TIME- OPTIMAL CONVERGECAST IN SENSOR NETWORKS WITH MULTIPLE CHANNELS

A Thesis by

Masaaki Takahashi

Bachelor of Science, Wichita State University, 2008

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

August 2010

TIME- OPTIMAL CONVERGECAST IN SENSOR NETWORKS WITH MULTIPLE CHANNELS

The following faculty members have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

_____

Bin Tang, Committee Chair

_____

Vinod Namboodiri, Committee Member

_____

 Mehmet Bayram Yildirim, Committee Member

DEDICATION

This thesis is dedicated to my parents, my wife and my son

ACKNOWLEDGEMENTS

ABSTRACT

Sensor networks consist of a set of wireless sensor nodes which sense the environment and route the sensed information to a sink node. Data gathering is one of the important applications in sensor networks. In this thesis, using multiple channels, we consider TDMA schedule algorithm focusing on many-to-one communication which is called convergecast.

In convergecast, a packet generated in each sensor node is delivered to a sink node without any data aggregation in the intermediate node. In large network environment with many hops, we found that usage of multiple channels significantly improves delay efficiency comparing to usage of a single channel. We formulate this problem and propose heuristic algorithms for proper channel assignment and convergecast in different topologies. In general topology, with 2 channels, we present the convergecast algorithm requiring timeslot at most $2N - 1 + d$ where N is the number of sensor nodes and d is the delay to avoid interference among sensor nodes. We also found that, for general topology, usage of 4 channels contributes to even more delay efficiency which can achieve $d = 0$. Furthermore, unlike using 3 or fewer channels, using 4 channels is also memory efficient without creating an extra internal memory table in a sensor node to avoid interference among nodes. Additionally, such delay efficiency can result in energy efficiency which is also another important issue in sensor networks.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

In this thesis, we design TDMA scheduling algorithms for many-to-one communication convergecast in sensor networks using multiple channels. In convergecast, a packet generated in each sensor node is delivered to a sink node without any data aggregation in intermediate node. We study TDMA scheduling algorithms for convergecast using multiple channels in various networks. We are concerned about the distributed algorithm which can be applied to a real convergecast application in wireless sensor network. We take a step by step process starting with the simplest network topology such as a linear topology and, in the later chapter, find a heuristic convergecast algorithm in general topology with minimum delay.

Unlike studying delay efficient convergecast algorithm with a fixed number of channels and fixed number of packets, we have generalized channel assignment and convergecast problem with multiple channels, multiple packets at each node, and with a single half duplex radio transceiver. Consequently, there are some questions that need to be answered.

1) At most how many channels are required for delay efficient convergecast in different network topologies?

2) How do we utilize such number of channels in question 1 to complete convergecast in more delay efficient way?

3) What are the other benefits and advantages of using multiple channels other than making convergecast delay efficient?

For the first two questions, we take a step by step process starting from a single channel to multiple channels and justifying delay efficiency as well as at most how many channels are necessary. For the last question, we justify how to utilize multiple channels affects other wireless

sensor network issues such as reducing memory consumption. For example, if a single channel is used for convergecast, each sensor node needs to store information for all the neighbor nodes in memory and use such information to avoid interference for later communication; therefore, in high density network, each node needs to consume more memory for storing many neighbor nodes. On the other hand, if multiple channels are used, each sensor node needs to remember only the neighbor nodes which use the same channels.

1.1 Contributions of Thesis

The main contributions are as follows:

1) In large network environment with many hops, we show that usage of multiple channels significantly improves delay efficiency of convergecast as compared to use of a single channel. We formulate this problem and propose channel assignment and convergecast algorithms in different topologies. In general topology, with 2 channels, we present the convergecast algorithm requiring timeslot at most $2N - 1 + d$ where N is the number of sensor nodes and d is the delay to avoid interference among nodes.

2) We show that, for general topology, usage of 4 channels improves delay efficiency of convergecast; especially it can achieve $d = 0$ where d is the delay to avoid interference among nodes. Consequently, at most 4 channels are required for delay efficient convergecast. Furthermore, it can contribute to reducing node-memory consumption without creating an extra internal memory table to avoid interference among nodes; therefore, a convergecast algorithm with 4 channels can be even more delay efficient than using 3 or a fewer channels.

3) We implemented novel channel assignment and convergecast algorithms using multiple channels and conducted an extensive simulation to show that the convergecast with multiple

channels (2 channels) outperforms the one with a single channel in terms of delay efficiency and throughput.

Chapter 2

LITERATURE REVIEW

2.1 Background and Motivation

2.1.1. Data Gathering in Sensor Networks

In sensor networks, a set of wireless sensor nodes are spread over the fields, called sensor fields. Each sensor node senses the physical phenomenon and routes the sensed data to a sink node in real time. In sensor networks, data gathering is one of the major applications. Data gathering is many-to-one communication between sensor nodes in the sensor field and the sink node. As condition of physical environment and phenomenon changes, sensor node samples such changes and reports sensed data back to the sink node. When all sensor nodes report sensory data to the sink node simultaneously, packet collision occurs. Without any technique for avoiding collision among sensor nodes, sensory data will not reach the sink node successfully.

In such scenarios, a proper data gathering technique is necessary so that each sensor node can efficiently send data to the sink node without collisions among sensor nodes. Typically, data gathering problems can mainly falls into some categories. One is data aggregation where sensory data is averaged in the intermediate nodes. In data aggregation, when data reaches the sink node, they are already averaged in the process of being sent to the sink node. All sensory data from each sensor are not kept and only averaged data in the sink node are the main concerned. Data aggregation is useful when finding tendency of physical phenomenon [1] without requiring individual sensory data. Another technique is convergecast when individual sensory data is demanded such as patient monitoring in the hospital [2] [3], disaster monitoring [4] and vehicle tracking. Since we are concerned about convergecast, we discuss its challenges in the next sub section.

2.1.2. Challenges of Convergecast in Sensor Networks

In this section, we address convergecast problems and challenges. In convergecast, a packet generated in each sensor node is delivered to a sink node without any data aggregation in the intermediate node. The convergecast is even more important when individual unique data is needed such as patient monitoring [2] [3], and other tracking applications. As we realize from these convergecast applications, convergecast has to be sufficiently delay efficient as well as data accurate, since those convergecast applications are used as one type of emergency application, it also needs real time data gathering from individual nodes unlike data aggregation scenarios. Take patient monitoring for example, data gathered from life-critical patients needs to be delivered to a sink node immediately with high successful rate. Furthermore, in convergecast, since each packet generated in node has to be sent to a sink node without any data aggregation in the intermediate node, how to achieve energy-efficient data gathering is more challenging. Data aggregation typically requires less energy than convergecast because multiple data is usually merged and averaged in the intermediate node.

There are several challenges of convergecast such as delay, memory, energy efficiency, throughput, and data accuracy. Delay efficiency contributes to how fast a node can report its sensed data back to the sink node. Memory, energy and throughput constrains are originally the issues of typical sensor networks. Data accuracy is particularly the problem of convergecast since convergecast application is concerned about the sensed data from individual nodes. On the other hand, in data aggregation, a few errors could be covered by just averaging multiple sensory data from multiple nodes.

2.2. Related Work

As one of the closest and influenced work, a convergecast heuristic with a single channel is discussed in [7]. In [7], BFS tree is constructed for general topology and the minimum timeslot to complete convergecast is at most $3N - 2$. Since [7] uses a single channel, in order to avoid collision between nodes in different branches, possible pair of interfering nodes are detected in initialization phase of the convergecast and stored in memory table, called conflicting map. Each node stores the conflicting map. In the convergecast algorithm proposed in [7], a branch with the maximum number of packets has a higher priority to send packets to the sink node. Since each node maintains its radio state (transmit, receive, or idle), there are at most 3 branches that can forward packets to the sink node at a timeslot t. If an eligible branch for forwarding packets has any node whose paired-node in conflicting map is scheduled in timeslot $t - 1$ and $t - 2$, then that branch is suspended and another branch becomes eligible. This technique is one of the ways to avoid collision between nodes in different branches; however, it causes extra delay if there is no alternate branch eligible. Furthermore, if the density of sensor networks is higher, there might be a situation where eligible nodes are suspended many times. That may lead to serious delay and such conflicting map is not memory friendly since memory constraint is one of the major issues in sensor networks.

Another related work proposed a convergecast scheduling algorithm using 2 channels in [8]. Since 2 channels are utilized, a proper channel assignment is proposed in each topology. In star topology, all of branches are divided into two groups and each group has the same number of nodes; however, division problems are stated as NP-completeness. In our work, we design heuristic convergecast algorithm using 2 channels as well as k ($k > 2$) channels in star

topology. Furthermore, we improve channel assignment algorithm in general topology and propose 4-channel utilization for convergecast algorithm.

Usage of multiple packets and multiple channels are proposed with WirelessHART in [9]. WirelessHART uses a channel hopping technology that typical multi-channel TDMA protocols do not support. In WirelessHART, parallel transmissions occurred in a timeslot have to select different channels from each other in order to avoid collision; therefore, the number of channels used for parallel transmissions requires at most $\lceil N/2 \rceil$ with a single-packet buffering capability, considering paired-nodes: one transmits and another receives. For unlimited-packet buffering capability, such parallel transmissions require fewer channels comparing to a single-packet one, but still needs at most $\left\lceil N - \sqrt{N(N-1)/2} \right\rceil$ channels. According to [9], a maximum network size is as large as 53 with unlimited-packet buffering capability since parallel transmissions occurred in the same timeslot have to select different channels from each other and WirelessHART supports 16 channels. In this thesis, we still use the same channels at the same timeslot if each pair of nodes is not in an interfering range with each other. In this way, we can keep the number of channels for transmission as minimum as possible. Furthermore, our network size is not limited to a particular size, simulating the performance with 100 nodes to make sure that our algorithm works with a large network.

In [26], receiver-based frequency scheduling algorithm is proposed with multi-channel scheduling for convergecast. All the interferences between parent nodes are detected by using SINR and non-conflicting channels are assigned to the parents. Then, child nodes use the same channel as their parent nodes. According to [26], different frequencies are selected for every pair of parents if there are interferences between the parent themselves or the children of other parents; however, there are some cases that are not clearly discussed when parents do not have

an interference link, but each child node still has an interference link. For channel overlapping, SINR value is used; however if such value is not distinguishable from multiple child nodes, then transmission is suspended and schedule for next slots. Similar to a conflicting map of [7], there are cases that no alternative branch can be found due to such suspended transmission and this causes delay of convergecast.

Multiple channels are discussed in [11], applying a tree based sensor network fields. In [11], for example, sensor networks are divided into two trees in parallel; therefore two different convergecast are processed at the same time using different channels with each other. In this way, throughput and latency are improved with lower interference among the nodes in different trees; however in their assumption, a single channel is used in each tree using CSMA-based MAC protocol and multiple radio transceivers are used for the base station. In our work, we propose TDMA scheduling algorithms using multiple channels in one tree assuming that each node (including the sink node) has single radio transceiver.

2.3. Organization of Thesis

The rest of the thesis is organized as follows. In Chapter 3, we state the network model and assumption, and we also discuss the constraints and conditions for minimizing the number of timeslots. In Chapter 4, we propose convergecast and channel assignment algorithms in various networks. In Chapter 5, we show the performance evaluation of convergecast algorithms by comparing single and multiple channels. Finally we arrive at the conclusion and discuss our future work in Chapter 6.

Chapter 3

MODELS AND PROBLEMS

3.1.Network Model and Assumptions

We assume that the sensor network is modeled as a disk graph $G = (V, E)$, where $V = \{1, 2, \dots, N\}$ is the set of sensor nodes, and E is the set of edges. Two sensor nodes can communicate with each other directly if there is an edge connecting them. All the sensor nodes and sink node are static. The network connectivity is fixed once the network is established.

For nodes in the network, we assume that each node including the sink node has a half duplex interface and is equipped with multiple channels. Therefore, a node cannot transmit and receive at the same time. Let $C = \{c_1, c_2, \dots, c_k\}$ be available radio channels. The current devices of wireless sensor network usually support 16 ($k = 16$) orthogonal channels. If a node u and v are neighboring nodes and use channels c_l and c_m respectively with $l \neq m$, then the node u and v can transmit simultaneously to their different adjacent node x and w respectively with $x \neq w$, even though the node u and v are neighboring nodes with each other. Interference is also an essential problem in wireless sensor networks. Using multiple channels significantly increases the number of transmission in networks contributing on delay efficiency in convergecast.
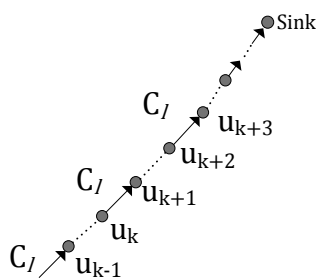

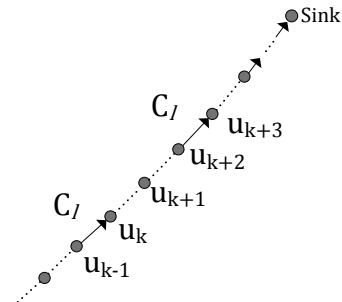Fig 1 Collision case with a single channel c_l Fig 2 No collision case with a single channel c_l

For example, as shown in Fig 1, suppose there is a simple wireless sensor network with a single channel c_l, arranging all the nodes in one line (we call this linear topology in later chapter). When node u_k and u_{k+2} send each packet to u_{k+1} and u_{k+3} respectively using channel c_l, packets are collided in node u_{k+1}; therefore, in order to avoid collision, as shown in Fig 2, every three-hop neighbor node ($u_{k-1+3n} \; \forall n \in \{..., -2, -1, 0, 1, 2, ...\}$) should send a packet to its immediate neighbor node; however, working this way, there are always idle nodes (e.g., node u_{k+1} in Fig 2) which can neither transmit nor receive a packet, increasing extra delay for convergecast.
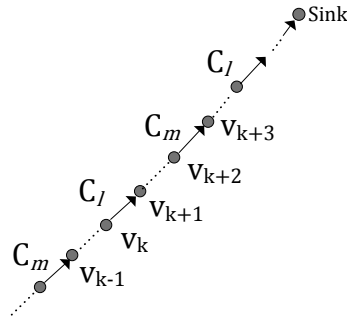
Fig 3 No collision case with multiple channels using c_l and c_m

On the other hand, as shown in Fig 3, suppose there is wireless sensor network with multiple channels using channel c_l and c_m, arranging all the nodes in one line. In here, even though node v_k and v_{k+2} send each packet to v_{k+1} and v_{k+3} respectively at the same time, packets are not collided in node v_{k+1} if the pair of v_k and v_{k+1} uses the channel c_l, and the pair of v_{k+2} and v_{k+3} uses channel c_m; therefore, every two-hop neighbor node ($v_{k+2n} \; \forall n \in \{..., -2, -1, 0, 1, 2, ...\}$) is able to send a packet to its immediate neighbor node using different

channels from each other such as channel c_l and c_m. Thus, utilization of multiple channels always enables a node to either transmit or receive without the node's being idle.

We also assume that every wireless link has the same bandwidth and each node has the same transmission range. Furthermore, as performing TDMA, we assume that time between nodes is always synchronized. Time synchronization is also a major topic in MAC protocols of wireless sensor networks, but we are concerned about delay-efficient using multiple channels. Therefore, here we ignore such time synchronization problem. In order to demonstrate convergecast characteristics, we assume that there is only one sink node. All nodes sense, send and relay sensory data back to the sink node without any data aggregation in the intermediate node.

For packet information, each packet length is fixed. Each node can store multiple packets and only one packet on a node is either received or transmitted in one timeslot.

In our convergecast scenario, a collision still happens when a node hears a message from more than two transmitters at the same time if those transmitters use the same channels.

3.2.Problem Statement

The main problem in this thesis is to minimize the number of timeslots using multiple channels in convergecast with a proper channel assignment and minimum number of channels. Before proposing the channel assignment and convergecast algorithms, we discuss the constraints and conditions for minimizing the number of timeslots.

Let $G = (V, E)$ be the set of sensor nodes where $V = \{1, 2, \dots, N\}$ and E is the set of edges. Any two sensor nodes can communicate with each other directly if there is an edge connecting them. Let l be the number of timeslots in order to complete convergecast. Let j be a

timeslot where $j \in \{1, ..., l\}$. Let $C = \{c_1, c_2, ..., c_k\}$ be available radio channels. Let $p_0(u)$ be the initial number of packets at node u and $p_j(u)$ be the number of packets at node u at a timeslot j. Let $N(C)$ be the total number of available channels. Let $Z(u)$ be all the one hop neighboring nodes of node u. Let $\varphi_j(u)$ be an assigned channel in node u at a timeslot j. The number of timeslots l in order to complete convergecast is minimized under the following transmission constraints from (1) through (4), channel assignment constraints from (5) through (8), and packet constraints from (9) through (11).

(1) The number of transmission from node u to another node v at a timeslot j is at most one.

(2) The number of neighbor nodes that node u can transmit a packet to in a timeslot j is at most one.

(3) Since each node is equipped with a half duplex interface, node u cannot transmit and receive simultaneously at a timeslot j.

(4) When node u transmits a packet at a timeslot j, one-hop neighbor nodes of node u can also transmit packets to their one-hop neighbors simultaneously if the one-hop neighbor nodes of node u are not in the same branch of node u. For example, as collision is discussed in network model and assumption section, with multiple channels, every two-hop neighbor node is able to send a packet to its immediate neighbor node if all the nodes are arranged in one line; however, there are situations that nodes are arranged in the topology as shown in Fig 4. We call this general topology discussed in detail in later chapter. In general topology, a topology is divided into multiple branches, and each branch consists of multiple nodes arranged in one line. Furthermore, at most two branches are able to forward packets to the sink node or closest node from the sink node in one timeslot.
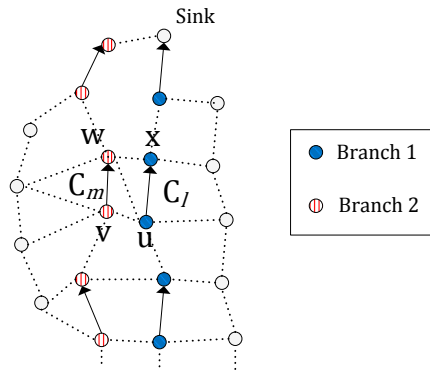
Fig 4 General topology using multiple channels using c_l and c_m

Thus, as Fig 4, there is a scenario that node u in branch 1 transmits a packet to one-hop neighbor node x in the same branch. At the same time, another one-hop neighbor node $v \in Z(u)$ in branch 2 is also able to transmit a packet to its one hop neighbor node $w \in Z(v)$ in its same branch. Therefore, we have to make sure such a case that one-hop neighbor nodes of node u can send their packets simultaneously. Both node v and w belong to the same branch with each other, but both are in different branch from node u. Since node v and w are a paired-node in the same branch where one node transmits a packet and another receives it, the number of transmission from node v and w is less than or equal to $Z(u)/2$.

(5) When node u transmits a packet to another node v, only one channel is selected from C at a timeslot j.

(6) When node u transmits a packet to another node v, its channel is always selected from C; therefore, the total number of selected channels for node u and v by a timeslot l does not exceed $N(C)$.

13

(7) When nodes u and v are in the different branches and the both nodes transmit packets to their receivers at timeslot j, then $\varphi_j(u) \neq \varphi_j(v)$ if interference between nodes among different branches are considered.

(8) Let μ_h be a node where h is a hop count from the sink node. When two-hop neighbor nodes μ_{h+1} and μ_{h+3} are in the same branch and the both nodes transmit packets to their receivers μ_h and μ_{h+2} respectively at timeslot j, then $\varphi_j(\mu_h) = \varphi_j(\mu_{h+1})$, $\varphi_j(\mu_{h+2}) = \varphi_j(\mu_{h+3})$ and $\varphi_j(\mu_{h+1}) \neq \varphi_j(\mu_{h+3})$. This ensures that, if two different paired-nodes are next to each other in the same branch, they need to use different channels.

(9) All the packets of each node at a timeslot $j = 1$ are collected to the sink node at a timeslot l.

(10) If node u is a transmitter, then $p_{j-1}(u) - p_j(u) \leq 1$ since node u can transmit a packet to at most one neighbor in a timeslot j. On the other hand, if node u is a receiver, then $p_j(u) - p_{j-1}(u) \leq 1$.

(11) There are multiple packets in a node u at a timeslot j; therefore, $p_j(u) \geq 0$.

Scheduling problem for convergecast is known as NP-Hard. In [27] and [28], solving an optimal convergecast problem in general graph is proved as NP-hard by using a partition problem which is also known as NP-hard problem.

Chapter 4

CHANNEL ASSIGNMENT AND CONVERGECAST IN NETWORKS

4.1. Linear Topology

In linear topology, all sensor nodes are arranged in a single line; therefore, one sensor can have at most two other adjacent neighbor nodes. The sink node of linear topology is allocated at the end of the line and only one closest sensor node to the sink node can send its packet to the sink node. In linear topology, node IDs $i \in \{1, 2 \dots N\}$ are assigned to each node based on the node hop count where the ID is equal to the node hop count from the sink node. We start with channel assignment algorithm which is a part of convergecast algorithm. For each algorithm, we begin with a single channel and then multiple channels. Furthermore, we analyze at most how many channels are required in channel assignment when there are multiple channels available.

4.1.1. Channel Assignment Algorithm in Linear Topology

It is obvious that without any proper channel assignment, sensor nodes cannot communicate properly. For example, in linear topology, if a node u with channel c_1 transmits a packet to a node v with channel c_2 where $u \neq v$ and $c_1 \neq c_2$, then there is no way that node u and v can communicate with each other. Therefore, when we use multiple channels, it is significant to have proper and efficient ways of assigning a channel for each node in each timeslot. From the following sub section, we start from using a single channel because, in order to show that usage of multiple channels improves delay efficiency for convergecast algorithm, it is necessary to justify characteristics of a single channel in convergecast and prove that utilizing multiple channels is more efficient than using a single channel in terms of performance of convergecast algorithm discussed in introduction.

With a single channel, it is no doubt that there exists no channel assignment. On the other hand, there need for a proper channel assignment with 2 channels. First, it is critical to realize that there are two radio operations (transmit and receive) in a sensor node based on our assumption. Therefore, we can accomplish the channel assignment as shown in Fig 5.
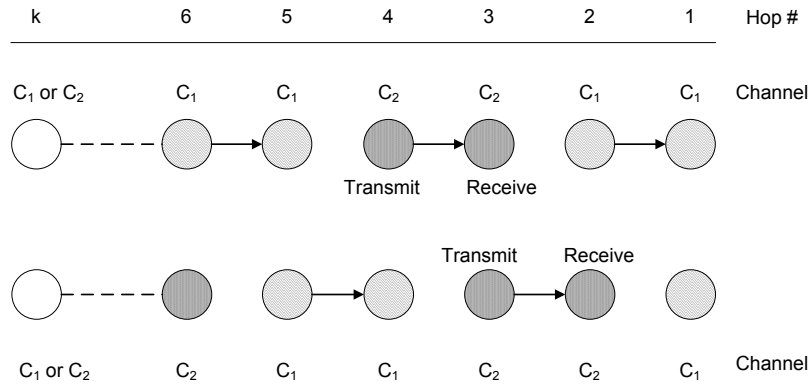

Fig 5: c_1 and c_2 represent channel 1 and 2 respectively

The algorithm for channel assignment is shown in table 1. Let $c_u(h, \varphi) \in \{c_1, c_2\}$ be the channel assigned for a node $u \in \{1, 2, .., N\}$ using its hop count h and the radio operation $\varphi \in \{\varphi | Transmit, Receive\}$. The channel assignment is shown as follows:

$$c_u(h, \varphi) \begin{cases} c_1 \; if \; h \; mod \; 4 = 1 \; \wedge \; \forall \varphi \in \{\varphi | Transmit, Receive\}, \\ \quad \vee \; if \; h \; mod \; 4 = 0 \; \wedge \varphi = Receive \\ \quad \vee \; if \; h \; mod \; 4 = 2 \; \wedge \varphi = Transmit \\ c_2 \; Otherwise \end{cases}$$

There are totally 4 cases to assign a channel based on node hop count and node's radio operation. If $h \bmod 4 = 1$, c_1 is always assigned to the node. If $h \bmod 4 = 2$, c_1 is initially assigned to the node when the radio state is in transmission. In next timeslot, c_2 is assigned when the radio state is in reception. In next timeslot, again, c_1 is assigned when the radio state is in transmission and so on. If $h \bmod 4 = 3$, c_2 is always assigned to the node. Finally, if $h \bmod 4 = 0$, c_2 is initially assigned to the node when the radio state is in transmission. In next timeslot, c_1 is assigned when the radio state is in reception. In next timeslot, again, c_2 is assigned when the radio state is in transmission and so on.

Table 1

Algorithm 1 (Channel assignment algorithm for 2 channels in linear topology)

Input hop count h, radio state of node u $\varphi(u)$
Output channel

begin
    if $h \bmod 4 = 0$ **then**
        if $\varphi(u) = transmit$ **then**
            return C_1
        else [receive]
            return C_1
        end
    else if $h \bmod 4 = 1$ **then**
        return C_1
    else if $h \bmod 4 = 2$ **then**
        if $\varphi(u) = transmit$ **then**
            return C_1
        else [receive]
            return C_2
        end
    else [$h \bmod 4 = 3$]
        return C_2
    end
end

When k ($k > 2$) channels are used in linear topology, (there is an assumption that there k ($k > 2$) channels exist), there are many ways to assign channels for each node. The example is shown in Fig 6. Interestingly, since there is the limited number of radio operations such as transmit and receive, more than 2 channels are not necessary in linear topology. Therefore, the channel assignment algorithm for 2 channels can be used for k ($k > 2$) channel case.
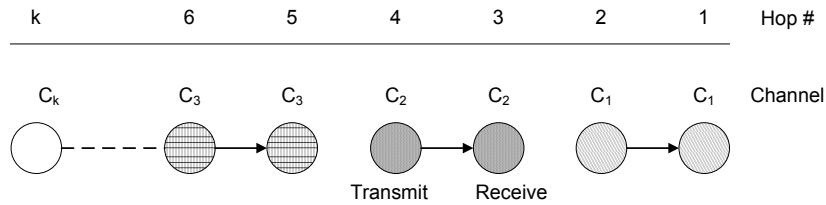

Fig 6: C_1, C_2, C_3, …, C_k represent channel 1, 2, 3 and k respectively

Theorem 1

Let k_u be the number of radio channels available in a sensor node u. In linear topology using $k_u > 1$ channels, the number of channels required in channel assignment algorithm is at most 2 for radio operations in convergecast.

Proof

Consider the formula discussed in using 2 channels case. Assuming there are k_u radio channels where $k_u > 1$ and radio transmission range covers only 1 hop adjacent nodes, since there are two radio operations (transmit and receive) in each sensor node, the number of channels required in convergecast in linear topology is at most 2.

4.1.2. Convergecast Algorithm in Linear Topology

Along with channel assignment algorithm, we also propose the convergecast algorithm with multiple channels. When using a single channel, Theorem 2 of [7] shows that "The convergecast scheduling algorithm requires $3N - 2$ timeslots to complete convergecast in linear networks, where N is the number of nodes in the network." On the other hand, when 2 channels are utilized, Lemma 1 and theorem 1 of [8] shows that "$2N - 1$ is the lower bound for delay of data gathering in linear topology, i.e., for any algorithm, the resulting delay is at least $2N - 1$, where N is the number of nodes in the network excluding the sink." As discussed in previous section, utilizing more than 2 channels are not necessary for linear topology; therefore, we can say the following:

Theorem 2

Let N be the number of sensor nodes and let k_u be the number of radio channels available in a sensor node u. In linear topology using $k_u \geq 1$, multiple channel convergecast algorithm requires $2N - 1$ timeslots to complete convergecast.

Proof

Let f_k be the number of the timeslots to complete convergecast where k is the number of the channels. For k (k≥ 1) channels, since Theorem 1 shows that radio operation requires at most 2 ($k = 2$) channels for transmitting or receiving in linear topology, and multiple channel convergecast algorithm also requires $f_k = f_2 = 2N - 1$ timeslots to complete convergecast as discussed in [8]. On the other hand, the number of timeslots required to complete convergecast using a single channel is $f_1 = 3N - 2$. Therefore, $f_1 - f_k = N - 1 > 0$.

Theorem 3

Let k_u be the number of radio channels available in a sensor node u. In linear topology with using $k_u \geq 1$, the number of channels required in channel assignment algorithm is at most 2 in order to complete the convergecast with 2N – 1 timeslots.

Proof

Consider the theorem 1 and 2 in channel assignment algorithm and convergecast algorithm. Assuming there exist k radio channels, in linear topology, since there are two radio operations which transmit and receive (In sleep mode, radio is turned off and it does nothing) in each sensor node with $k_u \geq 1$, the number of channels required in convergecast is at most 2.

4.2.Star Topology

Star topology consists of multiple linear topologies. The group of several sensor nodes linearly connected is called a branch of star topology. As shown in Fig 7, each sensor node belongs to at least one branch. The interference between nodes in different branches is ignored in this case.
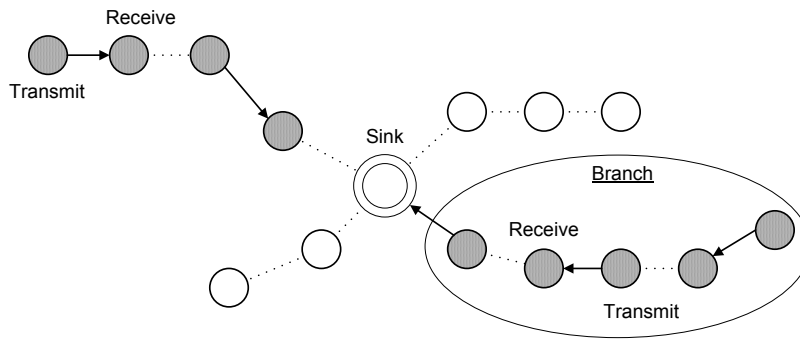

Fig 7 Star topology

4.2.1. Channel Assignment Algorithm in Star Topology

Similar to linear topology, there is no channel assignment with a single channel in star topology. For using 2 channels, initially we assume that unique IDs are assigned for each node and branch. Unlike linear topology, one sink node is shared by all branches. Therefore, only one sensor node with lowest node ID in a branch can send its packet to the sink node at one timeslot. During that time, other nodes in other branches are not able to forward packets to the sink and all those nodes needs to turn off the radio (with sleep mode) unless radio operation of the lowest sensor node ID in the branch is in receive mode and nodes in that branch can forward the packets to the sensor node with the lowest node ID. In order to reflect such scenarios in channel assignment algorithm, it is necessary to add the following conditional step (as shown in table 2) before any operation of the channel assignment algorithm 1 in linear topology:

21

Table 2

Algorithm 2 (modified part **algorithm 1** for 2 channels in star topology)

if $\varphi(u) = Sleep$ **then**

      return $\emptyset$ [no channel is assigned]

else

      [continue the channel assignment for 2 channels in linear topology]

end

When k $(k > 2)$ channels are concerned, there are many ways to assign a channel for each node as previously discussed in linear topology. Similar to linear topology, since there is the limited number of radio operations such as transmitting and receiving, having more than 2 channels is not necessary. Therefore, the channel assignment algorithm for 2 channels can be applied for k channels case. Thus, we can accomplish the channel assignment as shown in Fig 5. Let $c_u(h, \varphi) \in \{c_1, c_2\}$ be the channel assigned for a node $u \in \{1, 2, .., N\}$ using its hop count h and the radio operation $\varphi \in \{\varphi| \ Transmit. Receive, Sleep\}$, the channel assignment is shown as follows:

$$c_u(h, \varphi) \begin{cases} \emptyset \ if \ \varphi = Sleep \ (\emptyset \ is \ no \ channel) \\ c_1 \ if \ h \ mod \ 4 = 1 \ \wedge \ \forall s \in \{\varphi| \ Transmit. Receive\}, \\ \quad \vee \ if \ h \ mod \ 4 = 0 \ \wedge \varphi = Receive \\ \quad \vee \ if \ h \ mod \ 4 = 2 \ \wedge \varphi = Transmit \\ c_2 \ Otherwise \end{cases}$$

Theorem 4

Let k_u be the number of radio channels available in a sensor node u. In star topology with using $k_u > 1$, the number of channels required in channel assignment algorithm is at most 2 for radio operations in convergecast.

Proof

Consider the formula discussed in using 2 channels case in star topology. Assuming there are k_u radio channels where $k_u > 1$, since there are two radio operations (transmit and receive) in each sensor node, the number of channels required in convergecast in star topology is at most 2.

4.2.2. Convergecast Algorithm in Star Topology

For a single channel case, Theorem 5 of [7] shows that " If N represents the number of nodes in the network and n_k represents the maximum number of nodes in a branch, then the number of timeslots required by our convergecast scheduling algorithm for multi-line networks is given by $\max(3n_k - 1, N)$." Similarly, when using 2 or more channels, the branch which can send a packet to the sink node at one timeslot is the one which contains the maximum number of packets among all the branches. Furthermore, since we use 2 or more channels, nodes in at most two branches can forward their packets to either sink node or a closest node to the sink at one timeslot.

In order to achieve such an algorithm where at least one branch always transmits a packet to the sink without wasting any timeslot, each sensor node needs to have the following information in the initializing phase before starting convergecast: (1) the branch ID that a sensor

node belongs to and (2) initial number of the nodes in all the branches before starting convergecast. In this way, sensor nodes in the branch can determine when to transmit, receive or sleep. We extended an existing convergecast algorithm with a single channel to the one with multiple channels when $k = 2$ as shown in Table 3.

Table 3

Algorithm 3 (Convergecast algorithm using 2 channels for star topology)

Input timeslot t, node ID u, branch ID χ, hop count h
begin
    repeat
        $\theta = find\ (t)$ [find an set of eligible branches]
        $i = \max(p_j | j \in \theta)$ [find one branch which has maximum number of packets]
        $n_i = n_i - 1$ [subtract 1 packet from this node]
        $t_i = t + 1$
        if $t \le t_i$ **then**
            if $\varphi(u) = transmit$ **then**
                set channel $\rho(h, \varphi(u), t)$
                transmit a packet
                $\varphi(u) = receive$
            else if $\varphi(u) = receive$ **then**
                set channel $\rho(h, \varphi(u), t)$
                receive a packet
                $\varphi(u) = transmit$
            else
                [do nothing]
            end
        end
        $t = t + 1$
end

Theorem 5

Let N be the number of nodes in the sensor field, n_f be the number of sensor nodes in a branch where $f \in \{1, 2, ..., k\}$, n_k be the maximum number of the nodes in a branch and k_u be the number of radio channels available in a sensor node u. In star topology with using $k_u > 1$,

multiple channel convergecast algorithm requires $\max(2n_k - 1, N)$ timeslots to complete convergecast.
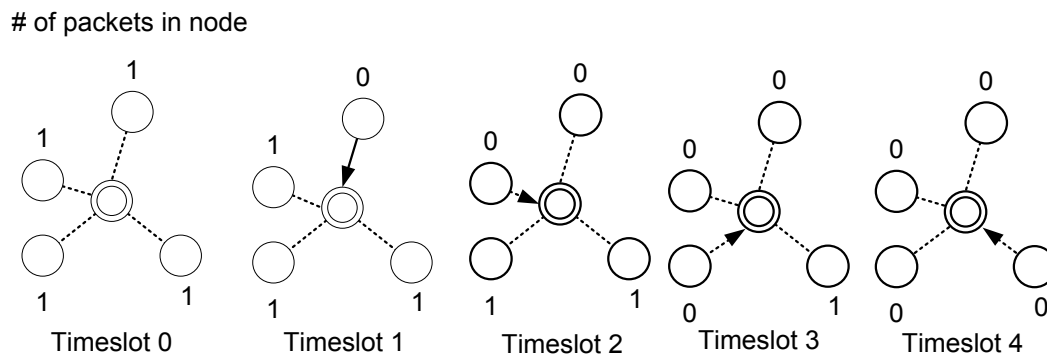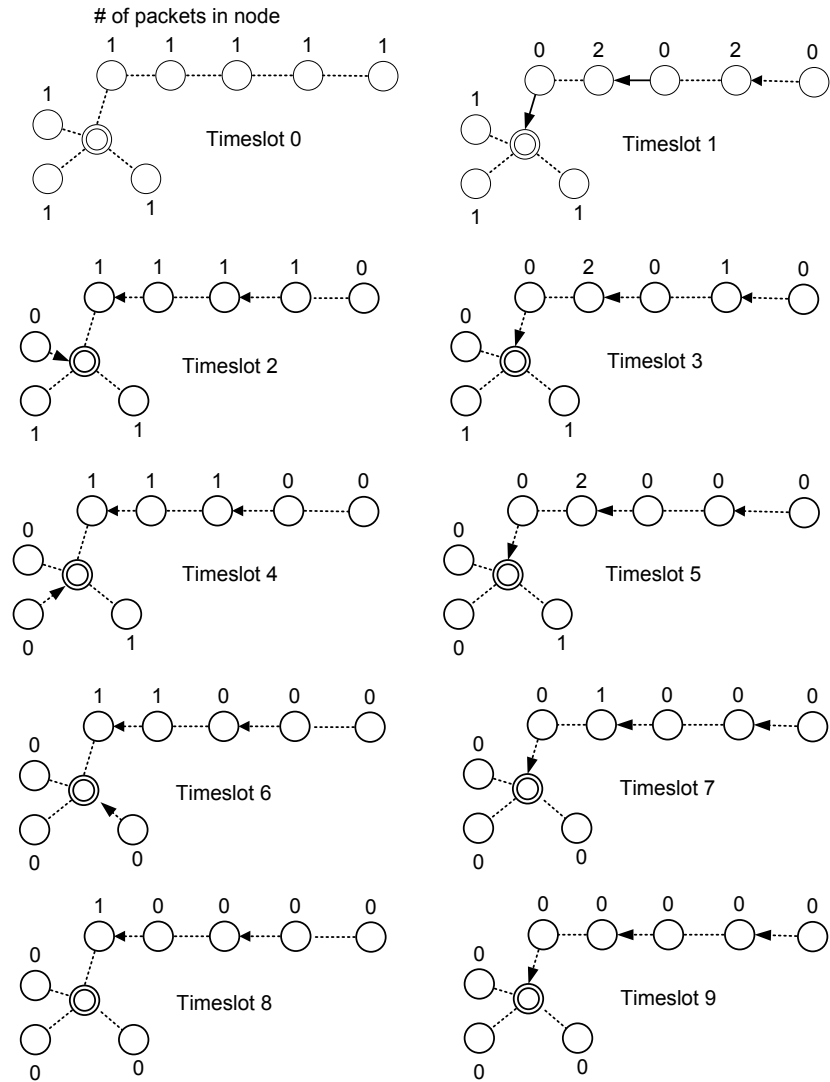

Fig 8. When timeslot of N is required

# of packets in node

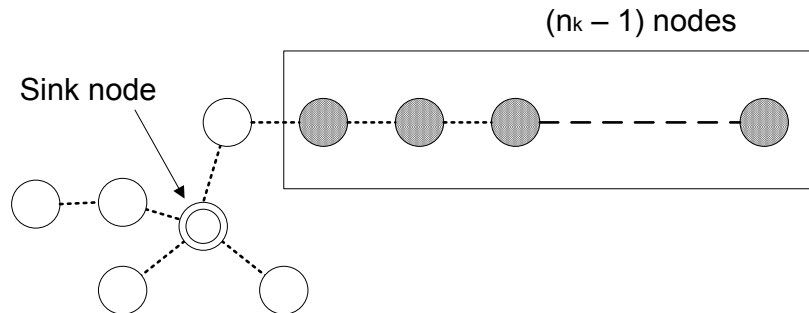Fig 9. When timeslot of $2n_k - 1$ is required

Fig 10. Branch which has the maximum
number of nodes (n_k) and other branches

26

Proof

If $2n_k - 1 \leq N$, required timeslot for convergecast is N.

Since the number of packets is initially equal to the total number of nodes N, the required timeslots has to be N when there is always a packet available sent to the sink node as shown in Fig 8. This is the case that, when the closest node (to the sink node) in a longest branch is in receiving mode, any one of nodes in other branches can send the packet to the sink node. Therefore, as shown in Fig 10, in order to always send a packet to the sink node, the following needs to be satisfied:

$n_k - 1 \leq n_1 + n_2 + \dots + n_{k-1}$ where $n_k \geq n_{k-1} \geq \dots \geq n_1$

Since we can say, $n_1 + n_2 + \dots + n_{k-1} = N - n_k$, then $n_k - 1 \leq N - n_k$.

Therefore, $2n_k - 1 \leq N$

Thus, required timeslot for convergecast when $2n_k - 1 \leq N$ is N.

If $2n_k - 1 > N$, required timeslot for convergecast is $2n_k - 1$.

Unlike the previous case, in this case, there are some timeslots that the sink node does not receive any packet from any node even though there are still packets remaining in a branch. Therefore, it has to satisfy as follows:

$n_k - 1 > n_1 + n_2 + \dots + n_{k-1}$ where $n_k \geq n_{k-1} \geq \dots \geq n_1$

Since we can say, $n_1 + n_2 + \dots + n_{k-1} = N - n_k$, then $n_k - 1 > N - n_k$.

Therefore, $2n_k - 1 > N$

When no branch can send a packet to the sink node, the branch which has remaining packets can send a packet every two timeslots. That means the required timeslots is $2n_k - 1$ which is for the

branch that initially contains the maximum number n_k of packets among all the branches. Therefore, the required timeslot for convergecast when $2n_k - 1 > N$ is $2n_k - 1$.

Theorem 6

Let N be the number of nodes in the sensor field, n_f be the number of sensor nodes in a branch where $f \in \{1, 2, ..., k\}$, n_k be the maximum number of the nodes in branch where k_u be the number of radio channels available in a sensor node u. In star topology using $k_u \geq 1$, multiple channel convergecast algorithm requires $\max(2n_k - 1, N)$ timeslots to complete convergecast if $3n_k - 1 > N$.

Proof

If $2n_k - 1 \leq N$ and $3n_k - 1 \leq N$ then, $t_k = t_2 = N$ timeslots are required to complete convergecast as theorem 5. On the other hand, the number of timeslots required to complete convergecast using a single channel is $t_1 = N$. Since $2n_k - 1 \leq 3n_k - 1 \leq N$,

$t_1 - t_k = N - N = 0$.

If $2n_k - 1 > N$ and $3n_k - 1 > N$, then $t_k = t_2 = 2n_k - 1$ timeslots is required to complete convergecast as theorem 5. On the other hand, the number of timeslots required to complete convergecast using a single channel is $t_1 = 3n_k - 1$. Since $3n_k - 1 > 2n_k - 1 > N$, $t_1 - t_k = n_k > 0$.

If $2n_k - 1 \leq N$ and $3n_k - 1 > N$, then $t_k = t_2 = N$ from theorem 5. On the other hand, $t_1 = 3n_k - 1$. Since $2n_k - 1 \leq N < 3n_k - 1$, $t_1 - t_k = 3n_k - 1 - N > 0$.

Theorem 7

Let k_u be the number of radio channels available in a sensor node u. In star topology using $k_u \geq 1$, the number of channels required in channel assignment algorithm is at most 2 in order to complete convergecast with $\max(2n_k - 1, N)$ timeslots if $3n_k - 1 > N$.

Proof

Consider the theorem 4 and 6 in channel assignment algorithm and convergecast algorithm. Assuming there exist k_u radio channels, in star topology, since there are two radio operations transmit and receive (When node is in sleep mode, the radio is turned off and it does nothing) in each sensor node with $k_u \geq 1$, the number of channels required in convergecast is at most 2.

4.3.Tree Topology

Tree topology is an extension of star topology using the idea of multiple linear topologies. In tree topology, there are common nodes among branches as shown in Fig 11. The interference between nodes in different branches is ignored in this case, too. However, in tree topology, because of some common nodes among several branches, an additional time scheduling is needed. For instance, as Fig 11, before nodes in branch 2 forward packets to the sink node, the nodes in the branch 2 need to wait until packets in a node b and c of branch 1 are forwarded to the common root a.
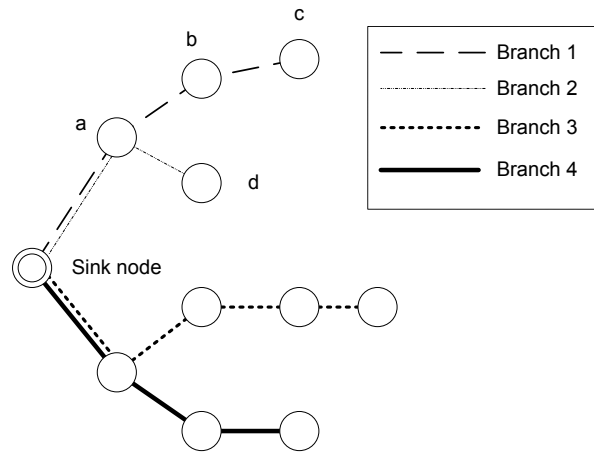

Fig 11 Branches in Tree Topology

4.3.1. Channel Assignment and Convergecast Algorithm

Using a single channel, as in linear and star topology, there is no channel assignment. When using 2 or more channels, since tree topology also consists of multiple linear topologies, channel assignment in tree topology with multiple channels is similar to the one in star topology; however the convergecast algorithm needs modification because some nodes in different branches are overlapping each other. Therefore, the required timeslot for the tree topology is

still max$(2n_k - 1, N)$, but n_k here represents a maximum number of packets that an immediate

sub-tree from the root has. In Fig 11, $n_k = 6$. Furthermore, we need to modify the proposed

algorithm in star topology as follows:

Table 4

Algorithm 4 (Convergecast algorithm using 2 channels for tree topology)

Input timeslot t, node ID u, branch ID χ, hop count h
begin
    repeat
        $\theta = find\ (t)$ [find an set of eligible branches using conflicting map]
        $i = \max(p_j | j \in \theta)$ [find one branch which has maximum number of packets]
        $n_i = n_i - 1$ [subtract 1 packet from this node]
        $t_i = t + 1$
        if $t \leq t_\chi \wedge p(n_\chi) \geq W_{v(\chi)}$ **then**
            $[W_{v(\chi)}$ is the number of packets that needs forwarding in χ
            before node v can transmit or receive]
            if $\varphi(u) = transmit$ **then**
                set channel $c(h, \varphi(u), t)$
                transmit a packet
                $\varphi(u) = receive$
            else if $\varphi(u) = receive$ **then**
                set channel $c(h, \varphi(u), t)$
                receive a packet
                $\varphi(u) = transmit$
            else
                [do nothing]
            end
        end
        $t = t + 1$
end

4.4.General Topology

In general topology, we construct a tree using BFS as in Fig 12 and apply tree topology idea for channel assignment and convergecast algorithms. Significant difference between tree and general topology is that, in general topology, we consider the interference between sensor nodes among different branches. Therefore, techniques to avoid node collision among nodes in the different branches must be considered. In this thesis, as we assume that there is no interference between a node and its two-hop neighbor nodes in the same branch. Thus, if a constructed graph forms the complete graph that every node has a link with each other, then a parent node of each node is the root node in a tree. In other word, each branch has only one node. In this case, the number of timeslots to complete convergecast is *N* discussed in theorem 5.
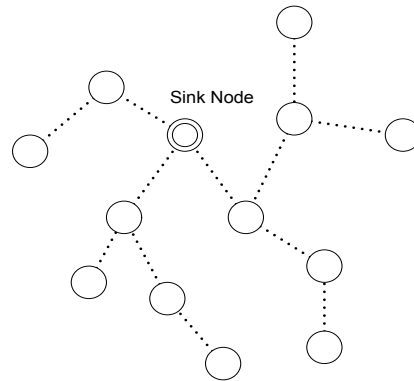

Fig 12 General topology constructed by BFS

4.4.1. Channel Assignment Algorithm in General Topology

Using a single channel, as in linear and star topology, there is no channel assignment. For using 2 channels, in order to avoid interference between nodes in the different branches, it is necessary for each node to know about neighbor's information so that multiple nodes do not transmit at the same time as shown in Fig 13. Use of a conflicting map proposed in [7] can solve

such problem in 2 channels. However, the conflicting map requires long initialization time. It also consumes an extra memory table for storing neighboring nodes information. In addition, in case of high density sensor fields, the conflicting map can be grown to a large memory table. For instance, If two one-hop-subtrees (one-hop-subtree is discussed in [7]) A and B are interfering because of existence of an interfering-edge (a, b), 2 timeslots are always wasted if number of nodes belonging to A and B are much larger than other branches. Also, in high density sensor fields, memory consumption issue becomes serious. If there are more than two interfering one-hop-subtrees such as A, B, C and D, etc, it causes plenty of delay due to wasting of 2 time slots. Such cases are not discussed in the paper [7].
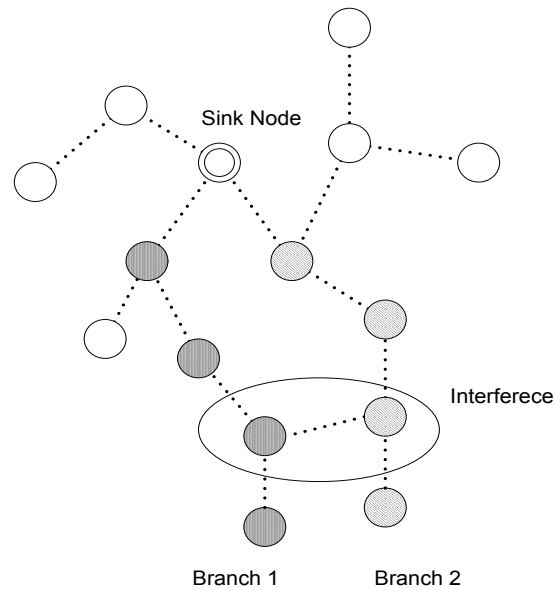

Fig 13: Interference in general topology

Therefore, it is important to have different technique without using any conflicting map. Since we assume that there are k channels available, we utilize more than 1 or 2 channels to deal with nodes interferences.

When we use 3 channels, it is still necessary to have techniques such as conflicting map. For example, as shown Fig 14, if node j and k use third channel c_3, there are still other collisions happening in node h. Node h receives a packet from both node b and i. Therefore, using 3 channels does not give us the solution to avoid conflicting edges between the nodes among different branches.
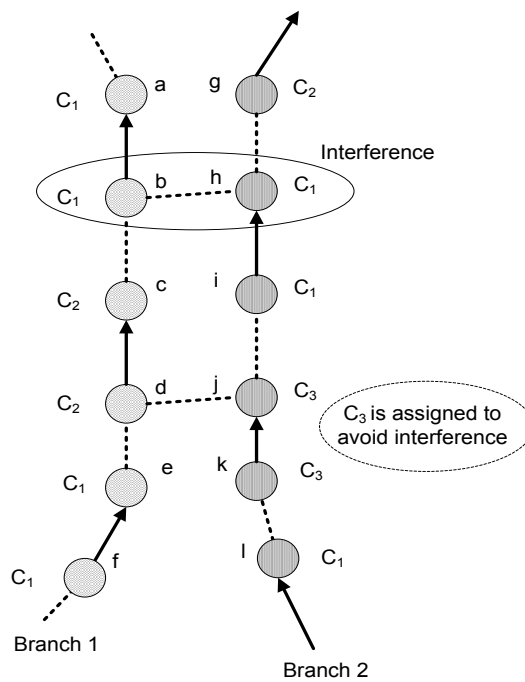

Fig 14. Interference when 3 channels are used

When we use 4 channels, as we know from star and tree topology, since a root node can receive a packet at one timeslot and there are only at most two branches forwarding packets to the sink node (root), we can divide timeslot into two categories.
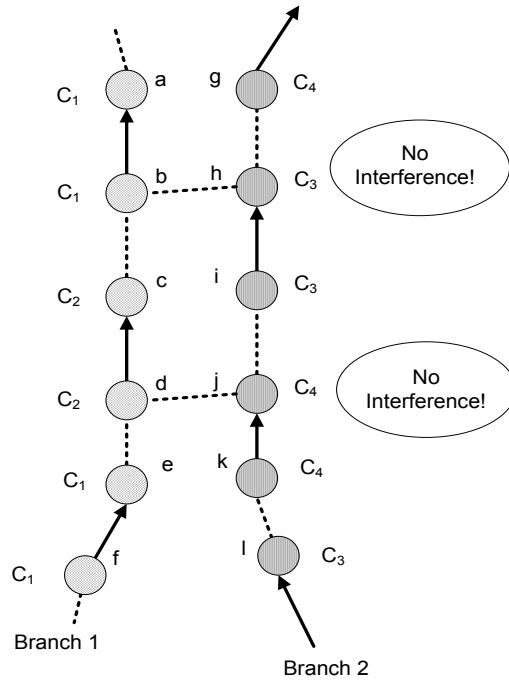

Fig 15. No interference when 4 channels are used

Let $t_j(i) \in \{0,1\}$ be the category in a timeslot j and branch i. $t_j(i) = 1$ if the timeslot j is an odd number timeslot. $t_j(i) = 0$ if the timeslot j is an even number timeslot.

For channel assignment using 4 channels, since there are two radio operations in sensor nodes (transmitting and receiving), there are at most two branches forwarding packets to the sink node or the closest node to the sink node. Therefore, we can accomplish the channel assignment as shown in Fig 15. Let $c_u(h, \varphi, j, i) = \{C_1, C_2, C_3, C_4\}$ be the channel assigned for a node $u \in \{1,2,...,N\}$ using its hop count h and the radio

operation $\varphi \in \{\varphi|\ Transmit.Receive, Sleep\}$, timeslot j and branch i, the channel assignment

is shown as follows:

$$
c_u(h, \varphi, j, i)
\begin{cases}
\emptyset\ if\ \varphi = Sleep\ (\emptyset\ is\ no\ channel) \\[1em]
c_1\ if\ t_j(i) = 0\ \wedge h\ mod\ 4 = 1\ \wedge\ \forall \varphi \in \{\varphi|\ Transmit.Receive\} \\[1em]
\qquad \vee\ if\ t_j(i) = 0\ \wedge h\ mod\ 4 = 0\ \wedge \varphi = Receive \\[1em]
\qquad \vee\ if\ t_j(i) = 0\ \wedge h\ mod\ 4 = 2\ \wedge \varphi = Transmit \\[1em]
c_2\ if\ t_j(i) = 0\ \wedge h\ mod\ 4 = 3 \\[1em]
\qquad \vee\ if\ t_j(i) = 0\ \wedge h\ mod\ 4 = 0\ \wedge \varphi = Transmit \\[1em]
\qquad \vee\ if\ t_j(i) = 0\ \wedge h\ mod\ 4 = 2\ \wedge \varphi = Receive \\[1em]
c_3\ if\ t_j(i) = 1\ \wedge h\ mod\ 4 = 1\ \wedge\ \forall \varphi \in \{\varphi|\ Transmit.Receive\} \\[1em]
\qquad \vee\ if\ t_j(i) = 1\ \wedge h\ mod\ 4 = 0\ \wedge \varphi = Receive \\[1em]
\qquad \vee\ if\ t_j(i) = 1\ \wedge h\ mod\ 4 = 2\ \wedge \varphi = Transmit \\[1em]
c_4\ Otherwise
\end{cases}
$$

Table 5

Algorithm 5 (Channel assignment algorithm for 4 channels in general topology)

Input hop count h, timeslot t, radio state of node u $\varphi(u)$, branch ID χ
Output channel

begin

    if $\varphi(u) = Sleep$ **then**
        return $\emptyset$ [no channel is assigned]
    else
        if $t_j(\chi) \bmod 2 = 0$ **then**
            if $h \bmod 4 = 0$ **then**
                if $\varphi(u) = transmit$ **then**
                    return c_2
                else [receive]
                    return c_1
            else if $h \bmod 4 = 1$**then**
                return c_1
            else if $h \bmod 4 = 2$**then**
                if $\varphi(u) = transmit$ **then**
                    return c_1
                else [receive]
                    return c_2
            else [$h \bmod 4 = 3$]
                return c_2
        else
            if $h \bmod 4 = 0$ **then**
                if $\varphi(u) = transmit$ **then**
                    return c_4
                else [receive]
                    return c_3
            else if $h \bmod 4 = 1$ **then**
                return c_3
            else if $h \bmod 4 = 2$ **then**
                if $\varphi(u) = transmit$ **then**
                    return c_3
                else [receive]
                    return c_4
            else [$h \bmod 4 = 3$]
                return c_4

            end
        end
end

When we use more than 4 channels ($k > 4$), there are many ways to assign channels; however since there only two radio operations (transmit and receive) and there are at most two branches that can forward the packets to the sink node or the closest node to the sink node at one timeslot, as shown in previous theorem 7, having more than 4 channels ($k > 4$) is not necessary.

Theorem 8

Let k_u be the number of radio channels available in a sensor node u. In general topology using $k_u > 1$, the number of channels required in channel assignment algorithm is at most 4 for radio operations in convergecast.

Proof

Consider the formula in channel assignment algorithm in general topology. Assuming there exist k_u radio channels, in general topology, since there are at most 2 branches forwarding packets to the sink node in one timeslot or the closest node to the sink node, and there are two radio operations (transmit and receive) in each sensor node with $k_u > 1$, the number of channels required in convergecast is at most 4. (Using 2 channels seems to be enough but there is still interference between nodes among different branches. Such interference cases should be considered when justifying at most how many channels are required; therefore, 2 is not an sufficient number of the channels for convergecast in general topology. Using 3 channels is also the same as 2 channels which still causes interference between nodes in different branches).

4.4.2. Convergecast Algorithm in General Topology

Using a single channel, Theorem 5 of [7] can be applied. On other hand, using 2 and 3 channels, Theorem 5 can be applied; however it is necessary to add the delay from conflicting map in general topology. Let d_k be the delay caused by conflicting map using k channels where $k \in \{2,3, \dots, K\}$ and n_m be the maximum number of packets in one hop subtree. Therefore, considering theorem 5, timeslot required for convergecast with 2 or 3 channels is at most $\max(2n_m - 1 + d_k, \; N + d_k)$. Using k ($k > 3$) channels, convergecast algorithm of 2 channels can be used, but use a new channel assignment algorithm which does not require conflicting map where $d_k = 0$. Therefore, considering theorem 5, timeslot required for convergecast with 4 or more channels is $\max(2n_m - 1, N)$

Theorem 9

Considering theorem 5 and 8, let k_u be the number of radio channels available in a sensor node u. In general topology with using $k_u \geq 1$, the number of channels required in channel assignment algorithm is at most 4 in order to complete convergecast with $\max(2n_m - 1, N)$.

.

Proof

Assuming there exist k_u radio channels, in general topology, the following facts are considered:

1) There are only two radio operations (transmit and receive) in each sensor node. In sleep mode, radio is turned off and it does nothing.

2) There are only two branches that can forward the packets to the sink node or the closest node to the sink node at one time slot.

Let d_k be the delay caused by conflicting map using k channels where $k \in \{1, 2, \ldots, n\}$.

Considering $d_j = 0 \ \forall j \in \{4, 5, \ldots, n\}, \ d_1 \geq d_2 \geq d_3 > d_j$.

Therefore, with $k_u \geq 1$, the number of channels required in delay optimal convergecast is at most 4.

Chapter 5

PERFORMANCE EVALUTION

5.1.Simulation Setup

In simulation, we use Rmase [12] simulator which is an extended application for prowler [13] simulator running on Matlab 6.5. Rmase is developed by Palo Alto Research Center. Rmase supports an event-driven structure and provides radio models for MacaZ and other Berkerly mote platform. Since TinyOS uses similar structure, it is useful when deploying to a real platform after the simulation. Rmase is particularly useful when researchers develop protocols such as routing, MAC, and application protocols etc. Once researchers create their own protocols, they can easily choose and combine various protocols in different protocol layers. Consequently, various kinds of protocol comparisons can be achieved.

In Rmase, it is possible to construct various topologies based on the researcher's needs. For example, as shown in Fig 16 and Fig 17, topology for 25 and 100 nodes respectively are constructed. Simulation can be animated during the simulation; therefore, we can also confirm the behavior of desired protocols by researcher's eyes.
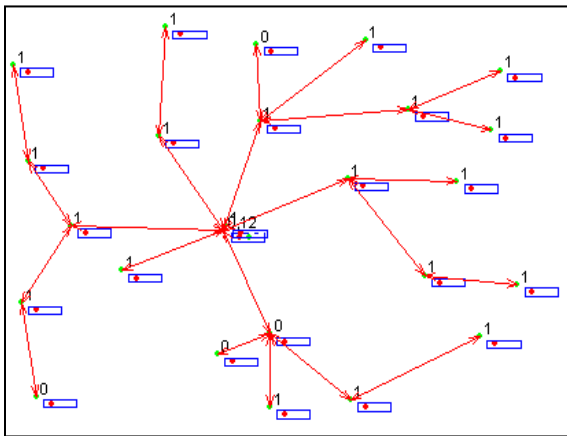
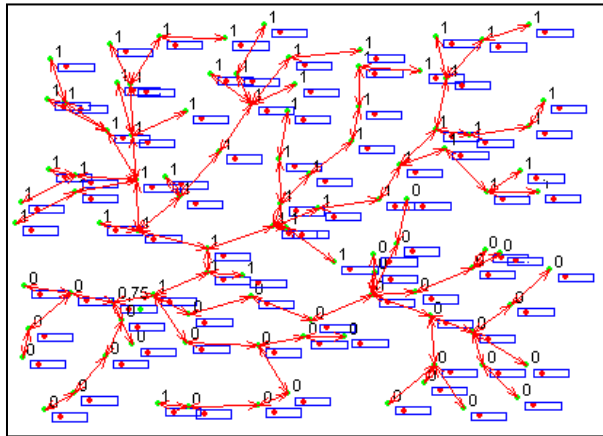

Fig 16. Topology with 25 nodes

Fig 17. Topology with 100 nodes

5.2.Simulation Results

In this simulation, we have mainly compared the performance of convergecast using multiple channels (2 channels) with the one using a single channel. The metrics used for measuring the performances are latency, successful rate, and throughput. Latency is the number of timeslots to complete the convergecast using either a single channel or multiple channels. Successful rate is calculated by actual number of packets delivered to the sink node over the initial total number of packets in all nodes where it is also equal to the number of sensor nodes. Throughput is calculated by taking an average of how many packets are delivered to the sink node per second.

Furthermore, as we have introduced in earlier chapters for conflicting map table of [7] in order to prevent sensor nodes in different branches from having collisions, we have also compared convergecast algorithm with and without conflicting map table, using the same metrics. Usage of conflicting map is significant since we do not want to have any collision in the real application deployment. Therefore, even though conflicting map bears an extra delay for the convergecast, it is necessary for the convergecast with a single channel or 2 channels to have it for real deployment in order to achieve more reliable convergecast. Again, our proposed convergecast using 4 channels do not require such conflicting map, and furthermore our convergecast is more delay efficient.

As shown Fig 18, latency using a single channel and multiple channels (2 channels) is compared. In the result, "1 CH" represents a convergecast with a single channel and "2 CH" represents a convergecast with 2 channels. In addition, "CM" represents using conflicting map to avoid collision among nodes in different branches. As we compare "2 CH" with "1 CH", convergecast with 2 channels clearly outperforms the one using a single channel. The difference

of delay efficiency increases when the number of hops increases, this result clearly says that having multiple channels significantly contributes to improve the delay efficiency.
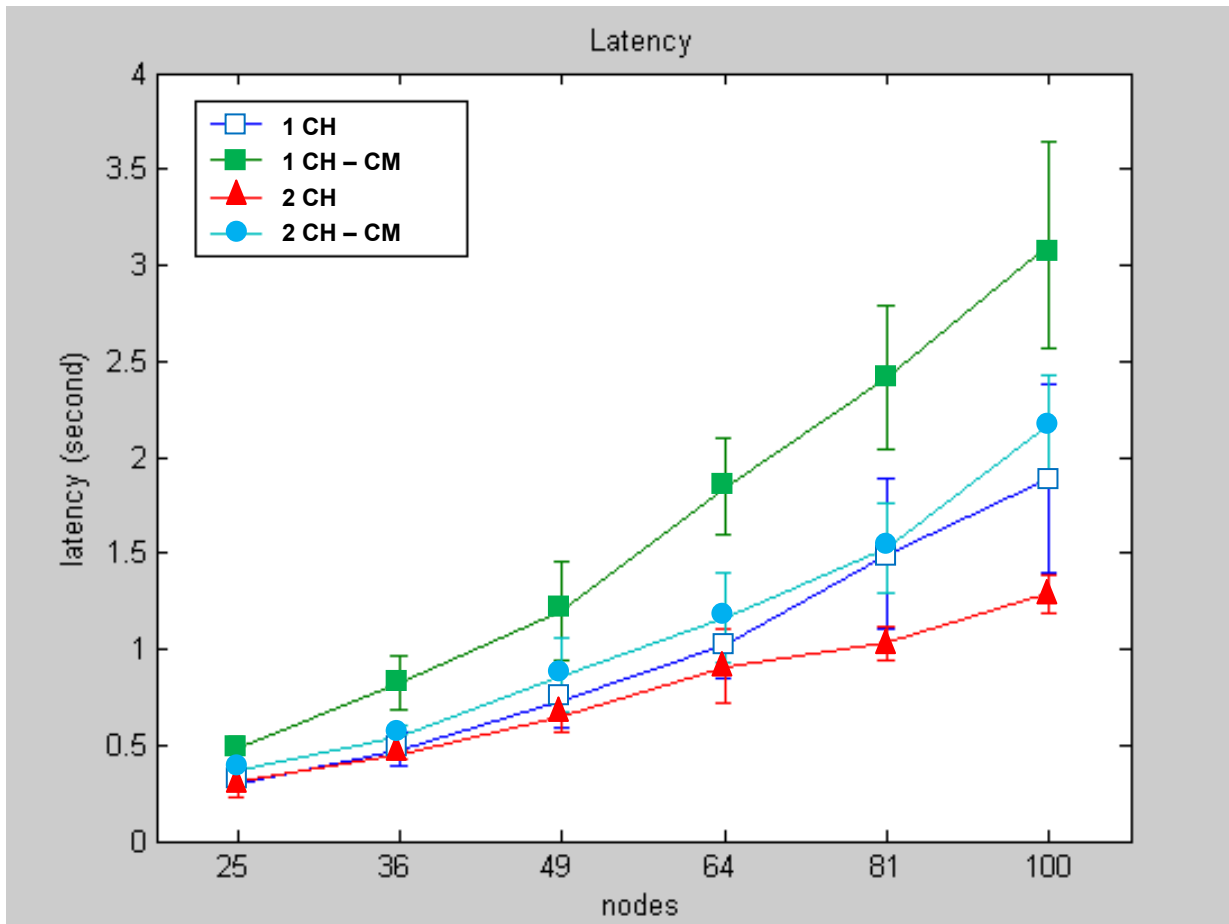

Fig. 18. Latency comparison of convergecast using a single channels (1 CH) and 2 channels (2 CH).

As shown Fig 19, successful rate using a single channel and multiple channels (2 channels) is compared. In this result, there is not much difference between algorithms since both algorithms are based on time scheduling; therefore, algorithms have almost the same successful

rate. Particularly, using conflicting map, both algorithms should be close to 100% of successful rate because of its contention free characteristics.
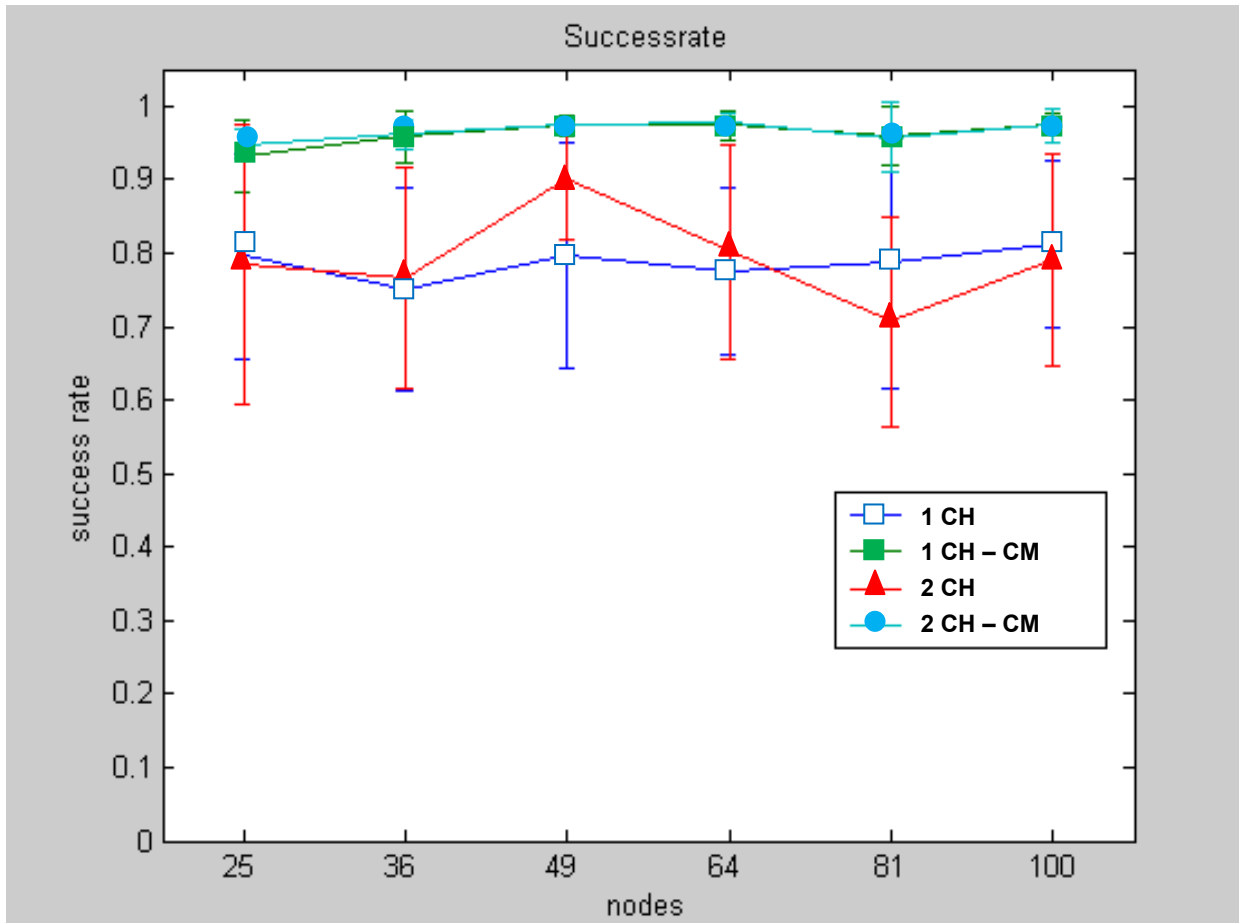

Fig 19 Successful rate comparison of convergecast using a single channels (1 CH) and 2 channels (2 CH).

As show in Fig 20, in terms of throughput, convergecast using multiple channels (2 channels) outperforms the one using a single channel. This is because, in multiple channels, as long as branches are forwarding the packets to the sink node, all the nodes in such branches are either transmitting the packets or receiving packets; however, in the convergecast with a single channel, there are always idle nodes existing in such branches beside transmitting and receiving

44

nodes. As summary, from this result, we can clearly see that using multiple channels outperforms using a single channel in large multi hop environment.
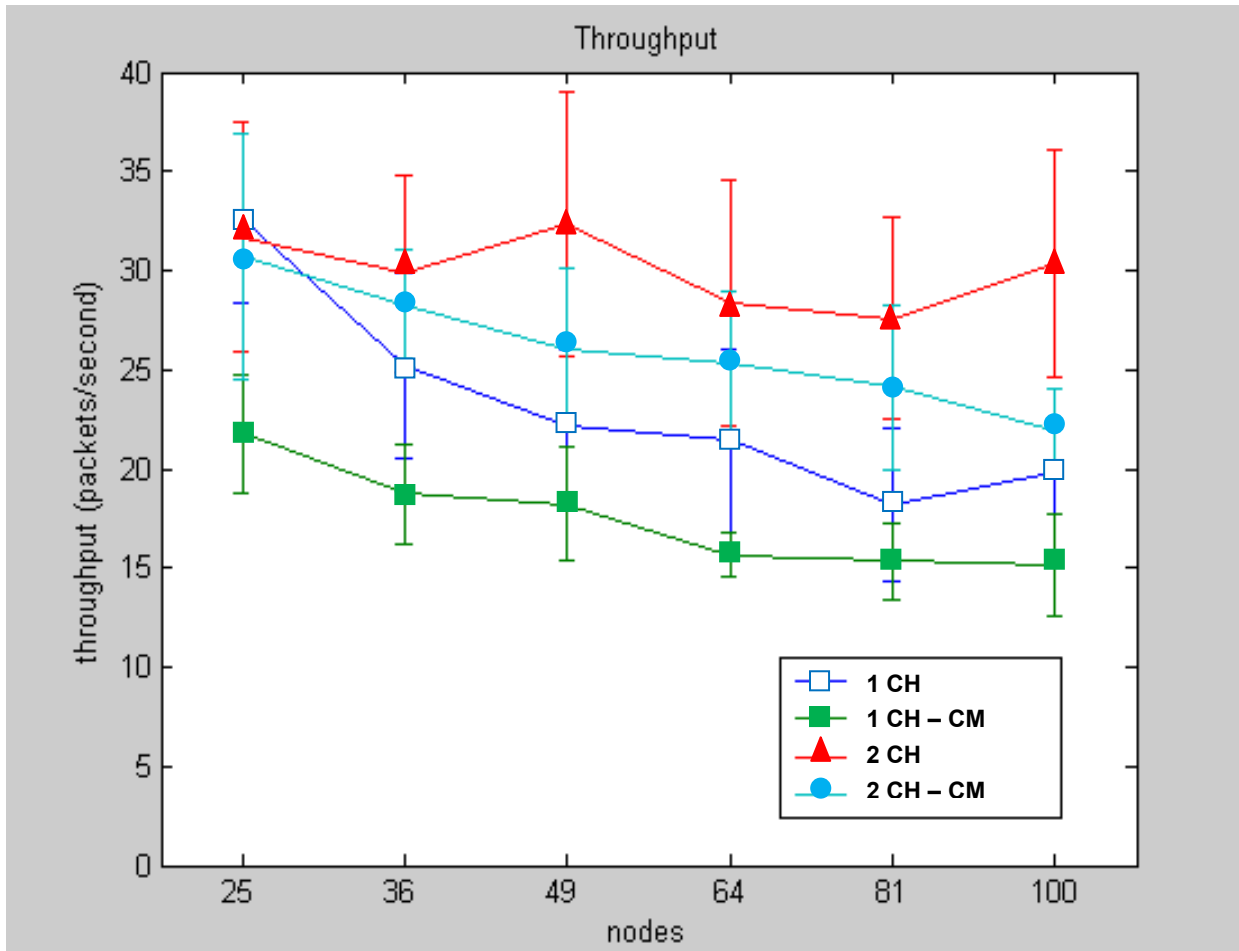

Fig. 20 Throughput comparison of convergecast using a single channels (1 CH) and 2 channels (2 CH).

Chapter 6

CONCLUSION AND FUTURE WORK

6.1.Conclusion

Unlike studying convergecast with a fixed number of channels and fixed number of packets, we have generalized channel assignment and convergecast algorithm with multiple channels, multiple packets at each node, and with a single half duplex radio transceiver. We also study the significance of multiple channels conducting extensive simulation by comparing convergecast using 2 channels with the one using a single channels in general topology. Convergecast algorithm with multiple channels outperforms the one with a single channel in latency and throughput. As further steps, we propose convergecast algorithms with a novel channel assignment technique in various sensor networks finding that the number of channels required completing convergecast is at most 4 in general topology with our scenarios and assumptions. We also found that usage of 4 channels contributes to not only delay efficiency but also solves other wireless sensor networks issues such as reducing significant amount of memory consumption. Without 4 channels, convergecast algorithm requires an internal table such as conflicting map for avoiding collision among nodes in different branches.

6.2.Future Work

As future work, several interesting works can be expected. First, considering energy consumption which is one of the major issues in sensor networks, we investigate how to save more energy using multiple channels. For example, in general topology only two branches forward packets to the sink node; therefore nodes in other branches should turn off the radio to save energy and also finding other ways of reducing energy with multiple channels. Trade off

between energy and latency is also another concerned, for example, the more the number of transmission increases, the more energy is consumed with less latency.

Another interesting work is that increasing the number of sink nodes. Take an example of using two sink nodes. Since we use multiple channels, at most 4 branches can forward packets at the same time. This has to be done with multiple channels and will significantly contribute on throughput of convergecast especially when density of sensor network is high or when data is intensively demanded.

REFERENCES

LIST OF REFERENCES

[1] Bhaskar Krishnamachari, Deborah Estrin, and Stephen Wicker , "The Impact of Data Aggregation in Wireless Sensor Networks", Proceedings of the 22nd International Conference on Distributed Computing Systems, Washington, DC, 2002

[2] G.Tia, G.Dan, W.Matt, "Improving Patient Monitoring and Tracking in Emergency Response," In Proceedings of the International Conference on Information Communication Technologies in Health, July 2005.

[3] B.R.Chris, A.Kenneth, B.Simon, B.Merwan, B.Vikas, B.Nathan, M.D.Artin, D.Gunes, G.Lilia, H.M.Brent, H.Christine, K.Mike, M.Jennifer, R.Stefanie, R.Matt, S.Miclas, S.Christo, S.Hanns, W.Andrew, L.S.Eli, P.Trevor, W.K.Paul, "Wireless Sensor Networks for Home Health Care," Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, p.832-837, May 21-23, 2007.

[4] C.Walter, S.Kris, D.Nicolas, D.Adam, "Satellite based wireless sensor networks: global scale sensing with nano- and pico-satellites," Proceedings of the 6th ACM conference on Embedded network sensor systems, November 05-07, 2008, Raleigh, NC, USA.

[5] Wai-Leong Yeow, Chen-Khong Tham, and Wai-Choong Wong, "A novel target movement model and energy efficient target tracking in sensor networks" Vehicular Technology Conference, IEEE 61st   VTC 2005, Spring. 2005

[6] Mauri Kuorilehto, Marko Hannikainen, and Timo D. Hamalainen, "A survey of application distribution in wireless sensor networks", EURASIP Journal on Wireless Communications and Networking, Volume 2005 , Pages: 774 - 788 , October 2005

[7] Shashidhar Gandham, Ying Zhang, and Qingfeng Huang, "Distributed time-optimal scheduling for convergecast in wireless sensor networks", Computer Networks, Publisher Elsevier, Vol. 52, Iss. 3, 2008

[8] Xianjin Zhu, Bin Tang, and Himanshu Gupta. "Delay Efficient Data Gathering in Sensor Networks", Proceedings of the International Conference on Mobile Ad-hoc and Sensor Networks (MSN'05), Lecture Notes on Computer Science, Vol. 3794, pages.380-389, Wuhan, China, December 2005.

[9] Haibo Zhang, Pablo Soldati and Mikael Johansson, "Optimal Link Scheduling and Channel Assignment for Convergecast in Linear Wireless HART Networks", Proceedings of the 7th international conference on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, Piscataway, NJ, USA, 2009

[10] Hua-Wen Tsai and Tzung-Shi Chen, "Minimal Time and Conflict-Free Schedule for Convergecast in Wireless Sensor Networks", Proceedings of the IEEE International Conference on Communications, Beijing, China, May 2008

[11] Yafeng Wu, John A. Stankovic, Tian He, Jiakang Lu, and Shan Lin, "Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks", Proceedings of The 27th Conference on Computer Communications on INFOCOM, Rio de Janeiro, Brazil, April 2009

[12] Y. Zhang, M. Fromherz, and L. Kuhn. "Rmase: Routing modeling application simulation environment", found on January 15, 2010 at http://www.parc.com/era/nest/Rmase

[13] G. Simon. "Probabilistic wireless network simulator", found on January 15, 2010 at http://www.isis.vanderbilt.edu/projects/nest/prowler

[14] Jie Gao, Leonidas Guibas, Nikola Milosavljevic, and John Hershberger, "Sparse data aggregation in sensor networks", Proceedings of the 6th international conference on Information processing in sensor networks, Cambridge, Massachusetts, 2007

[15] Zhang, Y and Huang, Q. "Coordinated convergecast in wireless sensor networks", IEEE Military Communications Conference (MilCom05), Atlantic City, NJ. NY, October 2005

[16] Baljeet Malhotra, Ioanis Nikolaidis, and Mario A. Nascimento, "On Maximizing the Throughput of Convergecast in Wireless Sensor Networks", IEEE Military Communications Conference (MilCom05), Atlantic City, NJ. NY, October 2005

[17] Ghosh, A, Incel, O.D, Kumar, V.S.A, Krishnamachari, B.mento, "Multi-channel scheduling algorithms for fast aggregated convergecast in sensor networks", IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, Macau China, October 2009

[18] Tzung-Shi Chen, Hua-Wen Tsai, and Yu-Hsin Chang, "Geographic convergecast in wireless sensor networks", Wireless Days. 1st IFIP, 2008

[19] Subramanian R, Chan K, and Fekri F, "Analysis of latency in secure wireless sensor networks with key predistribution", 42nd Asilomar Conference on Signals, Systems and Computers, Atlanta, GA, 2008

[20] Pal Y, Awasthi, L.K, Singh, A J, "Maximize the Lifetime of Object Tracking Sensor Network with Node-to-Node Activation Scheme", IACC 2009. IEEE International Advance Computing Conference, Patiala India, March 2009

[21] Naderan M, Dehghan M, Pedram H, "Mobile object tracking techniques in wireless sensor networks", International Conference on Ultra Modern Telecommunications & Workshops", St. Petersburg Russian, 2009

[22] Kam C, Hodgkiss, W.S., "Distributed Target Tracking in a Wireless Sensor Network", International Conference on Ultra Modern Telecommunications & Workshops", Pacific Grove, CA, 2006

[23] Shashidhar Gandham, Ying Zhang, and Qingfeng Huang, "Distributed Minimal Time Convergecast Scheduling in Wireless Sensor Networks", Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06), Lisbon, Portugal, July 2006

[24] Ozlem Durmaz Incel, Amitabha Ghosh, and Bhaskar Krishnamachari "Scheduling Algorithms for Tree-Based Data Collection in Wireless Sensor Networks", Theoretical Aspects of Distributed Computing in Sensor Networks, Eds. S. Nicoletseas, J. Rolim, Springer Verlag, 2010

[25] Amitabha Ghosh, Ozlem Durmaz Incel, V.S. Anil Kumar, and Bhaskar Krishnamachari, "Multi-Channel Scheduling Algorithms for Fast Aggregated Convergecast in Sensor Networks", IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS), October 2009, Macau, China

[26] Ozlem Durmaz Incel, Amitabha Ghosh, Bhaskar Krishnamachari, and Krishna Kant Chintalapudi, "Multi-Channel Scheduling for Fast Convergecast in Wireless Sensor Networks", CENG Technical Report, CENG-2008-9

[27] Ju Wang Hongsik Choi, Esther A. Hughes, "Scheduling on Sensor Hybrid Network", IEEE International Conference on Computer Communications and Networks, 2005

[28] Choi Hongsik, Wang Ju, Hughes Esther, "Scheduling for Information Gathering on Sensor Network", ACM Jounal of Wireless Networks, 15:127-140, January 2009.