 25-40%, and PSA produces the maximal throughput. Except on benchmark III, the throughput of PSA is more than BSC by up to 37.5%. In benchmarks III, the BSC utilizes the channel better than PSA by up to 12.64% because the frame length of BSC is less than PSA by up to 22.02%.

5. CONCLUSION

The packet scheduling algorithm is to schedule packet in network layer and higher to reduce packet congestion in MAC layer and to reduce the packet collision and end-to-end delay; better packet delivery ratio is a by product. This algorithm is based on a greedy technique that is simple and easily implemented in a sensor node.

This paper measured the performance of the PSA with mathematical results in term of frame length, throughput, average delay, and channel utilization. The PSA is compared to previous works with network benchmarks. Our algorithm produces the highest throughput and utilizes the channel better than other algorithms. The PSA limitation is that the average delay is more than other algorithms. If we consider mathematical results only, it can not be concluded that any algorithm is suitable for wireless sensor networks. The PSA should be simulated and implemented on network simulation in order to determine performance in network perspective and we hop to publish the results soon.

ACKNOWLEDGEMENTS

Authors thank Teacher Development Scholarship for the PhD program, Walailak University, Thailand which funded this research and Robert Elz for English proofing

REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. IEEE Communications Magazine, 40(8):102–105, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. Computer Networks, 52:2292–2330, August 2008.

[3] L. Zhaohua and G. Mingjun. Survey on network lifetime research for wireless sensor networks. In Broadband Network Multimedia Technology, 2009. IC-BNMT ’09. 2nd IEEE International Conference on, p. 899 –902, October. 2009.

[4] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. Mac essentials for wireless sensor networks. Communications Surveys Tutorials, IEEE, 12(2):222 –248, quarter 2010.

[5] A. El-Hoiydi. Spatial tdma and csma with preamble sampling for low power ad hoc wireless sensor networks. In Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on, p. 685 – 692, 2002.

[6] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans. Networks, 12:493–506, June 2004.

[7] T. Van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In Proceedings of the 1st international conference on embedded networked sensor systems, SenSys ’03, p. 171–180, 2003.

[8] Y. Peng, B.H. Soong, and L. Wang. Broadcast scheduling in packet radio networks using mixed tabu-greedy algorithm. Electronics Letters, 40(6):375 – 376, March 2004.

[9] G. Wang and N. Ansari. Optimal broadcast scheduling in packet radio networks using mean field annealing. Selected Areas in Communications, IEEE Journal on, 15(2):250 –260, February 1997.

[10] S. Salcedo-Sanz, C. Bousono-Calzon, and A.R. Figueiras-Vidal. A mixed neural-genetic algorithm for the broadcast scheduling problem. Wireless Communications, IEEE Transactions on, 2(2):277 – 283, March 2003.

[11] J. Yeo, H. Lee, and S. Kim. An efficient broadcast scheduling algorithm for tdma ad-hoc networks. Computers and Operations Research, 29(13):1793 – 1806, 2002.

[12] H. Shi and L. Wang. A hybrid neural network for optimal tdma transmission scheduling in packet radio networks. In Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on, volume 5, p. 3210 – 3213 vol. 5, August. 2005.

[13] I. Ahmad, B. Al-Kazemi, and A. Shoba Das. An efficient algorithm to find broadcast schedule in ad hoc tdma networks. Journal of Computer System, Networks, and Communication, 2008:12:1–12:10, January 2008.

[14] I. Ahmad and A. Shoba Das. A heuristic algorithm for the minimization of incompletely specified finite state machines. Computers & Electrical Engineering, 27(2):159 – 172, 2001.

[15] IEEE standard 802.15.3, wireless medium access controls (mac) and physical layer (phy) specifications for high rate wireless person area networks (wpans), September 2003

Authors

Chaiyut Jandaeng is a lecturer at the School of Informatics of Walailuk University. He completed the B.Sc and M.Sc. in Computer Science, PSU. He is a Ph.D candidate in Computer Engineering at Prince of Songkla University, Thailand. He was a teacher assistance and guest lecturer in information technology after completing his masters, before beginning the doctoral research. Areas of interest are computer networks security, and algorithm and programming technique in resource constrained devices.

Wannarat Suntiamorntut received the Ph.D Degree from the University of Manchester, UK. She was a research assistant at the School of Computer Science, the University of Manchester for 3 years, while she was completing the Ph.D. program. Since 1999, she has been a lecturer at the computer engineering department at Prince of Songkla University, Hat Yai Thailand and is now assistant professor. In 2008, she became the Director of Collaborative Research Unit in Wireless Sensor Network (CRU-WSN), a joint collaboration between National Electronics and Computer Technology Center and Prince of Songkla University.

Nittida Elz received her B.Sc. degree in Mathematics from Prince of Songkla University, Thailand. She then received her M.Sc. degrees in Computer Science from Chulalongkorn University, Thailand and The University of Melbourne, Australia, followed by a Ph.D. degree in Computer Engineering from La Trobe University, Melbourne, Australia. Her research interest is network management and security.

International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 4, August 2011

PERFORMANCE PREDICTION OF PACKET SCHEDULING ALGORITHM IN WIRELESS SENSOR NETWORKS

[1]C. Jandaeng,[1]W. Suntiamontut, [2]N. Elz

[1]Centre of Excellence in Wireless Sensor Network,
Department of Computer Engineering, Prince of Songkla University
cjundang@gmail.com

[2]Computer System and Network Laboratory,
Department of computer Science, Prince of Songkla University,
Hat Yai, Thailand 90112

ABSTRACT

The main cause of the wasted energy consumption in wireless sensor networks is the packet collision. The packet scheduling algorithm is therefore introduced to solve this problem. This paper presents an analytical model for predicting the performance of packet scheduling algorithms in wireless sensor networks. This model can be used to predict the performance of the packet scheduling algorithm with different node density and CBR in wireless sensor networks. The proposed prediction model can give the packet collision and packet delivery ratios very close to the experimental results. The accuracy of our model is 77% of packet collision model and 81% for packet delivery ratio model.

KEYWORDS

packet scheduling, wireless sensor network, performance prediction model

1. INTRODUCTION

A wireless sensor network is a self-configured network containing numerous small sensor nodes. Each node consists of sensing modules, a processing unit, radio frequency components and power sources [1]. They organize and communicate among themselves in an ad-hoc fashion. The wireless sensor networks technology has been deployed in several applications such as health care monitoring systems, home automation and environment monitoring systems [2]. These applications require inexpensive facilities and little manual maintenance. According to the application requirements, each node has been implemented using a low-power microcontroller and radio module. In addition node is supplied with a small battery. Energy usage is the indicator of network lifetime [3].

All sensor nodes share a single channel using a multiple access protocol. The packet transmission may lead to a time overlap of two or more packet receptions, called collisions. The packet collision problem causes the packets loss, packet retransmission, decreasing throughput, increased delay/latency and increased wasted energy consumption. Many research works on MAC protocol have been proposed to solve the packet collision [4] such as Spatial TDMA traffic-adaptive medium access protocol (TRAMA) [5], Sensor MAC (SMAC) [6], and

DOI : 10.5121/ijwmn.2011.3408

Timeout MAC (TMAC) [7]. MAC protocol based on IEEE 802.15.4 was developed for low-power communication. This protocol uses a random back off in order to reserve and access the channel. A node is authorized to send the packet when the channel is idle. In contrast, random back off is activated when the channel is busy. Unfortunately, this technique will not work properly when used in a large scale wireless sensor network.

Time Division Multiple Access (TDMA) is a solution to reduce the packet collision problem. Total transmission time is divided into frames and each frame is divided into time slots. After that each time slot will be assigned to a sensor node to guarantee that every node is granted to send a packet in its time slot permission guaranteeing collision avoidance. The latency directly varies with frame length. On other hand, the throughput inversely varies with frame length. There have been many approaches presented to minimize the frame length and maximize the throughput which are explained in section 2.

There have been some packet scheduling algorithms [15] for wireless sensor networks proposed recently. Packet Scheduling Algorithm (PSA) has to be considered carefully before it is chosen to deploy in each application. The packet collision ratio and the packet delivery ratio (PDR) are the main parameters which the developer uses to consider how efficient the PSA is. Therefore, this paper proposes a performance prediction model of our PSA to help the developers investigate whether the PSA is suitable for their applications.

The remainder of the paper is organized as follows. First we briefly introduce the packet collision problem and the existing packet scheduling algorithms in Section 2. Our system model and the packet scheduling algorithm are explained in section 3. The performance prediction model is proposed in Section 4. The prediction results are discussed in Section 5. Finally we give conclusions about the performance prediction model in Section 6.

2. PREVIOUS WORKS

Peng *et al* [8] presented the TDMA with a scheduling matrix. The row of the matrix denotes frame length while the column of matrix denotes nodes. The members of the matrix represent transmission authorization. In [8], they proposed to optimize the number of row that refers to the frame length with Tabu search and greedy algorithm. This approach can reduce the average latency and produce a high throughput in a dense area.

Wang, G. and Ansari, N. [9] have proved that the scheduling matrix optimization is a NP-complete problem. They also proposed the approximation method, mean field anneal (MFA) to optimize the schedule matrix. The matrix optimization is divided into two phases: minimize frame length and maximize throughput. After that approximation methods were proposed. Salcedo-Sanz, S *et al* [10] minimized frame length with a neural network (NN) and maximized throughput with a genetic algorithm (GA), whereas Yeo, J. *et al* [11] applied the sequence vertex coloring (SVC) in both phases. Moreover, Haixiang, S. and W. Lipo [12] proposed a hybrid algorithm which combined back tracking sequential coloring (BSC) and noisy chaotic neural network (NCNN) to optimize the scheduling matrix. BSC-NCNN gives the most minimal average time delay, while the NN-GA provides higher throughput.

Ahmad, I. *et al* proposed an idea to avoid packet collision. The network topology is represented by a finite state machine (FSM) [13]. The set of nodes are grouped with the maximal compatibles and incompatibles concept [14]. This idea started from setting up groups that equal the number of nodes. After that, combine groups together under the condition that all nodes in the same group are not neighbor nodes. Finally, all sensor nodes are grouped in many groups and they can send packet in the same time without collision. The number of groups is frame delay while the summation of number of node in all groups is throughput. This idea leads to minimize latency and maximize throughput.

3. SYSTEM MODEL AND PROBLEM STATEMENT

3.1 System Model

We assume that wireless sensor networks have *n* nodes using the same communication range with a sink node collecting the data from the other nodes. Therefore, we can represent the wireless sensor network as undirected graph $G=(V, E)$, where V is a set of nodes $(V_1, V_2, \ldots, V_n)$ and E is a set of edges for all communication links. In the case of $(u, v) \in E$, node *u* and *v* are one hop apart. Further, if *u* and *v* are not one hop apart and have an intermediate node *k* which $(u, k) \in E$ and $(k, v) \in E$, node *u* and *v* are two hops apart. The network topology as shown in Figure 1 network topology represented with graph theory can be represented by graph theory $V=\{1, 2, 3, 4, 5\}$ and $E=\{(1, 2), (2, 1), (2, 3), (3, 2), (3, 4), (4, 3), (3, 5), (5, 3), (4, 5), (5, 4)\}$.

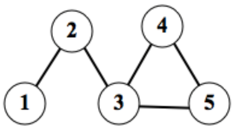
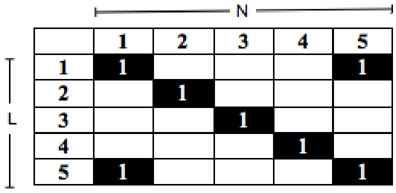


Figure 1 network topology represented with graph theory

There are two types of packet collisions: direct collision and hidden collision. Let $u, v, k \in V$ that are one hop apart. If *u* transmission time overlaps with *v* transmission time, a direct collision will occur. On the other hand, a hidden collision will occur when both *u* and *v* transmit the data to *k* at overlapping times, and *u* and *v* are not one hop apart. For example, there is a direct collision between node 4 and 5 because of sharing the same channel with node 3 as shown in Figure1. The hidden collision will occur between node 2 and 4 when both of them send the packet to the same destination (node 3) at the same time.

3.2 The Scheduling Matrix

In order to avoid packet collision, the scheduling with TDMA approach is introduced. The network topology can be represented by the scheduling matrix as shown in Figure 2.



	1	2	3	4	5
1	1				1
2		1			
3			1		
4				1	
5	1				1

Figure 2 scheduling matrix

Figure 2 shows the number of node, *N,* and the number of row, *L.* The number of row refers to frame length and refers to delay. The matrix elements assigned with 1 refers to throughput. In addition, if frames 1 and 5 are duplicated, then frame 5 can be deleted to reduce frame length.

The frame length is decreased while channel utilization is increased. The channel utilization is ratio between throughput and channel availability. There have been several methods proposed to find and remove unnecessary frames to reduce delays and increase channel utilization as explained before.

3.3 Packet Scheduling Algorithm

The Packet Scheduling Algorithm (PSA) [15] is the algorithm that schedules all packets from application layer and network layer in order to reduce network congestion in the data link layer to avoid the packet collision. When the PSA is implemented, packet collisions will be minimized and increasing of throughput as a by product. A PSA that based on greedy algorithm is a simple algorithm and easily implemented in a sensor node. The basic assumptions of the PSA are defined as follows. All packets communicate via IEEE 802.15.4 standard that avoids packet collision with a simple CSMA/CA mechanism. All sensor nodes must know the information of at least 2-hops neighbor nodes. Finally, time synchronization, neighbor discovery, and routing protocols are not considered in this work.

The color of each node represents node status. Two functions, *combine()* and *match(),* are used to reduce the frame length. The node statuses and functions are defined below:

Definition 1. *Node status* is represented by a node color for each time slot. A black node can send any packet with a guarantee of no collision. If a white node requests to send a packet, its packet may collide. A gray node is the initial status with no guarantees for packet collision. Finally, a gray node can change status to other colors by *combine()* and *match().*

Definition 2. *combine()* is used to reduce frame length by combining two frames. The two frames must be tested with *match()* before combination. They will be merged in case the *match()* function returns valid. In the combination process, the status of a node will be changed to another color as defined below. Let *A, B* and *R* denote frame and A_i, B_i, and R_i be node status i^{th} time slot of *A, B,* and *R; V* denotes the set of nodes and *R=combine(A,B).*

$$\forall_{i\, i\in V}, R_i = \begin{cases} A_i : if\ B_i = GRAY \\ B_i : \ \ otherwise. \end{cases}$$

Definition 3. The *match()* is used to validate two frames before combination. All node status must follow the condition below:

$$\begin{aligned} match(A,B) \leftrightarrow & \forall_{i_{i\in N}} A_i = GRAY \\ & \vee (A_i = BLACK \wedge B_i = GRAY) \\ & \vee (A_i = WHITE \wedge B_i \neq BLACK) \end{aligned}$$

The algorithm consists of two phases: frame length minimization and throughput maximization. In the frame length minimization phase, frames are sorted by the number of gray nodes. Sensor nodes find the two frames that have the most gray nodes and are valid with *match().* After that both frame are merged with *combine()* and added into scheduling matrix. All frames are sorted again. All steps are repeated until we cannot find two frames that are valid with *match().* In the throughput maximization phase, a sensor node scans the scheduling matrix in column order to find the gray node. If column *ith* has a gray node in frame *jth,* the frame *ith* from this scheduling matrix and *jth* from original scheduling matrix are merged with *combine()* when they are matched. The details and performance analysis of this algorithm are compared with previous work and explained in [15].

4. PERFORMANCE PREDICTION MODEL

4.1 Simulation

The IEEE 802.15.4 module for wireless sensor networks has been proposed by Jianliang Zhen and declared to be a standard module in Network Simulator (NS2). The details of the IEEE 802.15.4 standard are described in [16]. Wireless sensor networks have mostly been deployed in two types of applications, surveillance and tracking systems. The data communication in these applications is periodic as Constant Bit Rate (CBR) in NS2. The User Datagram Protocol (UDP) is selected as the transport protocol in order to disable the packet retransmission mechanism. Wireless sensor networks are similar to ad hoc networks. Thus we choose the AODV routing protocol to use in the network simulation. The data link layer and physical layer rely on non-beacon IEEE 802.15.4 standard that manages packet collision with simple CSMA/CA [17].

Three independent variables: density of node, CBR rate and quantum time are relevant to this performance model. From the density of node perspective, the sensor nodes in transmission range must receive and process incoming packets. Some packets must be forwarded which will increase network congestion and also lead to high packet collision. All sensor nodes are randomly placed in a field sized 40x40 m^2. Node density (*d*) is varied with $0.01 <= d <= 0.1$.

The CBR rate indicates the number of packets sent from a node. When the CBR data is passed to the lower layer, the overhead such as routing packet and MAC packet are added. In order to evaluate the effects of the traffic rate from the application layer, the CBR rates (*t*) are varied from 1 to 250 kbps ($1 <= t <= 250$) where the bandwidth of IEEE 802.15.4 standard is 250 kbps. All sensor nodes send the CBR packets with the size 40 bytes to the base station that is placed in the middle of the sensor field.

The quantum time is the time interval between frame i^{th} and $(i+1)^{th}$. If the quantum time is long, the sensor node has to wait longer than usual for the next frame. This will directly affect the packet delivery ratio. In AODV protocol [18], a node waits for the Route Reply (RREP) after broadcasting a Route Request (RREQ). If a route is not responded within 2,800 milliseconds, the node may discover a route by broadcasting another RREQ. Data from the upper layer protocol are buffered in a queue while waiting for the route. If the routing process takes a time longer than the transmission time of the upper layer protocol, the FIFO queue will become full and the data will be dropped. In the simulation, the quantum times (*q*) are varied from 0.1 ms to 1.0 ms ($0.1 <= q <=1$) in order to study the relation of the quantum time, packet collision rate and packet delivery ratio.

4.2 Performance metrics

The performance of the PSA is evaluated in terms of two aspects, packet collision rate and packet delivery ratio. The packet collision rate dominates the wasted energy in the network and the packet delivery ratio gives the throughput of the PSA. All parameters used in the simulation are described as follows:

- *Energy consumption*: The energy consumption of the radio transceiver can be calculated using the transmitting current (I_{TX}), the receiving current (I_{RX}) and the supply voltage (V) with the data rate of IEEE 802.15.4 standard (250kbps), one byte data will use $31.25\ \mu s$ for transmission. Then we can calculate the energy usage in the transceiver for a k bytes transmission using $E(k)=kTV(I_{TX} + I_{RX})$. For example, CC2420 module has an average current of 19.4mA and 17.4mA for data transmitting and receiving, respectively [19]. Therefore the total energy consumption of the CC2420 is *k2.74 uJ* for *k* bytes transmission when the supply voltage is 2.4 V. The main purpose of the PSA algorithm is to minimize the packet collision that leads to reduce wasted energy. Thus we can conclude that the energy dissipation depends directly on the packet collision rate.
- *Packet collision rate* (PCR) is the amount of dropped data in one second. All sensor nodes share a single channel using a multiple access protocol. Packet collision then occurs when a packet transmission leads to a time overlap of two or more packet receptions. High node density has higher possibility of collision than low node density. The CBR rate increases the number of control packets in the MAC layer increasing packet congestion.
- *Packet delivery ratio* (PDR) is the ratio of the total number of packets received by the nodes to the total number of packet transmitted. A PDR close to 1.0 means that the network has the high performance.

4.3 Multiple regression models

Regression analysis [20] is a statistical method to show the relation between dependent variables and independent variables. This method is used for a prediction based on the information collected in the past and the description of the relationship between dependent and independent variables. The proposed regression model is defined as:

$$y = f(x_1,\ldots,x_p) + \varepsilon, \varepsilon \in N(0,\delta^2)$$

$f(x_1,\ldots,x_p)$ is a regression function that represents the raw data. It may be a linear or non-linear function. ε is a random variable that represents the error and fits the normal distribution. In some cases, the relationship between dependent and independent variables is a non-linear function. The dependent variables are transformed to be linear data. The least square method (LSM) is a method to find the parameters ($\beta_0,\ldots, \beta_p$) and can produce ε^2 which is the smallest value. The linear regression function is presented follows:

$$f(x_1,\ldots,x_p) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

The R statistic package is used to obtain statistic value from the raw data. The correlation explains the relationship between each variable. The *F-test* indicates that the proposed model represents the observed data. In order to indicate the model accuracy, the goodness-of-fit and standard error are measured.

4.4 Performance model validation

To validate the performance model, NS2 has been used to simulate the simple CSMA/CA and CSMA/CA with PSA. The packet collision rate and packet delivery ratio results of the relevant parameters such as CBR rate (kbps), node density and quantum times have been analyzed the correlation hypothesis with Pearson's product-moment. Then the regression model is generated with the R package.

The multiple regression planes in tables 1 and 2 represent the simulation results. We can conclude that all regression models are represented in the simulation results with the strong evidence (*p-value* < *0.05*). There are few errors as can be noticed from the independent variables (R^2 and R^2_{adj} are closed). This information is very useful to assess the overall accuracy of the model.

Table 1 Packet Collision Rate of CSMA/CA

	Correlation		**Regression**		
	r	***p-value***	$\boldsymbol{\beta}$	***t***	***p-value***
log(t)	0.1025	0.000	0.24	7.216	0.000
d^2	0.9436	0.000	5177	58.426	0.000
$d^2 log(t)$	0.9352	0.000	322.90	16.489	0.000

$SE_{est} = \pm 5.389$

$R^2 = 0.9762; R^2_{adj} = 0.9762; F=40920; p\text{-}value = 0.000$

From table 1, we found that the packet collision rate is a weak correlation with CBR rate. In contrast, with the node density, the packet collision has a strong correlation. In simple CSMA/CA, the packet collision is related with CBR rate, density of node, and interaction between CBR rate and density of node with coefficient 0.24, 5177, and 322.90, respectively. The prediction model represents packet collision rate to 97.62% with the significant 0.05. The standard error of estimation is ± 5.389 kbps. When *t* denotes the CBR rate and *d* denotes the density of nodes, the prediction model of packet collision rate for simple CSMA/CA, *PCR(t,d)* can be shown below:

$$PCR(t,d) = 0.24\log(t) + 5177d^2 + 322.90d^2\log(t)$$

subject to:

$$1 \le t \le 250kbps$$
$$0.01 \le d \le 0.10$$

Table 2 Packet Delivery Ratio of CSMA/CA

	Correlation		**Regression**		
	r	***p-value***	$\boldsymbol{\beta}$	***t***	***p-value***
log(t)	0.1773	0.000	-0.94	-6.778	0.000
log(d)	-0.7645	0.000	-16.36	-84.688	0.000
log(d)log(t)	0.4970	0.000	2.57	42.982	0.000

$SE_{est} = \pm 7.659$

$R^2 = 0.7821; R^2_{adj} = 0.7819; F=3578; p\text{-}value = 0.000$

The CBR rate correlates with the packet delivery ratio less than density of node as packet collision. On the other hand, the density of nodes inversely varies with packet delivery ratio and the interaction between two parameters is average. The packet delivery ratio depends on density of nodes. The relationship between packet delivery ratio, density of node, and interaction of

CBR rate and density of nodes are –0.94, -16.36 and 2.57, respectively. The prediction model represents packet delivery ratio to 78.21% with the significant 0.05. The standard error of estimation is ± 7.659%. When *t* denotes CBR rate and *d* denotes the density of nodes, the prediction model of packet delivery ratio for a simple CSMA/CA, *PDR(t,d)* can be shown below:

$$PDR(t,d) = -0.94\log(t) - 16.36\log(d) + 2.57\log(t)\log(d)$$

subject to:

$$1 \le t \le 250kbps$$

$$0.01 \le d \le 0.10$$

The CBR rate and density of nodes affects the packet collision rate and packet delivery ratio. The density of node affects directly and growth in same trend line with packet collision rate and inversely growth with packet delivery ratio. The CBR rate has almost no correlation with either indicator.

When the PSA algorithm is applied together with the simple CSMA/CA, the results are similar to the simple CSMA/CA. However, the quantum time *(q)* is determined in this experiment. All parameters are plotted and tested with the R packages. After that the results are shown in Tables 3 and 4.

Table 3 Packet Collision Rate of CSMA/CA with PSA

	Correlation		**Regression**		
	r	***p-value***	***β***	***t***	***p-value***
$log(t)$	-0.0489	0.000	-0.08	-5.305	0.000
d^2	0.9576	0.000	5769.55	147.430	0.000
$log(q)$	0.0126	0.000	0.64	3.316	0.000
$d^2log(t)$	0.9246	0.000	307.96	35.209	0.000
$log(t)log(q)$	-0.0422	0.000	-0.17	-3.838	0.000
$d^2log(q)$	-0.9486	0.000	-362.99	-7.704	0.000
$d^2log(q)log(t)$	0.9153	0.000	105.23	10.244	0.000

$SE_{est} = \pm 5.197$

R^2 = 0.9792; R^2_{adj} =0.9792; F=201000; p-value= 0.000

Table 3 shows the statistic value of packet collision rate. The density of nodes affects the packet collision rate. All parameters that are concerned with the density of nodes are strongly correlated with packet collisions. Although the CBR rate and quantum time are weakly correlated with packet collision rate, these two parameters are strongly correlated with packet collision rate when they interact with density of nodes. The coefficient between packet collision rate implementing PSA algorithm, density of node, CBR rate, quantum time, and all reaction of all parameters are -0.08, 5769.55, 0.64, 307.96, -0.17, -362.99, and 105.23, respectively. The prediction model represents the packet collision rate to 97.92% with the significant 0.05. The standard error of estimation is ± 5.197 kbps. The prediction model of packet collision rate for simple CSMA/CA with PSA, *PCR(t,d,q),* is shown below:

$$
\begin{aligned}
PCR(t,d,q) = {} & -0.08\log(t) + 5769.55d^2 + 0.64\log(q) \\
& +307.96d^2\log(t) - 0.17\log(t)\log(q) - 362.99d^2\log(q) \\
& +105.23d^2\log(q)\log(t)
\end{aligned}
$$

subject to:

$$
\begin{gathered}
1 \le t \le 250kbps \\
0.01 \le d \le 0.10 \\
0.1 \le q \le 1ms
\end{gathered}
$$

The CBR rate directly affects the packet delivery ratio but in the inverse direction. The density of nodes has medium affect and packet delivery ratio and the quantum time has less effect on packet delivery ratio. The packet delivery ratio of CSMA/CA with PSA algorithm is shown in table 4. Moreover, the other parameters interact with high significance. The prediction model represents the packet collision rate to 72.16% with the significant 0.05. The standard error of estimation is ± 7.68%. When t denotes CBR rate and d denotes the density of nodes, the prediction model of packet collision rate for simple CSMA/CA with PSA, *PDR(t,d,q),* is shown below:

Table 4 packet delivery ratio of CSMA/CA with PSA

	Correlation		**Regression**		
	r	***p-value***	***β***	***t***	***p-value***
log(t)	-0.6523	0.000	-0.68	-10.08	0.000
log(d)	-0.2728	0.000	-15.08	-158.01	0.000
log(q)	0.0012	0.000	19.27	22.60	0.000
log(d)log(t)	0.3605	0.000	2.42	83.16	0.000
log(q)log(t)	-0.6136	0.000	-3.70	-19.54	0.000
log(q)log(d)	0.2494	0.000	4.51	16.34	0.000
log(t)log(q)log(d)	0.5476	0.000	-0.87	-14.20	0.000

$SE_{est} = \pm 7.868$

$R^2 = 0.7216;\ R^2_{adj} = 0.7216;\ F=11110;\ p\text{-value}= 0.000$

$$
\begin{aligned}
PDR(t,d,q) = {} & -0.68\log(t) - 15.08\log(d) + 19.27\log(q) \\
& +2.42\log(d)\log(t) - 3.70\log(q)\log(t) + 4.51\log(q)\log(d) \\
& -0.87\log(t)\log(q)\log(d)
\end{aligned}
$$

subject to:

$$
\begin{gathered}
1 \le t \le 250kbps \\
0.01 \le d \le 0.10 \\
0.1 \le q \le 1ms
\end{gathered}
$$

5. Results and Discussion

5.1 Prediction results

The main cause of packet collision is the large number of transmitted in the same interval time. This study noted that the data transfer rate in application layer (CBR rate) and the density of nodes cause the number of packets to increase in the MAC layer. After the simple CSMA/CA is applied with PSA the quantum time is determined.

We compare the packet collision and packet delivery ratio between the experimental results and the prediction model as shown in figures 3 to 6. The scatter plot of packet collision rate and packet delivery ratio from the simulation results and the prediction model are much close that concern with the R^2 of prediction model. The R^2 is close to 1.0 means that the *PCR(t,d)* and *PCR(t,d,q)* from the prediction model is close to the simulation results. Moreover, the packet collision rate varied directly with the density of nodes and CBR rate as a parabolic function and a logarithmic function. Whereas, the packet delivery ratio varies inversely with the density of node as a logarithmic function and varies directly with the interaction between the density of nodes and CBR rate as a logarithmic function. From these observations, we found that these relationships are very useful when the prediction of the packet collision rate and packet delivery ratio is required.

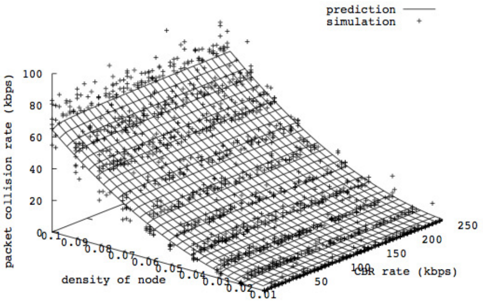


Figure 3 comparison of packet collision rate

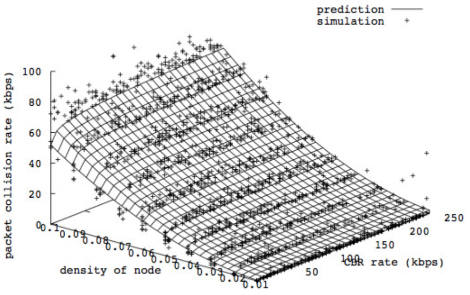


Figure 4 comparison of packet collision rate (with PSA)

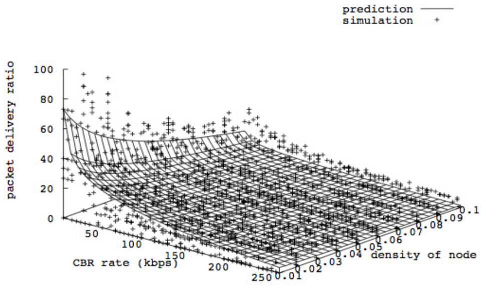


Figure 5 comparison of packet delivery ratio

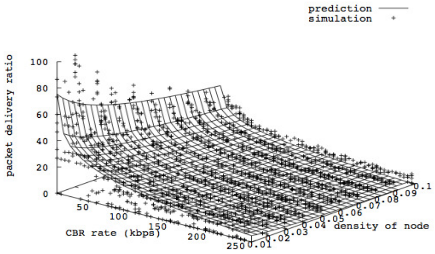
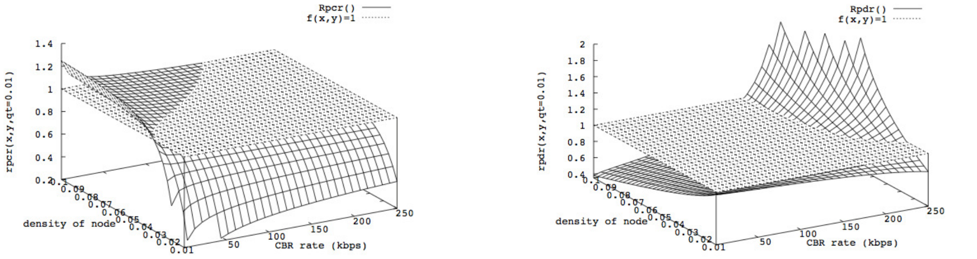


Figure 6 comparison of packet delivery ratio (with PSA)

5.2 Performance of PSA

This section compares the performance between a simple CSMA/CA and PSA. Both performance metrics are transformed to *Rpcr(t,d,q)* and *Rpdr(t,d,q)* functions that are the ratio between PSA and CSMA/CA. The *f(t,d) = 1.0* is threshold of packet collision rate and packet delivery ratio. Figure 5 shows the comparison between *Rpcr(t,d,q)* represented with solid surface and *f(t,d)* represented with dash surface. For packet collision rate, the solid surface being lower than the dash surface means that the packet collision of PSA is less than the standard. On the other hand, the solid surface of packet delivery ratio being higher than the dash surface mean that the CSMA/CA with PSA is better than the original CSMA/CA.



a) packet collision rate

b) packet delivery ratio

Figure 5 performance comparison

Figure 5a shows that the PSA decreases packet collision rate if the density of nodes is less than 0.06 for low CBR rare. It decreases packet collision rate 20% for the low density of nodes (d > 0.03) while packet collision rate is decreased up to 50% while the density of nodes is 0.02 and more than 50% when density of node closes to 0.01. However, this algorithm increases packet collision rate more than simple CSMA/CA only 5% when density of node closes to 0.1 and CBR rate is less than 75 kbps. Moreover, packet collision rate are increase up to 8% when the CBR rate closes to 1.0 kbps.

Unfortunately, PSA can reduce PDR. The ratio of PDR is shown in figure 5b. The PSA decreases PDR for in cases: 1) density of nodes less than 0.05 for all CBR rate and 2) density of nodes more than 0.05 if CBR rate is less than 150 kbps. The PSA increases packet delivery ratio less than 15%. On the hand, the PSA increases PDR up to 50% when CBR rate is more than 150 kbps and density of nodes more than 0.05. The PDR is 200% increased for high density of nodes and CBR rate. The density of nodes effects PDR more than CBR rate because it corresponds with packet collision. PDR from PSA is better than simple CSMA/CA with low density of node.

To evaluate the accuracy of prediction model, the 5000 samples from simple CSMA/CA and 50,000 samples from CSMA/CA with PSA are divided into two parts. The 60% of samples are used for prediction model generation whereas the 40% of samples are used for testing the prediction model. The PCR model and PDR model of simple CSMA/CA are 76.43% and 81.11% respectively. The prediction mode of standard with PSA is 77.64% for PCR and 80.05% for PDR.

The results in figure 5 show that the density of nodes is the main affect upon performance while CBR rate and quantum time have less impact upon performance. The next section is the factor analysis that affects the network performance.

5.4 Effects of parameters

This performance prediction model consists of three parameters: density of nodes, CBR rate, and quantum time. The density of nodes directly affects packet collision rate because packet collision rate grows in a parabolic function when the density of nodes is increased. The density of node is increased by 10% and other parameters are arbitrary values, the packet collision rate is increased in the interval of 20% to 50%. The density of node affects packet collision rate because route discovery packets (RREQ) are broadcasted in MAC layer and the number of RREQ packet at destination depends on the density of nodes. Therefore, it results in the packet having high congestion and high probability of packet collision.

The CBR rate affects packet collision for two reasons: controlled packet increasing and CBR packet in MAC layer. The controlled packet is high in both establishment phase and maintenance phase. When controlled packets are dropped, the routing protocol determines that topology is changed or cannot find a route and it results in the route discovery mechanism is restarted.

The quantum time has less affect upon packet collision rate because the packet collision rate is increased only 0.2% when quantum time is increased 10%. In contrast, the quantum time is the major impact upon packet delivery ratio, more than any other parameter. When quantum time is increased by 10% and other parameters are arbitrary values, the packet delivery ratio is decreased up to 21%.

We conclude that the density of nodes affects packet collision while other parameters have little affects. The quantum time and CBR rate affect packet delivery ratio more than density of nodes. However, the changing of parameters has almost no effect on the packet delivery ratio but density of nodes has a direct impact on packet delivery ratio.

6. CONCLUSION

The packet scheduling algorithm is proposed to reduce packet collision in wireless sensor networks. All packets in application and network layers are scheduled before forwarding to the data link layer. This can reduce the wasted packets. The proposed performance prediction model is able to help the developers to obtain the packet collision rate and the packet delivery ratio when they employ the packet scheduling algorithms in their applications.

The prediction models predict the packet collision rate and packet delivery ratio with accuracy 97.92% and 72.16%, respectively. The PSA reduces the packet collision 5% for the high density of node and up to 60-70% for the low density of nodes. On the other hand, the PSA decreases the packet delivery ratio 15% for the low density of nodes. On the hand, the PSA increases PDR up to 50% when CBR rate is more than 150 kbps and density of nodes more than 0.05. The PDR is 200% increased for high density of node and CBR rate.

The accuracy of PCR model and PDR model of simple CSMA/CA are 76.43% and 81.11% respectively. The prediction model is accurate to 77.64% and 80.05% for PSA and PDR, respectively.

ACKNOWLEDGEMENT

Authors thank Teacher Development Scholarship for the PhD program, Walailak University, Thailand which funded this research and Robert Elz for English proofing.

REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. IEEE Communications Magazine, 40(8):102–105, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. Computer Networks, 52:2292–2330, August 2008.

[3] L. Zhaohua and G. Mingjun. Survey on network lifetime research for wireless sensor networks. In Broadband Network Multimedia Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference on, p. 899 –902, October. 2009.

[4] A. Bachir, M. Dohler, T. Watteyne, and K.K. Leung. Mac essentials for wireless sensor networks. Communications Surveys Tutorials, IEEE, 12(2):222 –248, quarter 2010.

[5] A. El-Hoiydi. Spatial tdma and csma with preamble sampling for low power ad hoc wireless sensor networks. In Computers and Communications, 2002. Proceedings. ISCC 2002. Seventh International Symposium on, p. 685 – 692, 2002.

[6] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Trans. Networks, 12:493–506, June 2004.

[7] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In Proceedings of the 1st international conference on embedded networked sensor systems, SenSys '03, p. 171–180, 2003.

[8] Y. Peng, B.H. Soong, and L. Wang. Broadcast scheduling in packet radio networks using mixed tabu-greedy algorithm. Electronics Letters, 40(6):375 – 376, March 2004.

[9] G. Wang and N. Ansari. Optimal broadcast scheduling in packet radio networks using mean field annealing. Selected Areas in Communications, IEEE Journal on, 15(2):250 –260, February 1997.

[10] S. Salcedo-Sanz, C. Bousono-Calzon, and A.R. Figueiras-Vidal. A mixed neural-genetic algorithm for the broadcast scheduling problem. Wireless Communications, IEEE Transactions on, 2(2):277 – 283, March 2003.

[11] J. Yeo, H. Lee, and S. Kim. An efficient broadcast scheduling algorithm for tdma ad-hoc networks. Computers and Operations Research, 29(13):1793 – 1806, 2002.

[12] H. Shi and L. Wang. A hybrid neural network for optimal tdma transmission scheduling in packet radio networks. In Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on, volume 5, p. 3210 – 3213 vol. 5, July-4 August. 2005.

[13] I. Ahmad, B. Al-Kazemi, and A. Shoba Das. An efficient algorithm to find broadcast schedule in ad hoc tdma networks. Journal of Computer System, Networks, and Communication, 2008:12:1–12:10, January 2008.

[14] I. Ahmad and A. Shoba Das. A heuristic algorithm for the minimization of incompletely specified finite state machines. Computers & Electrical Engineering, 27(2):159 – 172, 2001.

[15] C. Jandaeng, W. Suntiamontut, and N. Elz. Throughput improvement of collision avoidance in wireless sensor networks. In Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on, pages 1 –5, sept. 2010.

[16] J. Zheng and Myung J. Lee. A comprehensive performance study of ieee 802.15.4. Sensor Network Operations, pages 218–237, 2006.

[17] Ieee standard 802.15.3, wireless medium access controls (mac) and physical layer (phy) specifications for high rate wireless person area networks (wpans), September 2003.

[18] C. Perkins, E. Belding-Royer, and S Das. Ad hoc on-demand distance vector (aodv) routing. RFC 3561, July 2003.

[19] Chipcon as smartrf cc2420 preliminary datasheet (rev 1.2). Website, 2009. http://inst.eecs.berkeley.edu/˜cs150/Documents/CC2420.pdf.

[20] S. Chatterjee and A.S Hadi. Regression Analysis by Example. Wiley Interscience, 4th edition, 2006.

Authors

Chaiyut Jandaeng is a lecturer at the School of Informatics of Walailuk University. He completed the B.Sc and M.Sc. in Computer Science, PSU. He is a Ph.D candidate in Computer Engineering at Prince of Songkla University, Thailand. He was a teacher assistance and guest lecturer in information technology after completing masters, before beginning the doctoral research. Areas of interest are computer networks security, and algorithm and programming technique in resource constrained devices.

Wannarat Suntiamorntut received the Ph.D Degrees from the University of Manchester, UK. She was a research assistant at the School of Computer Science, the University of Manchester for 3 year, while she was doing the Ph.D. program. Since 1999, she has been a lecturer at the computer engineering department at Prince of Songkla University, Hat Yai Thailand and is now assistant professor of computer engineering department. In 2008, she became the Director of Collaborative Research Unit in Wireless Sensor Network (CRU-WSN), a joint collaboration between National Electronics and Computer Technology Center and Prince of Songkla University.

Nittida Elz received her B.Sc. degree in Mathematics from Prince of Songkla University, Thailand. She then received her M.Sc. degrees in Computer Science from Chulalongkorn University, Thailand and The University of Melbourne, Australia, followed by a Ph.D. degree in Computer Engineering from La Trobe University, Melbourne, Australia. Her research interest is network management and security.

Waste Energy Reduction in Wireless Sensor Networks with Packet Scheduling Algorithm

Chaiyut Jandaeng and Wannarat Suntiamontut
Ubiquitous Networked Embedded System (UbiNES),
Department of Computer Engineering,
Faculty of Engineering,
Prince of Songkla University, Thailand
Email: cjundang@jacaranda.in.th

Nittda Elz
Computer System and Network (CSN)
Department of Computer Science,
Faculty of Science,
Prince of Songkla University, Thailand

***Abstract*—Energy is the main constraint that affect to the lifetime of wireless sensor network while packet collision is the one of the reasons of waste energy. This paper review the packet collision avoidance approach and proposes the PSU algorithm in order to reduce packet collision in wireless sensor networks. Greedy algorithm has been applied in scheduling policy to update frame length. We show both mathematical results and network simulation results. The mathematical results show that the PSU algorithm increase throughput 35% but increase delay to 50%, while the simulation result show that the PSU provide throughput less than IEEE 802.15.4 standard but it reduce energy consumption and packet collision.**

***Index Terms*—Waste Energy, PSU Algorithm, Packet Collision,**

I. Introduction

A wireless sensor network is a self-configured network containing numerous small sensor nodes. Each node consists of sensing modules, a processing unit, radio frequency components and power sources. They organize and communicate among themselves in an ad-hoc fashion. The light weighted operating systems (eg. TinyOS, Contiki) may be embedded in sensor nodes. The wireless sensor network technology has been employed in several applications such as health care monitoring system, home control automation and environment monitoring system.

The major sources of waste energy are divided into four reasons [16]. When a transmitted packet is corrupted, it has to be discarded, and follow on retransmissions increase energy consumption. *Packet collision* increases latency as well and energy from packet retransmissions. The *idle listening* is listening to receive possible traffic that is not sent. This is especially true in many sensor network applications. If nothing is sensed, nodes are in idle mode for most of the time. *Overhearing* mean that a node picks up packets that are destined to other nodes. The last reason is *control packet overhead*.

There are many approaches such as increase duty cycle and propose the new MAC protocol to reduce idle listening problem. The energy efficiency for network protocols are proposed to decrease protocol overhead. The broadcast scheduling problem and the approximation approaches are presented in order to reduce packet collision.

The main contribution of this paper is testing the performance of IEEE 802.15.4 standard after we applied the PSU algorithm in order to reduce packet collision and wasted energy. In section II, we summaries network communication modes in IEEE 802.15.4 standard, packet collision, broadcasting scheduling problem, and previous works. Our work, the PSU algorithm, is explained in section III and IV. For our algorithm, we describe definition, algorithm, mathematical results and network simulation results. Discussion and Conclusion are shown in section V and VI.

II. Literature Reviews

A. IEEE 802.15.4 MAC Protocol

The IEEE 802.15.4 standard is to provide a low-power, low-cost, and highly reliable protocol for wireless connectivity among inexpensive, fixes and portable devices. These devices can form a sensor network or a Wireless Personal Area Network (WPAN). In addition, the IEEE 802.15.4 standard consists of two modes: beacon enabled (slotted) and non-beacon modes (unslotted). In beacon enabled modes, communication is synchronized and controlled by a network coordinator, with transmits periodic beacons in order to define the start and the end of superframe. For non-beacon enabled mode, no regular beacons are transmitted. Unslotted CSMA/CA is used as channel access mechanism with back-off technique to random transmission time. If packet collision occurs, Sensor nodes will random the new back-off time. [1] [9]

B. Packet Collision Problem

Wireless sensor network is represented on a undirected graph $G = (V, E)$ where V represents the set of sensor nodes and E represents the set of edges. In the case of $(u, v) \in E$, node u and v are said to be *one hop apart*. Further, if u and v are not one hop apart but there has an intermediate node k such that $(u, k) \in E$ and $(v, k) \in E$, node u and v are said to be *two hops apart*. [12]

There are two types of collisions: [12] direct collision and hidden collision. Let $u, v \in V$ that are one hop apart. If u's transmission time overlaps with v's transmission time, a direct collision will occur. On the other hand, a hidden collision will occur when both u and v transmit data to k at overlapping

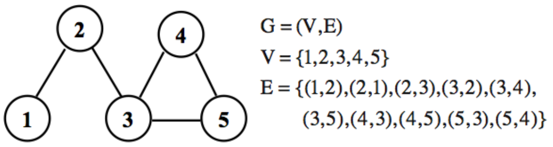


Fig. 1. Network represented with graph theory

times, and u and v are not one hop apart. For the example, node 4 and 5 are direct collision because they share channel on node 3 and be one hop apart. Node 2 and 4 are hidden collision because even they do not share communication channel, but their packet may collide if they send packet to same destination (node 3) at the same time.

C. Broadcast Scheduling Problem (BSP)

In order to decrease packet collisions in a multiple access channels, transmission time is divided into duration times (called frames). Furthermore, each frame is divided into time slots and assigned for transmission nodes. The metrics focus on average delay time, fairness to reduce collisions, and increase throughput.

All neighbored sensor nodes are represented in to scheduling matrix. A schedule matrix, $S = \{s_{ij}\}$, is proposed in order to schedule packet to frame. S refer to a matrix of size NxL, N denotes the number of nodes (columns) while L denotes time slots (row) of the schedule matrix [7], [12]. And s_{ij} is defined in eq. 1

$$s_{ij} = \begin{cases} 1 & \text{if node } i \text{ can send packet in time slot } j \\ 0 & otherwise. \end{cases} \quad (1)$$

The packet scheduling algorithm minimizes frame length (refer to delay time) and maximizes channel utilization (refer to throughput). The several proposed algorithms used graph theory in order to represent network and optimize schedule matrix.

D. Related Works

The proposed algorithms are categorized into four approaches: the simple heuristic search, the heuristic search for NP-complete problem, the finite state machine minimization and the zone-based broadcasting scheduling.

The first approach, wireless networks are represented in the graph theory and schedule matrix. This idea consists of two steps: reducing the row of schedule matrix and attempt to increase throughput with any algorithm. The method reported in [10] proposed the heuristic search to find the optimal schedule matrix that minimizes frame length and maximizes throughputs. This approach is the combination of the tabu search and the greedy algorithm. The tabu search is meta heuristic that applied to find the minimal frame length, while they maximize throughput with the greedy algorithm. The results were compared with HNN-GA [4]. The proposed algorithm did not introduce any shorter frame lengths, but only slightly improve in average delay time. In addition, this algorithm produced higher throughput when the number of nodes are increased.

In [15], finding the optimal schedule matrix in broadcast scheduling problem could be proved to be NP-complete. In order to minimize the frame length and maximize throughput, the approximation methods are proposed to find a optimal schedule matrix. The new approaches base on heuristic search are proposed to find the optimal schedule matrix. The approximation methods were HNN-GA [4], MFA [15], SVC [17], and BSC-NCNN [13]. The minimal frame length from each method produced not much different in value. BSC-NCNN produced the most minimal average time delay, while the combination of neural network (NN) and genetic algorithm (GA) provided the higher throughput.

The new idea is proposed in order to finding packet scheduling algorithm. The schedule matrix is modeled with finite state machine (FSM) [2]. The main idea of other algorithms are generated all time slot first, after that they propose to reduce the unused time slot. In contrast, this method generate the time slot from scratch. This idea combines all nodes that do not cause any collision under concept of maximal compatibles and incompatibles in FSM definition [2]. It lead to result in the minimal time slots are generated. The new schedule matrix produce the minimal frame length and maximal throughput. This algorithm is efficient for all benchmark cases and generated schedules within tight lower bound in negligible time.

Author of [3] proposed the other idea of BSP. They designed a Zone-based broadcasting protocol (ZBP) in order to reduce the cost of broadcasting and alleviate the packet collision phenomenon, this article presents an efficient broadcasting protocol for transmitting a packet from source to a region of all sensor nodes in a WSN. The source could be considered as the first sensor node that receives the query request from sink node and is located in the specified region based on the Cellular-Based Management. This protocol is compared with the traditional flooding operations, experimental results show that the proposed broadcasting protocol reduces the bandwidth and power consumption, avoids the packet collisions, and achieves high success rate of packet delivery.

Although many approaches are presented to minimize packet collision and maximize throughput but their approaches are just the propose idea and it can not be implemented in sensor nodes suddenly beside resources constraints, frequently changes topology and sensor nodes prone to fail. All scheduling algorithm should be implemented in sensor nodes and operate in realtime mode.

III. The PSU Algorithm

A. Problem Statements

Wireless sensor networks frequently change and prone to fail and the centralized paradigm affects from single point of failure and traffic jam around central node. In slotted mode, The coordinator nodes control the traffic channel in order to avoid packet collision, whereas all node must random backoff interval time to grant communication channel in unslotted node. We decide to consider the problem of packet collision under the following the assumptions: 1) All nodes must have the information of 2-hops neighbor nodes and 2) Time synchronization, neighbor discovery, and routing protocol are not concerned in this algorithm

Packet **S**cheduling Algorithm for **U**nslotted IEEE 802.15.4 (shortly called PSU): [7] the waste energy in wireless sensor network can be reduced by scheduling packet to avoid packet collision before forwarding to unslotted IEEE 802.15.4 MAC protocol in lower layer. The objectives of this this algorithm are to improve performances such as throughput maximization and waste energy minimization by reducing packet collision.

However, this paper we simulate this algorithm in grid topology in unslotted mode and slotted node in order to evaluate performance of this algorithm.

B. Algorithm

The PSU is designed base on greedy algorithm by defining *node status, $combine()$, and $match()$* to optimize the frame length.

- *node status* represents nodes in color status for each time slot. In any time slot, black colored node can send any packet under guarantee of collision free while for, white colored node, if any node requests to send packet in this time slot, its packet maybe collide with other packet. Gray colored node is initial status that no guarantee for packet collision. Howover gray node can be changed to other status by $combine()$ and $match()$.
- *combine()* is used to reduce frame length by combining two time slots. While, the two time slots must be tested with $match()$ before combination. They will be combined in case they valid with $match()$ testing. In combination procedure, each status of node in time slot will be changed to other status with (2). Let A, B, and R denote time slots and A_i, B_i, and R_i are node status i^{th} in time slot of A, B, and R; N denotes set of node and $R = combine(A, B)$.

$$\forall_{i \in N}, R_i = \begin{cases} A_i & \text{if } B_i \text{ is gray} \\ B_i & otherwise. \end{cases} \tag{2}$$

- *match()* The $combine()$ is node status replacement in time slot. Two time slots must be validated with $match()$. All node status must accept condition in (3) before combination.

$$\begin{aligned} match(A, B) \Leftrightarrow & \forall_{i \in N}, A_i = grey \\ & \vee A_i = black \wedge B_i = gray \\ & \vee A_i = white \wedge B_i \neq black \end{aligned} \tag{3}$$

For, PSU algorithm groups all time slots with $combine()$ and $match()$ in order to provide minimal frame length and maximal gray node under greedy concept until they can not find any two time slots that valid with $match()$. As shown below.

Algorithm 1 finding minimal frame length

```
$SCH = \{TS_i\}$ and $1 < i <= L$
loop
    if $TS_a = max_a(SCH, GRAY)$ and $TS_b = max_b(SCH, GRAY)$ and $TS_a \neq TS_b$ and $match(TS_a, TS_b) = TRUE$ then
        $R = combine(TS_a, TS_b)$
        $SCH = SCH - \{TS_a, TS_b\}$
        $SCH = SCH \bigcup \{R\}$
    end if
    if can not get any $TS_a$ and $TS_b$ that matched conditions then
        return SCH
    end if
end loop
```

Finally, the gray nodes in each time slot must be eliminated and replaced with black status to increase throughput. As shown below.

Algorithm 2 throughput increasing

```
$initSCH$ is initial scheduling matrix
for $n \in Nodes$ do
    $TS = \{TS_i \in SCH | TS_{ni} = GRAY\}$
    for $t \in TS$ do
        if $match(t, initSCH_n)$ then
            $t = combine(t, initSCH_n)$
        end if
    end for
end for
return SCH
```

C. Mathematical Result

There are two performance metrics: average delay and throughput. Both of them are calculated from the schedule matrix, S size NxL. When N is number of node, L is frame length, and s_{ij} is the status of node in each time slot.

Average delay depends on the number of time slot. and indicates delay of each node for the next round of packet sending. When we minimize the frame length, it brings to the minimal average delay.

Average delay (τ)

$$\tau = \frac{L}{N} \sum_{i=1}^{N} \left(\frac{1}{\sum_{j=1}^{L} s_{ij}} \right) \tag{4}$$

The average delay is calculated from eq.4 and shown in fig 2

On the other hand, throughput is number of reserved time slots that be assigned to sensor nodes and indicate the performance of network per frame in TDMA.

Throughput (σ)

$$\sigma = \sum_{i=1}^{N} \sum_{j=i}^{L} s_{ij} \tag{5}$$

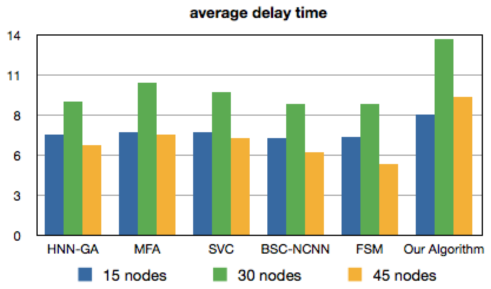


Fig. 2. Mathematical Results: average delay

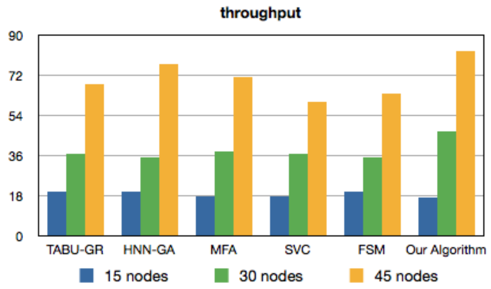


Fig. 3. Mathematical Results: throughput

Throughput is calculated from eq.5 and shown in fig 3

Scheduling algorithm must trade-off between average delay time and throughput. Number of time slot increase throughput but also decrease average delay. From mathematical result, the PSU algorithm maximize throughput by 35% while average increase to 50%. In order to archive the better performance, this paper simulate this algorithm with network simulation and observe throughput, energy consumption, end-to-end delay and packet collision.

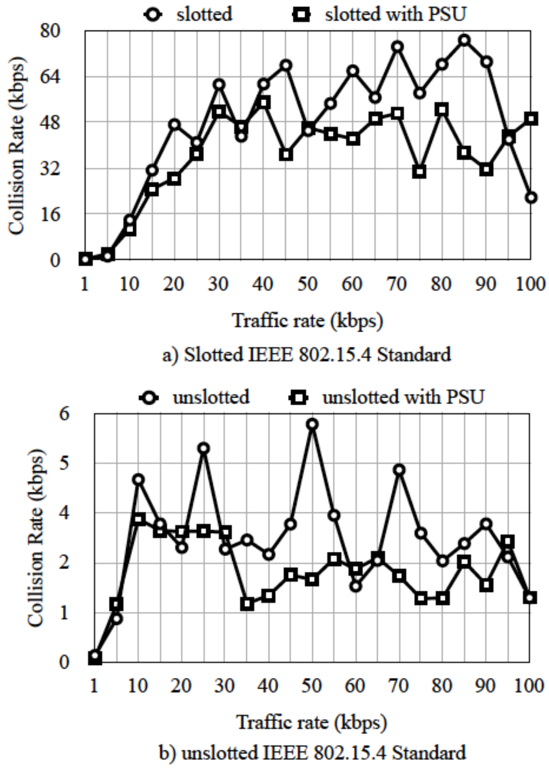
IV. Simulation Results

A. Scenarios

In order to archive PSU performance in packet collision and waste energy. We simulate PSU algorithm in logical link layer that schedule packet before forwarding to mac layer and independent from routing layer. The 25 sensor nodes are placed in grid topology sized 5x5 consecutively. All sensor nodes send 40 bytes CBR message over UDP and route with AODV protocol to sink nodes (id 0). CBR traffic rate varies from 1,5,10,15,...,and 100 kbps. Each nodes initiate 24kJ energy and take power 47.28mW for sending packet and 47.16mW for receiving packet. Transmission range is 9m. All scenarios simulate for 1000s.

B. Collision rate

Because the PSU algorithm is the collision avoidance, then packet collision still exists. The collision rate measure from the total of number collision packet (in bits) divides by simulation time. The waste energy depend on packet length then proposing the collision rate in term of bit per second is reasonable more than number packet per second.



a) Slotted IEEE 802.15.4 Standard

b) unslotted IEEE 802.15.4 Standard

Fig. 4. Simulation result: Packet Collision

We schedule all packets before passing to MAC layer to reduce the number of packet in a duration time by delay some packet. The packet collision is decreased as show in fig. 4. The packet collision is increased when traffic rate increase because there are many packet in MAC layer. Although there are many packets in the same duration time, our algorithm still reduce the packet collision up to 10% in slotted mode and 20% in unslotted mode.

C. Average energy usage rate

The average energy usage is energy consumption per node during a simulation run. This paper we determine the energy usage for transmission and receiving only while waste energy that cause from packet collision indicate by number of collision bits in fig. 4.

The energy usage as shown in fig. 5 are the energy per second per node. There are increasing of packet when CBR traffic rates are increased and bring to the more energy consumption. Because of decreasing of packet collision, some of average energy usages are decreased and they are decreased up to 15.6% in slotted mode and 8.6% in unslotted mode.

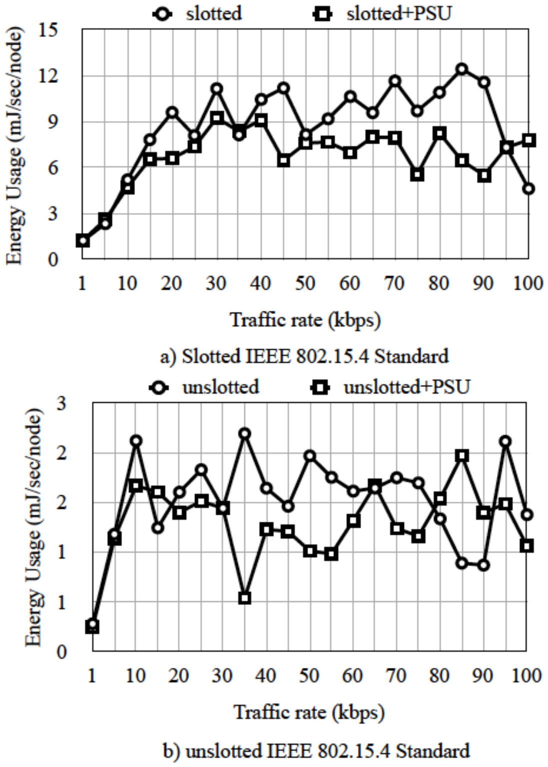


Fig. 5. Simulation result: Energy Usage

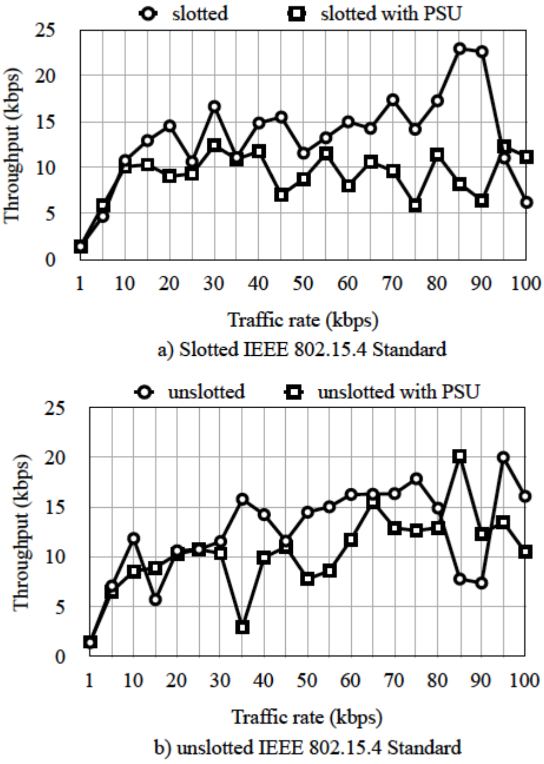


Fig. 6. Simulation result: Throughput

D. Throughput

The throughput of a node is measured by counting the total number of data packets successfully received at the node, and computing the number of bits received, which is finally divided by the total simulation runtime.

Throughput of slotted and unslotted modes increase when traffic rate increasing but throughput are not more than 25 kbps shown in fig 6. We compare MAC standard and MAC with PSU in term of throughput both modes. We found that growth trend lines of our algorithm similar to standard but the PSU algorithm decrease throughput to 3% in slotted mode and 6% in unslotted modes.

V. Discussion

A. The effect of packet scheduling in LLC

We schedule all packets from upper layer before passing to MAC layer to reduce the number of packet in a duration time in order to reduce packet collision that is a one of reason of waste energy. But we cannot scheduling the packet in MAC layer such as neighbor discovery packets and acknowledged packets. Because of simulating topology is static, the packet collision is clearly divided into two phases: network establish phase and data collection phase.

Most of collided packets in network establish phase are neighbor discovery packets and acknowledged packets and result in the network establishment time in standard mode is more than the standard with PSU algorithm. Then they are bring to end-to-end delay time increasing. In addition, communication in MAC layer is broadcast. When a packet is sent, there are many are replied especially neighbor packets that bring to congestion and contribute to packet collision.

After network establish phase, all source node periodically send CBR packets in data collection phase. Most of collied packets are CBR packets that cause throughput minimization.

However, packet scheduling in LLC have advantage in flexibility. The varieties of MAC protocols and routing protocols do not effect to scheduling algorithm. We must precise in types of packet from upper layer in order to improve performances.

B. The effect of CBR traffic rate

Traffic rate indicates the number of packet that be injected into network in the period of time. The increasing of incoming packet results in higher probability of collision. For example, the number of packet in 40 byes CBR traffic are sent on 1,2,.., and 5 kbps are 4,7,10,13,and 16 packets. Every application packet cause congested control packet that will be overhead of network. When application packet is sent in to routing layer in the first time. The routing protocol start by sending route request packet in order to find network path. While MAC layer broadcast neighbor discovery packet in order to find and exchange packet with neighbor nodes. Hence, increasing of application packet causes the traffic congestion and causes packet collision.

VI. Conclusion

We schedule all packets from upper layer before passing to MAC layer to reduce the number of packet in a duration time in order to reduce packet collision that is a one of reason of waste energy.

The mathematical results show that the PSU algorithm maximize number of empty slots for sensor nodes. But this algorithm provide the higher delay that indicated by the number of time slots.

When we experiment this algorithm in grid topology with network simulator, we found that the packet collision is reduced and bring to reduction of average energy consumption.

However, throughput from our algorithm lower that the result from standard. Many CBR packet are dropped in MAC layer because of full queue.

References

[1] Ieee standard 802.15.4: Wireless medium access control (mac) and physical layer (phy) specification for low-rate wireless personal area networks (lr-wpans).

[2] Imtiaz Ahmad, Buthaina Al-Kazemi, and A. Shoba Das. An efficient algorithm to find broadcast schedule in ad hoc tdma networks. *J. Comp. Sys., Netw., and Comm.*, 2008:1–10, 2008.

[3] Chih-Yung Chang, Kuei-Ping Shih, and Shih-Chieh Lee. Zbp: A zone-based broadcasting protocol for wireless sensor networks. *Wireless Personal Communications*, 33:53–68, 2005. 10.1007/s11277-005-6327-3.

[4] N. Funabiki and Y. Takefuji. A parallel algorithm for broadcast scheduling problems in packet radio networks. *Communications, IEEE Transactions on*, 41(6):828 –831, jun. 1993.

[5] Rong gang Bai, Yu gui Qu, Yang Guo, and Bao hua Zhao. An energy-efficient tdma mac for wireless sensor networks. *Asia-Pacific Service Computing Conference, The 2nd IEEE*, pages 69 –74, dec. 2007.

[6] Chaiyut Jandaeng, Wannarat Suntiamorntut, and Nittida Elz. Packet scheduling to minimize packet collision and maximize throughput for unslloted ieee 802.15.4. *Intelligent Network and Computing, 2010. ICINC'10. International Conference on*, pages xx–xx, Nov. 2010.

[7] Chaiyut Jandaeng, Wannarat Suntiamorntut, and Nittida Elz. Throughput improvement of collision avoidance architecture in wireless sensor networks. *Wireless Communications, Networking and Mobile Computing, 2010. WiCom '10. 6th International Conference on*, pages xx–xx, sep. 2010.

[8] In Taek Leem, M. Wu, and Chonggun Kim. A mac scheme for avoiding inter-cluster collisions in wireless sensor networks. *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, 1:284 –288, feb. 2010.

[9] Xuedong Liang and Ilangko Balasingham. Performance analysis of the ieee 802.15.4 based ecg monitoring network. *Wireless and Optical Communications, 7th IASTED International conference on*, pages 99 –104, May. 2007.

[10] Y. Peng, B.H. Soong, and L. Wang. Broadcast scheduling in packet radio networks using mixed tabu-greedy algorithm. *Electronics Letters*, 40(6):375 – 376, mar. 2004.

[11] Prasan Kumar Sahoo, Jang-Ping Sheu, and Yu-Chia Chang. Performance evaluation of wireless sensor network with hybrid channel access mechanism. *Journal of Network and Computer Applications*, 32(4):878 – 888, 2009.

[12] S. Salcedo-Sanz, C. Bousono-Calzon, and A.R. Figueiras-Vidal. A mixed neural-genetic algorithm for the broadcast scheduling problem. *Wireless Communications, IEEE Transactions on*, 2(2):277 – 283, mar. 2003.

[13] Haixiang Shi and Lipo Wang. A hybrid neural network for optimal tdma transmission scheduling in packet radio networks. volume 5, pages 3210 – 3213 vol. 5, jul. 2005.

[14] Hisa-Aki Tanaka, Hiroya Nakao, and Kenta Shinohara. Self-organizing timing allocation mechanism in distributed wireless sensor networks. *IEICE Electronics Express*, 6(22):1562–1568, 2009.

[15] G. Wang and N. Ansari. Optimal broadcast scheduling in packet radio networks using mean field annealing. *Selected Areas in Communications, IEEE Journal on*, 15(2):250 –260, feb. 1997.

[16] Wei Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3):493 – 506, jun. 2004.

[17] Jaehyun Yeo, Heesoo Lee, and Sehun Kim. An efficient broadcast scheduling algorithm for tdma ad-hoc networks. *Computers & Operations Research*, 29(13):1793 – 1806, 2002.

Packet Scheduling to Minimize Packet Collision and Maximize Throughput for Unslotted IEEE 802.15.4 Networks

Chaiyut Jandaeng and Wannarat Suntiamorntut
Department of Computer Engineering
Prince of Songkla University
Had Yai, Songkla, Thailand 90112
Email: {s4910130006, wannarat.s}@psu.ac.th

Nittida Elz
Department of Computer Science
Prince of Songkla University
Had Yai, Songkla, Thailand 90112
Email: nittida.n@psu.ac.th

Abstract—**Energy is the main constraint that affect to the lifetime of wireless sensor network while packet collision is the one of the reasons of waste energy. This paper proposes that the PSU algorithm can reduce packet collision in unslotted IEEE 802.15.4. Greedy algorithm has been applied in scheduling policy to update frame length. This algorithm has been introduced aiming for minimal packet collision and maximal throughput in logical link control in order to reduce packet before forward to MAC layer. We improved the performance of PSU algorithm by simulator called NS2. The simulation results from NS2 show that our algorithm gives a high throughput and low packet collision and it can reduce the waste of energy.**

Index Terms—**PSU algorithm, Packet Collision, Throughput, Unslotted IEEE 802.15.4 Standard**

I. Introduction

Wireless sensor network has been used in many applications. The devices of this system operate by using batteries thus energy efficiency is very important issue [1]. Waste energy is required to be reduces much as possible. The cause of waste energy comes from packet collision while devices concurrently attempt to transmit the packet.

The main contribution of this paper is simulation of packet scheduling algorithm in unslotted IEEE 802.15.4 environment. We considered throughput maximization and packet collision minimization that lead to reduce waste energy for various scenarios such as nodes density and traffic rates. We described PSU algorithm in [2] and we archive network performance in this paper.

Section II of this paper offers a review of IEEE 802.15.4 standard, we start with reviewing the background works such as packet scheduling for collision avoidance in term of minimize frame length and maximize throughput and conclude the PSU algorithm in section III, In particular, performance issues such as throughput, packet collision, and waste energy will be explained in section IV and we will also show simulation result and discussion in section V. In the last section, we conclude this paper and propose tune up approach in order to improve performances are presented .

II. Overview of IEEE 802.15.4 standard

A. IEEE 802.15.4 Standard

Author of [3] reviewed and concluded that the goal of the IEEE 802.15.4 standard is to provide a low-power, low-cost, and highly reliable protocol for wireless connectivity among inexpensive, fixes and portable devices. These devices can form a sensor network or a Wireless Personal Area Network (WPAN). In addition, the IEEE 802.15.4 standard consists of two modes: beacon enabled (slotted) and non-beacon modes (unslotted).

In beacon enabled modes, communication is synchronized and controlled by a network coordinator, with transmits periodic beacons in order to define the start and the end of superframe. The superframe consists of active and inactive periods, the active part is divided into 16 equally sized slots and consists of two groups: the contention access period (CAP) and and optional contention access free period (CFP). In CAP, slotted CSMA/CA is used channel access mechanism, where the backoff slot aligns with the beginning stage of beacon transmission. In CFP, time slots are assigned by the coordinator, devices which have been assigned specific time slots can transmit packets in this period. All communications must take place during the activate part. In the inactivate part, devices can power down to conserve energy.

In non-beacon enabled mode, no regular beacons are transmitted. Unslotted CSMA/CA is used as channel access mechanism. When any network device wishes to transmit packets, it will wait for a random number of backoff slots, which chosen uniformly between 0 and $2^{BE}-1$, where BE is the backoff exponent. The default minimal value (macMinBE) is set to be 3, it will check whether the channel is idle or not. If so, the network device will begin to transmit data packet. In the other hand, if channel is nor idle, BE will be incremented by 1, and the network device backoff again with the new value. Theses procedure will be repeated until the number of BE exceeds the maximum number of backoff exponent ($aMaxBE$), which is set to be 5. Similarly the number of iterations is also limited by NB, the maximum number of backoff, before declaring a channel access failure. The default value of NB is 4.

More detailed description of the IEEE 802.15.4 can be found in [4]

B. Problem Statement

Wireless sensor networks frequently change and prone to fail and the centralized paradigm affects from single point of failure and traffic jam around central node. In star topology, there are coordinator nodes that control traffic channel in order to avoid packet collision, whereas all node must random backoff interval time to grant communication channel in mesh network. Therefore, we decide to consider the problem of packet collision bases on decentralized scheme that communicate in non-beacon modes. All sensor nodes must schedule its packet with itselves under the following the assumption:

- All packets communicate via unslotted IEEE 802.15.4 MAC protocol
- Packet scheduling is the function of source node
- All nodes must have learnt the information of 2-hops neighbor nodes
- Time synchronization, neighbor discovery, and routing protocol are not concerned in this paper

The problem can then be stated as follows.

Packet Scheduling on Unslotted IEEE 802.15.4 (PSU) Problem: waste energy in wireless sensor network can be reduced by scheduling packet to avoid packet collision before forwarding to unslotted IEEE 802.15.4 MAC protocol in lower layer. The objectives of this paper are to improve performances such as throughput maximization and waste energy minimization by reducing packet collision.

C. Background Work

The packet scheduling algorithm minimizes frame length (refer to delay time) and maximizes channel utilization (refer to throughput). The several proposed algorithms used graph theory in order to represent network and optimize schedule matrix. The proposed algorithms are categorized into three approaches: the simple heuristic search, the heuristic search for NP-complete problem, and the finite state machine minimization.

The first approach, wireless networks are represented in the graph theory and schedule matrix. This idea consists of two steps: reducing the row of schedule matrix and attempt to increase throughput with any algorithm. The method reported in [5] proposed the heuristic search to find the optimal schedule matrix that minimizes frame length and maximizes throughputs. This approach is the combination of the tabu search and the greedy algorithm. The tabu search is meta heuristic that applied to find the minimal frame length, while they maximize throughput with the greedy algorithm. The results were compared with HNN-GA [6]. The proposed algorithm did not introduce any shorter frame lengths, but only slightly improve in average delay time. In addition, this algorithm produced higher throughput when the number of nodes are increased.

In [7], finding the optimal schedule matrix in broadcast scheduling problem could be proved to be NP-complete. In order to minimize the frame length and maximize throughput, the approximation methods are proposed to find a optimal schedule matrix. The new approaches base on heuristic search are proposed to find the optimal schedule matrix. The approximation methods were HNN-GA [6], MFA [7], SVC [8], and BSC-NCNN [9]. The minimal frame length from each method produced not much different in value. BSC-NCNN produced the most minimal average time delay, while the combination of neural network (NN) and genetic algorithm (GA) provided the higher throughput.

The new idea is proposed in order to finding packet scheduling algorithm. The schedule matrix is modeled with finite state machine (FSM) [10]. The main idea of other algorithms are generated all time slot first, after that they propose to reduce the unused time slot. In contrast, this method generate the time slot from scratch. This idea combines all nodes that do not cause any collision under concept of maximal compatibles and incompatibles in FSM definition [11]. It lead to result in the minimal time slots are generated. The new schedule matrix produce the minimal frame length and maximal throughput. This algorithm is efficient for all benchmark cases and generated schedules within tight lower bound in negligible time.

III. Our Packet Scheduling Algorithm

Although many approaches are presented to minimize packet collision and maximize throughput but their approaches are just the propose idea and it can not be implemented in sensor nodes suddenly beside resources constraints, frequently changes topology and sensor nodes prone to fail. All scheduling algorithm should be implemented in sensor nodes and operate in realtime mode.

In [2], author proposed the **P**acket **S**cheduling Algorithm for **U**nslotted IEEE 802.15.4 (shortly called PSU) to find minimal frame length that provide high throughput. The PSU is designed base on greedy algorithm by defining *node status, $combine()$, and $match()$* to optimize the frame length.

- *node status* representes nodes in color status for each time slot. In any time slot, black colored node can send any packet under guarantee of collision free while for, white colored node, if any node requests to send packet in this time slot, its packet maybe collide with other packet. Gray colored node is initial status that no guarantee for packet collision. Howover gray node can be changed to other status by $combine()$ and $match()$.
- *combine()* is used to reduce frame length by combining two time slots. While, the two time slots must be tested with $match()$ before combination. They will be combined in case they valid with $match()$ testing. In combination procedure, each status of node in time slot will be changed to other status with (1). Let A, B, and R denote time slots and A_i, B_i, and R_i are node status i^{th} in time slot of A, B, and R; N denotes set of node and $R = combine(A, B)$.

$$\forall_{i \in N}, R_i = \begin{cases} A_i & \text{if } B_i \text{ is gray} \\ B_i & otherwise. \end{cases} \quad (1)$$

- *match()* The $combine()$ is node status replacement in time slot. Two time slots must be validated with $match()$. All node status must accept condition in (2) before combination.

$$\begin{aligned} match(A,B) \Leftrightarrow & \forall_{i \in N}, A_i = grey \\ & \vee A_i = black \wedge B_i = gray \\ & \vee A_i = white \wedge B_i \neq black \end{aligned} \quad (2)$$

For, PSU algorithm groups all time slots with $combine()$ and $match()$ in order to provide minimal frame length and maximal gray node under greedy concept until they can not find any two time slots that valid with $match()$. Finally, the gray nodes in each time slot must be eliminated and replaced with black status to increase throughput. All detail explain in [2].

From mathematical result [2], The PSU algorithm maximize throughput by 35%. In order to archive the better performance, this paper simulate this algorithm with network simulation and observe throughput, and packet collision (by product is waste energy).

IV. Performance Evaluation

A. Effects of Traffic Rate (λ)

Traffic rate (λ) indicates the number of packet that be injected into network in the period of time. The increasing of incoming packet result in higher probability of collision that measured by (3). Let l denotes packet length in application layer.

$$\#pkt = \lceil \frac{\lambda}{l * 8} \rceil \quad (3)$$

For example, the number of packet in 40 byes CBR traffic are sent on 1,2,.., and 5 kbps are 4,7,10,13,and 16 packets. Every application packet cause heavy control packet that will be overhead of network. Sending packet mechanism consists of passing application packet to lower layer. Then routing protocol start by sending route request packet in order to find route path. While MAC layer broadcast ARP packet in order to communicate with neighbor nodes. Hence, increasing of application packet causes the traffic congestion and causes packet collision in all layer, especially MAC in layer.

B. Effects of Nodes Density

MAC protocol is broadcast communication. When sensor node send a packet, all neighbor nodes will get packet and determine to forward or drop. In routing phase, route packet are forwarded. The number of copied packets depends on the number of neighbor nodes act as route node or node density that make the probability of packet collision. Therefore, collision rate direct variate with node density especially control packet. When node density increases, the probability of collision will be also increased too. This paper varies node density from 0.01, 0.02, and 0.03 by random place 24 sensor nodes in 30x30, 40x40, and 50x50 and fixes sink node in the middle of sensor field in order to observe relation between the performance issues and nodes density.

C. Performance Issues

1) Throughput (τ) : It is a measure of the amount of successful data transmitted in a unit period of time (second). Considering the low data rates and throughputs supported by IEEE 802.15.4, the throughput is measured in total bits received per second (bps), note that this metric only measures the total data throughput and ignoring all other overhead, over the network. The throughput of a node is measured by counting the total number of data packets successfully received at the node, and computing the number of bits received, which is finally divided by the total simulation runtime. The throughput of the network is finally defined as the average of the throughput of all nodes involved in data transmission.

Therefore, throughput (τ) can be stated as:

$$\tau = \frac{\sum bit_{rx}}{t} \quad (4)$$

Similarly the percentage of average throughput of sensor node ($\%\Omega$) can be defined as:

$$\%\Omega = \frac{\sum_{i=1}^{N} \tau_i}{\lambda.N} x100 \quad (5)$$

2) Packet Collision Rate (ζ): Packet Collision (ζ) is a measure of the amount of data that is dropped in a unit period of time (second). All sensor nodes share a single channel using a multiple access protocols. Packet collision occurs when the packet transmission lead to a time overlap of two or more packet receptions. High node density has higher possibility of collision than low node density, because there are more nodes sharing the same communication channel. The packet collision is measured by counting the total of collided packet and divide by the total simulation runtime that is shown below in (6).

$$\zeta = \frac{\sum col_{pkt}}{t} \quad (6)$$

3) waste energy (ξ): Packet collision is a one of cause of waste energy (ξ). Both source and destination consume energy for packet transmission. In case, any packet collides, it will be required to retransmit and lead to consuming the sane amount of energy again. There are varieties types (differences of length) of packet collision such as spreading MAC packet, route packet, and application packet. The energy consumption depend on transmission time (packet length). Therefore, waste energy from packet collision is measured by summation of energy for sending and receiving collided packet as shown in (7) when e_{tx} and e_{rx} are energy for transmitting and receiving packet .

$$\xi = \sum^{\#col_{pkt}} (e_{tx} + e_{rx}) \quad (7)$$

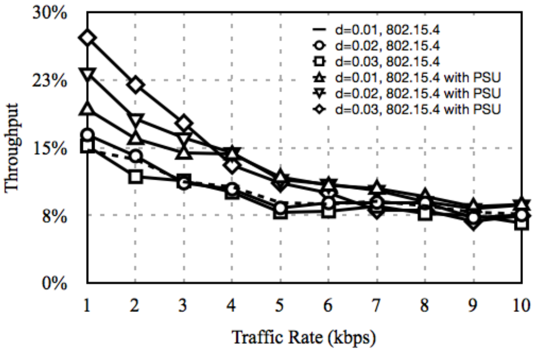


Fig. 1. relation of throughput and traffic rate

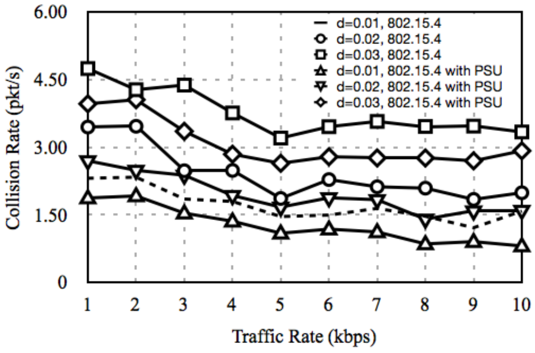


Fig. 2. relation of collision rate and traffic rate

V. Simulation Results

In order to archive PSU performance in throughput, packet collision, and waste energy, we simulate PSU algorithm in logical link layer that schedule packet before forwarding to mac layer and independent from routing layer. The sink node is placed in the middle of sensor field and 24 sensor nodes are randomly placed in 30x30, 40x40, and 50x50 sqm. They communicate via unslotted IEEE 802.15.4 standard and route with AODV routing protocol. All sensor nodes send 40 bytes CBR message over UDP to sink nodes. Traffic rate of CBR varies from 1,2,3,...,and 10 kbps. Each nodes initiate 24kJ energy and take power 47.28mW for sending packet and 47.16 mW for receiving packet. Transmission range is 10m. All scenario simulate for 1000s. The results are shown and discussed below.

A. Throughput

In fig. 1, throughput is nearly 15% of transmitted packets with 1kbps CBR traffic rate on node density 0.03 and slightly decrease when CBR traffic is increased. When we increase CBR traffic rate bring to increasing of application packet and packet in other layers. The number of packet increase cause more dropped (because of queue full, collision, and others) and decreasing of throughput as shown in fig. 1. After we integrate PSU in logical link layer, we found that throughput is up to 27% of 1kbps CBR traffic rate. The PSU algorithm maximize throughput equal and more than traditional IEEE 802.15.4 standard. Moreover, we found that the PSU algorithm is a optimal solution for low traffic. When traffic rate increases, the PSU algorithm can increase throughput, however it is overcome low traffic rate.

In node density observation, throughput increases along with node density increases because sensor nodes directly send packet to destination. In addition, hop count of high density is less. This bring to reducing of control packet in lower layer. It can be concluded that for dense nodes even packet collision increases throughput still can be increased.

TABLE I
Coefficient of correlation between packet collision and waste energy

	Node Density		
	0.01	**0.02**	**0.03**
802.15.4 only	0.9826	0.9909	0.9874
802.15.4 with PSU	0.9986	0.9963	0.9974

B. Packet Collision

The result in fig. 2 shows packet collision when increase application packet (in the term of CBR traffic rate). When incoming packet increases, packet are highly dropped because there are some collision occurrence. We consider in packet collision and found that packet collision is inverse growth with traffic rate because control packets are dropped by any other reason beside packet collision. After we compare number of dropped packet collision caused by other reasons, we found that packets are dropped because of fully queue and timeout of routing packet more than there are drop because of packet collision. However, packet collision still exist in network. After we integrate IEEE 802.15.4 standard with PSU algorithm, we found that the integrated standard with PSU algorithm lead to reduce packet collision more than pure standard. The PSU algorithm reduces packet collision on 0.01 density average 27% compare with pure standard.

When node density increases, packet collision is increased because there are many neighbor nodes, therefor number of broadcast packet are increased as show in fig. 2. The PSU algorithm can reduce packet collision although the traffic is more congestion such as collision decrease up to 18% in 0.02 and 0.03 density.

C. Waste energy

Trend line of waste energy in fig. 3 similar to packet collision chart but there are some difference because the different of types and number of collided packet. Most collided packets are control packet such as acknowledge, ARP, and routing message etc. Each message take the difference transmission time lead to waste energy consumption. Although their trend

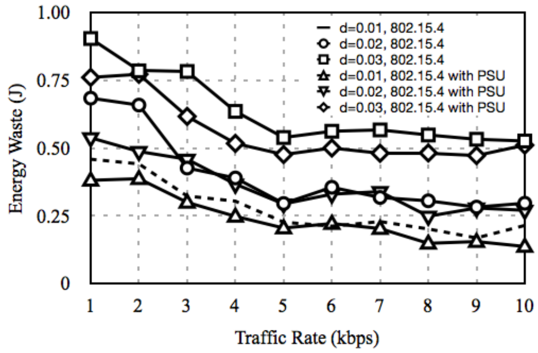


Fig. 3. relation of waste energy rate and traffic rate

line are difference, we determine in coefficient of correlation between collision rate and waste energy. From table I, we found that the correlation between collision rate and waste energy more than 0.98 and close to 1.00. We conclude that the correlation between collision rate and waste energy is the degree to which there is a linear relationship between them. In other words, the trend of two performances are not difference and can be said that it growth in the same trend.

D. Discussion

The increasing of application packet result in increasing of network energy consumption because all CBR packets require route packet, MAC packet, and other control message to transmit packet to destination. All of packets consume the energy in transmission and operation. Although our algorithm reduce waste energy (by collided packet reduction), the network energy can not be reduced.

The PSU algorithm cause side effect on network energy consumption. Although waste energy is decreased, the network energy consumption after adding PSU algorithm take more than the standard protocol. The importance reason is increasing of dropped packet because of packet scheduling. Some urgent control packets such as routing packet etc., are dropped. Source node will wait for acknowledge packet. If it does not reply within timeout, source node will retransmit packet again. From simulation result, we found that dropped packet is increased when traffic rate is increased and PSU algorithm is integrated.

All nodes in network process together although there are not neighbor nodes. Some nodes do not wish to send packet in its time slot while some node has urgent packet that must be sent. Topology maybe change. All limitations come from the pre-deployment of schedule algorithm. Moreover, there are varieties of packet length, frame length ought to change automatically. From wireless sensor networks behavior, they require the adaptive algorithm that can be change schedule policy with themselves to suite with network environment in each time.

VI. Conclusion

The PSU algorithm can maximize throughput overtake IEEE 802.15.4 both high and low CBR traffic. Especially, for the low traffic, it can improve throughput more than in high traffic. Moreover, the PSU algorithm can reduce packet collision that bring to reducing in waste energy. Although waste energy is reduced, network energy consumption increases because of packet retransmission. In addition, some time slots idle because there are no packet outgoing while some urgent packet is dropped because it is not in its time slot.

In node density perspective, it can be concluded that the increasing of neighbor nodes has two sides both advantage and disadvantage. The advantage is throughput is increased because of the lower hop count decrease which lead to number of routing packet and MAC packet. That disadvantage is the increasing of number of neighbor nodes cause increasing of broadcast packet.

The next step of PSU algorithm is improving the scheduling policy. The scheduling policy will be composed of prediction model to update frame length automatically and QoS model for scheduling any urgent packet in order to reduce retransmitted packets. All operations ought to operate in realtime mode.

References

[1] Yick, J., B. Mukherjee, and D. Ghosal, *Wireless sensor network survey*. Computer Networks, 2008. 52(12): p. 2292-2330.

[2] Jandaeng, C. , Suntiamorntut, W., and Elz, N., Throughput Improvement in Packet Collision Avoidance for Wireless Sensor Networks, in Wireless Communications, Networking and Mobile Computing, 2010. WiCom 10. 6th International Conference on, 2010, pp. xx-xx.

[3] Liang, X. , and Balasingham, I., Performance Analysis of The IEEE 802.15.4 Based ECG Monitoring Network, in Wireless and Optical Communications. 7th IASTED International Conference on, 2007.

[4] IEEE Standard 802.15.4: Wireless medium access control (MAC) and physical layer (PHY) specification for low-rate wireless personal area networks (LR-WPANs) (New York:IEEE, 2003)

[5] Peng, Y., B.H. Soong, and L. Wang, *Broadcast scheduling in packet radio networks using mixed tabu-greedy algorithm*. Electronics Letters, 2004. 40(6): p. 375-376.

[6] Funabiki, N.T., Y.;, *A parallel algorithm for broadcast scheduling problems in packet radio networks*. Communications, IEEE Transactions on, 1993. 41(6): p. 828 - 831

[7] Wang, G. and N. Ansari, *Optimal broadcast scheduling in packet radio networks using mean field annealing*. Selected Areas in Communications, IEEE Journal on, 1997. 15(2): p. 250-260.

[8] Yeo, J., H. Lee, and S. Kim, *An efficient broadcast scheduling algorithm for TDMA ad-hoc networks*. Computers & Operations Research, 2002. 29(13): p. 1793-1806.

[9] Haixiang, S. and W. Lipo. *A hybrid neural network for optimal TDMA transmission scheduling in packet radio networks*. in Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on. 2005.

[10] Ahmad, I., B. Al-Kazemi, and A.S. Das, *An efficient algorithm to find broadcast schedule in ad hoc TDMA networks*. J. Comp. Sys., Netw., and Comm., 2008. 2008: p. 1-10.

[11] Imtiaz Ahmad, A.S.D., *A heuristic algorithm for the minimization of incompletely specified finite state machines*. Computers and Electrical Engineering, 2001: p. 159-172.

Throughput Improvement of Collision Avoidance in Wireless Sensor Networks

C. Jandaeng and W. Suntiamontut
Department of Computer Engineering
Prince of Songkla University,PSU
Songkla, THAILAND
Email: {s4910130006,wannarat.s}@psu.ac.th

Nittida Elz
Department of Computer Science
Prince of Songkla University,PSU
Songkla, THAILAND
Email: nittda.n@psu.ac.th

***Abstract*—The packet collision is one of the causes of wasted energy consumption in wireless sensor networks. This paper describes and analyzes the behavior of packet collisions in wireless sensor networks by simulations. The problems of packet collision include both direct and hidden collisions, and are classified by layers architecture: application, routing, and MAC. Moreover, we review and analyze the existing scheduling algorithms, and also propose a new algorithm of packet collision avoidance in wireless sensor network environments. The result shows that our proposed algorithm provides the better throughput compared with existing works.**

***Index Terms*—packet collision, throughput, wireless sensor network.**

I. Introduction

A wireless sensor network is a self-configured network containing numerous small sensor nodes. Each node consists of sensing modules, a processing unit, radio frequency components and power sources. They organize and communicate among themselves in an ad-hoc fashion. The light weighted operating systems (eg. TinyOS, Contiki) may be embedded in sensor nodes. The wireless sensor network technology has been employed in several applications such as health care monitoring system, home control automation and environment monitoring system [1].

All sensor nodes share a single channel using a multiple access protocols. The packet transmission may lead to a time overlap of two or more packet receptions, called collisions. The packet collision problem causes packets loss at destination, packet retransmission, decreasing throughputs, increasing delay/latency, and wasting energy. Although there are many solutions, which has been proposed in many levels of network, the collision still exists. The appropriate solution is combining techniques from all layers such as sleep/wake up scheduling of sensor node in physical layer, random back-off time in MAC protocol, allocation of time slot for all messages to avoid collisions, and schedule aggregation time in application layer to reduce number of query requests and data replies.

In order to decrease packet collisions in a multiple access channels, transmission time is divided into duration times (called frames). Furthermore, each frame is divided into time slots and assigned for transmission nodes. The metrics focus on average delay time, fairness to reduce collisions, and increase throughput.

This paper reviews and analyzes the problems caused by packet collision in wireless sensor networks in terms of collision ratio and throughput, and also discusses about possible solutions to avoid the collisions in wireless sensor networks with the IEEE 802.15.4 Standard. We also propose our new possible solution to avoid packet collision aiming for throughput improvement.

In order to show that the new algorithm provides batter throughput and perform appropriately with wireless sensor networks, we must implement the algorithm and simulate in computer first. After that we can experiment the implemented algorithm in wireless sensor node environment.

The remainder of the paper is organized as follows. In section II, the background and behavior of packet collision are described and analyzed. Next, we review existing algorithm that proposed to avoid packet collision and introduce our algorithm. Finally, we discuss and outline possible future work in section IV.

II. PACKET COLLISION IN WIRELESS NETWORKS

A. Network collision representation

We represent the wireless sensor network based on a undirected graph $G = (V, E)$ where V represents the set of sensor nodes and E represents the set of edges. In the case of $(u, v) \in E$, node u and v are said to be *one hop apart*. Further, if u and v are not one hop apart but there has an intermediate node k such that $(u, k) \in E$ and $(v, k) \in E$, node u and v are said to be *two hops apart*. [2]

There are two types of collisions: direct collision and hidden collision. Let $u, v \in V$ that are one hop apart. If us transmission time overlaps with vs transmission time, a direct collision will occur. On the other hand, a hidden collision will

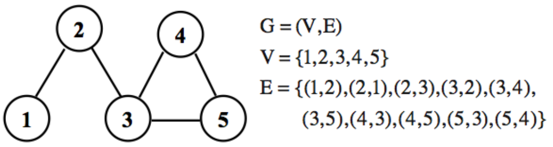
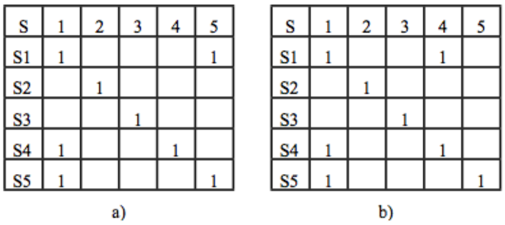


Fig. 1. Network represented with graph theory

978-1-4244-3709-2/10/$25.00 ©2010 IEEE



S	1	2	3	4	5
S1	1				1
S2		1			
S3			1		
S4	1			1	
S5	1				1

a)

S	1	2	3	4	5
S1	1			1	
S2		1			
S3			1		
S4	1			1	
S5	1				1

b)

Fig. 2. initial scheduling matrix

occur when both u and v transmit data to k at overlapping times, and u and v are not one hop apart. For the example, node 4 and 5 are direct collision because they share channel on node 3 and be one hop apart. Node 2 and 4 are hidden collision because even they do not share communication channel, but their packet may collide if they send packet to same destination (node 3) at the same time.

Time Division Multiple Access (TDMA) is one of solutions to reduce packet collision problem. Total transmission time is divided into frame and each frame is divided into time slot. After that each time slot will be assigned to sensor node to guarantee that every nodes are authorized for sending packet in its time slot. A schedule matrix, $S = \{s_{ij}\}$, is proposed in order to schedule packet to frame. S refer to a matrix of size NxL, N denotes the number of nodes (columns) while L denotes time slots (row) of the schedule matrix [2].

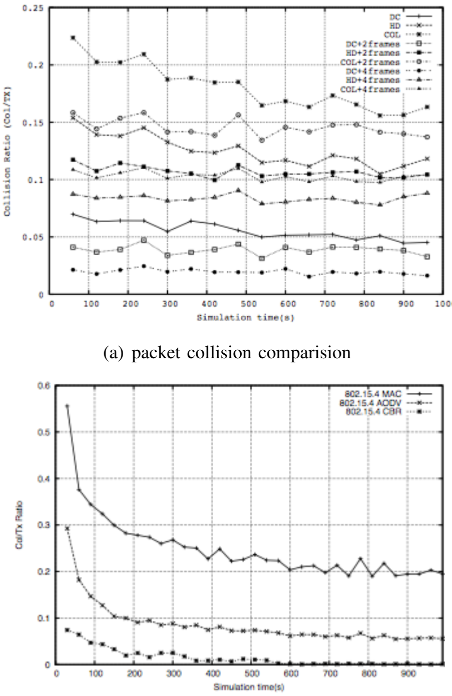
The wireless sensor network in figure 1 is represented in graph theory notation and transformed to the schedule matrix which is show in figure 2. Every nodes must be guaranteed to have at least one slot for example node i must be granted in slot i. In figure 2a time slot 1(S1) is reserved for node 1 thus node 1 can send message in this time slot. To increase throughput, node 5 is granted for sending message in S1 because both node 1 and 5 are not the collided node. Although, node 4 does not collide with node 1 but the S1 is not assigned to its in order to avoid conflict when node 1, 4, and 5 are granted in the same time slot.

The optimal schedule matrix will minimize frame length in order to decrease average delay time, and maximizes channel utilization in order to increase throughput. We will review the algorithm for the optimization of the schedule matrix and also explains and compare the performance with other approached in section III.

B. Network collision representation

To study the problems and find an optimal solution for packet collisions in wireless sensor networks, we simulate IEEE 802.15.4 standard using NS2. This experiment compares the collision ratio of network when the packet collisions occur.

We show the ratio between direct collision and hidden collision and classify all collided packets in each layer: application, routing, and MAC layer. The 100 sensor nodes are randomly placed in a 100m x 100m area. Each node has a 10m transmission range. Each source node sends 4 packets per second. Each data packet is 70 bytes constant bit rate (CBR) generated by NS2. All data packets are sent to a single sink (node id 0) over AODV routing protocol.



(a) packet collision comparision

(b) packet collision classified in layers

Fig. 3. packet collision problem study

We randomly divide sensor nodes into 1,2, and 4 frames(group). The sensor nodes are the same frame start to send packet in the same time.

According to the simulation, we can classify the collision problem. The hidden-node collision is clearly noticeable in figure 3(a). Whereas the direct collision is approximated 30% of the overall collision. This result conforms to the existing research [3], [4]. When we compare the results from 2 and 4 frame, we found that collision ratio in 2,4 frames less than one group because the number of nodes in the same time period decreases. This also causes collision in the MAC layer decreased. However, both direct collisions and hidden collisions still occur within frame.

We set up the experiment to study packet collision in different network layers (MAC, routing and application layers) using NS2 to find out the ability to dominate the collision. Our simulation results show the collision ratio in wireless sensor networks in figure 3(b). We found that the highest collision rate happen in the MAC layer while the application layer has a few collisions. As can be seen, the collision ratio is high at the beginning of the process because of the broadcast packets in routing phase are establishing network. When any node would like to send a CBR packet, the route request packet will be sent in order to route to destination node. After that routing packet is resolved Ethernet address to send message to neighbor, and will be sent in the MAC layer. From the result, it can be concluded that each outgoing application packet generates many control packets in low layer. Their packets are collided, dropped, and retransmitted until the communication stop. Because the importance characteristics of wireless sensor network topology are usually changed, node prone to fail, and sleep and wakeup itself that cause a lot of broadcasted packet and bring to packet flooding in network. However, the back-off technique in MAC layer is designed to reduce the 2^nd collision. After the collision occur, the collided nodes will random a positive integer and start to count down until it reaches zero. Then its packet is sent again. As the time pass by, the collision graph is in a downward trend.

We conclude from our simulation that the scheduling algorithm should be aimed to reduce the collision at the network establishment stage. We also consider designing our scheduling algorithm in the logical link control by using TDMA technique as the simulation, we can filter unnecessary packets that are passed to lower layer. Unfortunately, hidden-node collision still exists. The throughput, node delay, and fairness have to be considered carefully when we designed the scheduling algorithm. The first issue that we consider is throughput. The objectives of wireless sensor networks are monitoring the phenomenal and tracking objects. Eventhough, this network has long life, it will be useless if it can not collect any data.

Therefore, we proposed the scheduling algorithm for wireless sensor network aiming to reduce the collision at the logical link control that trade-off between throughput and average delay. Our algorithm emphasizes to improve the throughput and solve hidden-node collision. However, this paper scopes to explains the algorithm and analyzes throughput and average time delay only. The simulation and experiment in sensor node are set to be the future work.

III. Packet Collision Avoidance Algorithm

A. Literature Review

The packet scheduling algorithm minimizes frame length (refer to delay time) and maximizes channel utilization (refer to throughput). The several proposed algorithms used graph theory in order to represent network and optimize schedule matrix. The proposed algorithms are categorized into three approaches: the simple heuristic search, the heuristic search for NP-complete problem, and the finite state machine minimization.

The first approach, wireless networks are represented in the graph theory and schedule matrix. This idea consists of two steps: reducing the row of schedule matrix and after that increasing throughput with any algorithm. The method reported in [5] proposed the heuristic search to find the optimal schedule matrix that minimizes frame length and maximizes throughputs. This approach is the combination of the tabu search and the greedy algorithm. The tabu search is meta heuristic that applied to find the minimal frame length, while they maximize throughput with the greedy algorithm. The results were compared with HNN-GA [6]. The proposed algorithm did not introduce any shorter frame lengths, but only small improvements in average delay time were achieved. In addition, this algorithm produced the higher throughput when the number of nodes increases.

In [7], finding the optimal schedule matrix in broadcast scheduling problem could be prove to be NP-complete. In order to minimize the frame length and maximize throughput, the approximation methods are proposed to find a optimal schedule matrix. The new approaches based on heuristic search are proposed to find the optimal schedule matrix. The approximation methods were HNN-GA [6], MFA [7], SVC [8], and BSC-NCNN [9]. The minimal frame length from each method produced the near value. BSC-NCNN produced the most minimal average time delay, while the combination of neural network (NN) and genetic algorithm (GA) provided the higher throughput.

The new idea is proposed in order to finding packet scheduling algorithm. The schedule matrix is modeled with finite state machine (FSM) [10]. The main idea of other algorithms are generated all time slot first, after that they will reduce the unused time slot. In contrast, This method generate the time slot from scratch. This idea combines all nodes that do not cause any collision together under concept of maximal compatibles and incompatibles in FSM definition [11]. The result is the minimal time slots are generated. The new schedule matrix produce the minimal frame length and maximal throughput. The algorithm is efficient for all the benchmark cases and generates schedules within tight lower bound in negligible time.

B. Performance evaluation

There are two performance metrics: average delay and throughput. Average delay depends on the number of time slot. and indicates delay of each node for the next round of packet sending. When we minimize the frame length, it brings to the minimal average delay. On the other hand, throughput is number of reserved time slots that be assigned to sensor nodes and indicate the performance of network per frame in TDMA. [12]. Scheduling algorithm must trade-off between average delay time and throughput. Number of time slot increase throughput but also decrease average delay. The schedule matrix, S size NxL. When N is number of node, L is frame length, and s_{ij} is the status of node in each time slot.

Average delay (τ)

$$\tau = \frac{L}{N}\sum_{i=1}^{N}\left(\frac{1}{\sum_{j=1}^{L} s_{ij}}\right) \tag{1}$$

Throughput (σ)

$$\sigma = \sum_{i=1}^{N}\sum_{j=i}^{L} s_{ij} \tag{2}$$

C. *Our proposed algorithm*

We propose an algorithm for finding an optimal schedule matrix based on the greedy algorithm. This algorithm consists of three steps: initiating schedule matrix, finding the minimal frame length, and increasing throughput. Before explain this algorithm, we must define three importance terminology first.

Definition 1 node status: There are three statuses in our algorithm: black, gray, and white. Any node in black status can send message out under collision free. In contrast, if any node in white status send message out, packet may be collided. At the start time, all nodes is unknown status that labeled with gray and can be changed to other status depend on topology. The figure 4(a) show the example of time slot that consists of 15 nodes. Each node is set with any status. The black, gray, and white status are filled with black, gray with x, and white respectively.

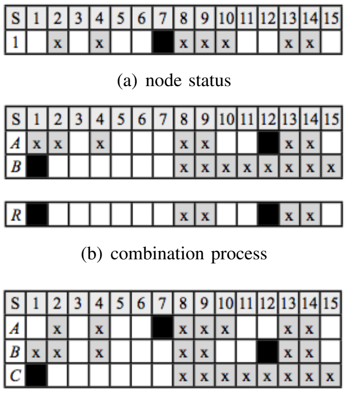
Definition 2 combine() : This algorithm find the minimal row of schedule matrix by combining time slot. Let A and B are arbitrary time slots. Let T_i denotes arbitrary node status in time slot T while T_1, T_2, and T_3 denote node status of node 1, 2, and 3 in time slot T respectively. The notation $combine(A, B)$ mean that time slot A is combined with time slot B under the conditions:

2.1) All new node status are replaced with status of A_i if B_i is gray node.

2.2) Otherwise, they will be replaced with B_i.

2.3) $combine(A, B) = combine(B, A)$.

In figure 4(b), $combine(A, B)$ can be explained that the definition 2.1 bring to $R_8 - R_{15}$, while $R_1 - R_7$ come from definition 2.2. From definition 2.1, the gray nodes can be changed to black or white nodes because gray nodes are unknown status. Whether, the collision will occur or not depend on other nodes status in the same time slot. For example, R_2 is set to any status that depend on B_1. If R_2 is set to black, it will conflict with R_1. In contrast, R_i depend on B_i if Ai is not gray node.



S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		x		x			■	x	x	x			x	x	

(a) node status

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	x	x		x				x	x			■	x	x	
B	■							x	x	x	x	x	x	x	x

R	■							x	x			■	x	x	

(b) combination process

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A		x		x			■	x	x	x			x	x	
B	x	x		x				x	x			■	x	x	
C	■							x	x	x	x	x	x	x	x

(c) matching process

Fig. 4. definition in our algorithm

Definition 3 match(): Only two matched time slot can be combined together. The notation $match(A, B)$ test that the time slot A and B are matched before combination process in definition 2. Time slot A and B are matched if only if all nodes in each time slot follow at least one of three conditions below:

3.1) A_i is gray node while B_i is any status. Because A_i is unknown status and can be replaced with any status of B_i.

3.2) A_i is black node and B_i is gray node mean that although A_i is reserved for node i, they can be combined because B_i can be changed to any status.

3.3) $A_i$is white node while B_i is not black node. If any node is blocked in time slot A, the same node in time slot B must be blocked or will be set as unknown status.

From figure 4(c), time slot B match with time slot C while time slot A does not match with time slot B. When we determine node B_1, B_2, B_4, and $C_8 - C_{15}$, we found that they fail in definition 3.1 because there were gray node. While node B_3 and $B_5 - B_7$ fail in definition 3.3 because they were all white nodes in B and C. In the same way, node C_1 and B_{12} also fail in definition 3.2 because there were black node while other time slots were gray node. When we determined $match(A, B)$, we concluded that they did not mach because A_7 and B_{12} conflicted with definition 3.2. One of them was black while the other was white that brought us to conclude they could not be combined.

Our proposed algorithm consists of three steps that explain below:

Step 1) scheduling matrix initiation: The network is transformed to a schedule matrix. Let S denotes schedule matrix and S_i denotes the set of status nodes in time slot T_i. While s_{ij} denotes the node status of node j in T_i that represent with black, gray, or white status.

At the beginning stage, T_i is assigned for node i. All s_{ij} are set to gray except node i is set to black. All adjacency nodes i are set to white in order to prevent directed collision, and all adjacency of adjacency nodes are set to white in order to prevent hidden collision. Finally, T_i is added into initial schedule matrix. After repeat all steps above apply to all node in network. The frame length of initial schedule is equal to number of node.

Step 2) finding the minimal frame length: The frame length indicates the average delay. To minimize the frame length of schedule matrix, we try to groups all time slot follow definition 2 and provide the maximum gray nodes in one time slot under greedy concept. First, we find two time slots from schedule matrix under three conditions:

2.1) They are the first and second time slots that provide the maximum gray status.

2.2) Two time slot must follow definition 3.

2.3) After they are combined according definition 2, the new time slot must provide the maximum gray status.

After that, we remove two time slots that accept three conditions above from schedule matrix, combine them together and add new time slot into schedule matrix. Then we find two time slots and combine them together again until we can not

TABLE I
PERFORMANCE COMPARISION

	15 nodes		30 nodes		45 nodes	
algorithm	τ	σ	τ	σ	τ	σ
TABU-GR [5]	-	20	-	37	-	68
HNN-GA [6]	7.00	20	9.30	35	6.30	77
MFA [7]	7.20	18	10.67	38	6.99	71
SVC [8]	7.20	18	9.99	37	6.76	60
BSC-NCNN [9]	6.80	-	9.20	-	5.80	-
FSM [10]	6.84	20	9.20	35	6.00	64
our algorithm	8.40	17	13.71	47	9.62	83

find any time slots that match together. Finally, we get the new schedule matrix that provides a minimal frame length.

Step 3) increasing throughput: The last step is the status replacement propose to increase throughput, thus gray status must be eliminated and replaced with black status.

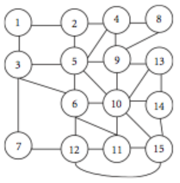
The schedule matrix is traversed in column order to find the node i that is the minimal gray status. After that let T denotes the set of time slot and T_i represent gray status of node i in T. Let O_i denote the time slot of node i in initial schedule matrix. After T_i that match with O_i is selected, they are combined to be new T_i that replace old T_i in schedule matrix. After that repeat all steps until the minimal gray node in T that match the conditions is not found.

D. Experimenting result

We execute our algorithm with benchmark that proposed by [9] that is represented with undirected graph shown in figure 5 and is presented the schedule matrix in figure 5, after that we evaluate the performance matrix in the term of average delay and throughput and compare our algorithm performance with exist works in Table I.

Most proposed methods as show in Table I have the advantages in the minimal delay. However, our algorithm gives the highest throughput compared the others. As be seen, when the number of nodes increases, our algorithm has better throughput. Our algorithm produces the higher frame length that increase average delay in contrast free slot is increasing as well. When we trade-off between the average time delay and throughout, we found that average time delay approximately increase 50% but throughput approximately increase 35%. However, our algorithm still allocate the highest number of frame length. This disadvantage can be improved in the future.

Moreover, this algorithm results in unfairness and consuming memory for numerous scheduling matrix when number of node is increased. According to figure 5, node 1 broadcasts two times per frame while node 2 broadcasts only once. In the addition, the priorities of packet are required. If any node requests to send control messages, they can occupy the time slot before other nodes request to send the data packets. As soon as the control messages are dropped, the retransmit mechanism will start and inject new control message into network. Thus, it causes a jam traffic and packet collisions once again.



TS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1				■			■							■	
2	■											■	■		
3	■								■			■			
4		■													■
5								■			■				
6					■										
7										■					
8						■		x							
9			■										x	x	x

Fig. 5. Network and its schedule matrix

IV. CONCLUSION

According to our experimenting result, the the highest collision packet ratio in the wireless sensor networks is in the MAC layer. Our algorithm is designed in order to avoid collision by delay some packet in corresponding time slot. This idea can also improve the performance of back-off technique in MAC layer in order to increase throughput. Our algorithm produces a high average delay, however, it gives the better throughput for number of nodes increases. We compare our algorithm with FSM technique. We found that our algorithm approximately increased throughput by 35% and 34%. In order to achieve the better performance, we will simulate our algorithm with network simulation and observe the relation of energy, packet collisions, throughputs in application layer, and average delay time.

REFERENCES

[1] Yick, J., B. Mukherjee, and D. Ghosal, *Wireless sensor network survey*. Computer Networks, 2008. 52(12): p. 2292-2330.

[2] Salcedo-Sanz, S., C. Bousono-Calzon, and A.R. Figueiras-Vidal, *A mixed neural-genetic algorithm for the broadcast scheduling problem*. Wireless Communications, IEEE Transactions on, 2003. 2(2): p. 277-283.

[3] Sun, M., K. Sun, and Y. Zou, *Analysis and Improvement for 802.15.4 Multi-hop Network*, in Proceedings of the 2009 WRI International Conference on Communications and Mobile Computing - Volume 02. 2009, IEEE Computer Society.

[4] Zheng, J. and M.J. Lee, *A Comprehensive Performance Study of IEEE 802.15.4*. 2004.

[5] Peng, Y., B.H. Soong, and L. Wang, *Broadcast scheduling in packet radio networks using mixed tabu-greedy algorithm*. Electronics Letters, 2004. 40(6): p. 375-376.

[6] Funabiki, N.T., Y.;, *A parallel algorithm for broadcast scheduling problems in packet radio networks*. Communications, IEEE Transactions on, 1993. 41(6): p. 828 - 831

[7] Wang, G. and N. Ansari, *Optimal broadcast scheduling in packet radio networks using mean field annealing*. Selected Areas in Communications, IEEE Journal on, 1997. 15(2): p. 250-260.

[8] Yeo, J., H. Lee, and S. Kim, *An efficient broadcast scheduling algorithm for TDMA ad-hoc networks*. Computers & Operations Research, 2002. 29(13): p. 1793-1806.

[9] Haixiang, S. and W. Lipo. *A hybrid neural network for optimal TDMA transmission scheduling in packet radio networks*. in Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on. 2005.

[10] Ahmad, I., B. Al-Kazemi, and A.S. Das, *An efficient algorithm to find broadcast schedule in ad hoc TDMA networks*. J. Comp. Sys., Netw., and Comm., 2008. 2008: p. 1-10.

[11] Imtiaz Ahmad, A.S.D., *A heuristic algorithm for the minimization of incompletely specified finite state machines*. Computers and Electrical Engineering, 2001: p. 159-172.

VITAE

Name Chatanan Jandaeng

Student ID 4910130006

Educational Attainment

Degree	Name of Institution	Year of Graduation
B.Sc. (Computer Science)	Prince of Songkla University	2003
M.Sc. (Computer Science)	Prince of Songkla University	2006

Scholarship Awards during Enrollment

- In 2007-2009, Lecturer Development Scholarship for the PhD program, Walailak University, Thailand

List of Publication and Proceeding

- C. Jandaeng, Wannarat Suntiamorntut, and Nittida Elz, PSA : Packet Scheduling Algorithm for Wireless Sensor Networks, International Journal on applications of graph theory in wireless ad hoc networks and sensor networks, vol. 3, issue 3, pp. 1-12, Sep, 2011.
- C. Jandaeng, Wannarat Suntiamorntut, and Nittida Elz, Performance Prediction Model of Packet Scheduling Algorithm in Wireless Sensor Networks, International Journal of Wireless & Mobile Networks, vol. 3, issue 4, pp. 113-126, Aug, 2011.
- C. Jandaeng, W. Suntiamorntut, and N. Elz. Waste Energy Reduction in Wireless Sensor Networks with Packet Scheduling Algorithm, Embedded System and Intelligent Technology, 2011. ICESIT 10. International Conference on, Feb. 2011., Phuket, Thailand.
- C. Jandaeng, W. Sunthiamontut and N. Elz, Packet Scheduling to Minimize Packet Collision and Maximize Throughput for Unslotted IEEE 802.15.4 Networks, The 2010 International Conference on Intelligent Network and Computing, ICINC, pp. 243-247, Nov., 2010., Kuala Lamper, Malaysia.
- C. Jandaeng, W. Suntiamorntut,and N. Elz. Throughput Improvement of Collision Avoidance Architecture in Wireless Sensor Networks, Wireless Communications, Networking and Mobile Computing, 2010. WiCom 10. 6th International Conference on, pages 1-5, Sep. 2010., Chengdu, China.