A Communication-Optimized Distributed 1-D FFT

on Wireless Sensor Network Array

BY

AVINASH ATHIRASALA
B.S. Equivalent, Jawaharlal Nehru Technological University, 2010

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2012

Chicago, Illinois

Defense Committee:

Rashid Ansari, Chair and Advisor
Ashfaq Khokhar, Advisor
Ajay Kshemkalyani, Computer Science

To my parents, Subramaniam and Prasuna and sister Avathamsa whose unconditional love and unwavering support motivated me and made this possible.

ACKNOWLEDGEMENTS

I wish to thank my advisors Prof. Rashid Ansari and Prof. Ashfaq Khokhar from UIC for their invaluable guidance and immeasurable patience in assisting me with this thesis. I would also like to acknowledge the contributions made during the course of many discussion sessions by the members of the Wireless Sensor Network research group at UIC.

I would also like to thank the Department of Electrical and Computer Engineering at the University of Illinois at Chicago for giving me the opportunity to be a part of the Masters' program and to participate in the active research.

I owe my deepest gratitude to my parents Subramaniam and Prasuna, and sister Avathamsa for their constant encouragement and to my uncle Prasad for his advice and direction throughout the course of my education without which this thesis would not be possible.

A special thanks to my friend Jahnavi for her invaluable suggestions made towards the writing of this thesis report.

AA

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

LIST OF ABBREVIATIONS

WSN	Wireless Sensor Network
DSP	Digital Signal Processing
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
DIT	Decimation-In-Time
RF	Radio Frequency
ADC	Analog to Digital Converter
CPU	Central Processing Unit
RAM	Random Access Memory
USB	Universal Serial Bus
IEEE	Institute of Electrical and Electronics Engineers
MAC	Medium Access Control
1-D	One-Dimension

SUMMARY

The problem of an efficient distributed load sharing in-network implementation of one-dimensional Discrete Fourier Transform (DFT) on a Wireless Sensor Network (WSN) is addressed in this thesis. It is assumed that the sensors nodes are placed in a uniform grid. The WSN collaboratively senses the spatial-temporal data of a physical phenomenon. Discrete Fourier Transform, implemented with a Fast Fourier Transform (FFT) algorithm is the most commonly used tool for frequency analysis in Digital Signal Processing (DSP), which makes it suitable to be used for the analysis of the data sensed by the WSN. In-network collaborative processing of DSP algorithms is much more power efficient and has shorter computation latency than a data aggregated processing in a sink node.

The communication exchanges required for the load sharing in-network processing of the 1-D FFT need to be optimized as radio communication consumes the largest fraction of battery power in commonly performed signal processing tasks. The power-efficient minimal latency algorithm also needs to be well distributed in its communication and computational load.

Two in-place algorithms for a similar distributed in-network processing of 1-D FFT have been put forward previously. The first algorithm proposed in (1) is a conventional in-place approach in which the communication exchanges and the sensor computations are modeled on the input in-order flow-diagram of decimation-in-time (DIT) radix-2 FFT. The second approach proposed in (2) is modeled instead on the standard flow chart of DIT radix-2 FFT where the inputs are arranged in a bit-reversed order and additionally proposes that after half the stages are completed the data-points held by the sensors be shuffled by a bit-complement bit-reversal of their order. This shuffling result in bringing the data-points required for computation in the remaining half of the stages closer to each other and also balances out the computational load overall. The second approach is shown to be better than the first in terms of power-time efficiency but it requires the input to be pre-arranged into a bit-reversed order. The communication exchanges required for this pre-arrangement of the input into a bit-reversed order is not discussed and the very significant cost of power and time consumed by it is not considered.

SUMMARY (continued)

The proposed reduced-communication implementation of 1-D FFT on WSN array builds on the conventional in-place approach and computes the algorithm with the input data-points in-order. In the conventional in-place approach several redundant communication exchanges are seen to take place just to ensure that all the N sensor nodes participate in every stage of the FFT computation. In the proposed reduced-communication approach this redundancy in communications is eliminated by judiciously selecting $\frac{N}{2}$ sensor nodes to do the butterfly computation for each of the $\frac{N}{2}$ butterfly pairs of data points for every stage of the FFT computation. As every kilobit of radio communication consumes power that is several times higher than required to execute a million CPU instructions in a wireless sensor node, eliminating redundant communications at the much smaller cost of a marginally non-uniform distribution of the computation load results in an overall significant improvement in power-efficiency and lengthened network life time. Also a more optimized communication exchange reduces the number of packet collisions and processing latency. Experimental results and theoretical comparisons with the conventional in-place approach show that the proposed reduced-communication algorithm gives us shorter latencies at significantly lower energy costs and extends the network lifetime.

CHAPTER 1

INTRODUCTION

Equipping sensors with wireless network interfaces gives them an additional ability to network and exchange information. As a network these sensor nodes are suited to collaboratively monitor physical phenomena through a distributed sensing of spatial-temporal data. The wireless nature of the network gives it several degrees of flexibility that a wired interface does not possess. This wirelessly inter-connected network of sensing nodes is termed as a Wireless Sensor Network (WSN).

The small size and low cost of wireless sensor nodes along with the scalability and flexibility of the WSN allows for an unprecedented scope for dense deployment close to diverse physical phenomena and the possibility for wide-ranging applications.

Each sensor node also comes integrated with an on-board processor that can perform low-complexity computations. The capacity of the sensors nodes to carry out computation along with being able to communicate with one and another gives the network an added capability of processing, inferring, and acting upon the sensed distributed spatial-temporal data in real-time.

The small size of the wireless sensor nodes and the flexibility of the WSN accrue from powering the wireless sensor nodes with their own individual on-board batteries. The fundamental challenge associated with the design and operation of a WSN is the limited power supplying capacity and lifetime of the wireless sensor nodes' batteries. Restricted power supply in turn results in diminished processing capability, memory storage and also communication range and throughput of the nodes. Constrained processing and memory resources of individual nodes translate into their being ill-equipped to carry out computationally demanding algorithms. One alternative is to aggregate data into a sink node which is better able to handle the computationally complex algorithms such as those of Digital Signal Processing (DSP). Short communication ranges and throughput along with relatively high power consumption cost of communications makes data aggregation into a sink node an unviable option due to its high power consumption and the higher latency involved. A better alternative is a load-sharing approach to implementing algorithms by the local sensing nodes themselves which is shown to have better power-efficiency and shorter computation latency (1, 10). This is referred to as an in-network processing or local processing in the WSN. An efficient distributed in-network processing would imply a collaborative computing supported by an reduced-communications scheme where in both the communication and the computational load are well distributed among the wireless sensor nodes.

Digital signal processing of the sensed data plays a fundamental role in WSN applications (19). One of the common tools for interpretation of the sensed data is its frequency analysis. Fourier transform is widely recognized as an effective means for frequency analysis and has matured over several decades (6). There are several other alternatives to Fourier transform popular among them being wavelet analysis. Fourier transform is best suited for frequency analysis of signals in applications that requires either a high resolution of frequency for a low time resolution of the signal else a low resolution of frequency for a high time resolution of the signal. The resolutions of time and frequency having an inverse relationship while operating with Fourier transform. For applications requiring a high resolution of frequency analysis of a high time resolution signal wavelet analysis is more appropriate (22). Time dimension can suitably be replaced with a spatial dimension depending upon the type of signal.

In WSN the sensed data is a discretized spatial-temporal signal. Discrete Fourier Transform (DFT) provides a framework for frequency analysis of discretized spatial-temporal signals (20). Fast Fourier Transform (FFT) refers to an efficient algorithm for computing of DFT (6). An efficiently designed in-network processing of FFT would transform the capabilities of the network, by allowing the network to interpret sensed data and respond to it in real-time.

The main objective of this thesis is to design an efficient load-sharing in-network implementation of N-point 1-D FFT processing of N single spatial data points sensed at the same fixed sampling time interval by a WSN array, with the sensor nodes arranged on a uniform - grid. The 1-D conventional FFT on spatial data can only be computed if the data is equally-spaced, hence the uniform-grid arrangement is necessitated. As FFT is most commonly implemented for data sets of powers of two, N is assumed to be a power of two, i.e. $N = 2^m, \ m \epsilon Z^+$.

The shared concurrent computation of the 1-D FFT needs to be facilitated through a well-defined communication algorithm among the wireless sensor nodes. The communication algorithm among the sensor needs to be designed to curtail energy costs along with minimizing latency (10). The communicational load needs to be well-distributed and the number of radio communications taking place should be optimal along with a limited possibility of them colliding.

Two in-place algorithms for a similar distributed in-network processing of 1-D FFT have been put forward previously. The first algorithm proposed in (1) is a conventional in-place approach in which the communication exchanges and the sensor computations are modeled on the input in-order flow-diagram of decimation-in-time (DIT) radix-2 FFT. The second approach proposed in (2) is modeled instead on the standard flow chart of DIT radix-2 FFT where the inputs are arranged in a bit-reversed order and additionally proposes that after half the stages are completed the data-points held by the sensors be shuffled by a bit-complement bit-reversal of their order. This shuffling result in bringing the data-points required for computation in the remaining half of the stages closer to each other and also balances out the computational load overall. The second approach is shown to be better than the first in terms of power-time efficiency but it requires the input to be pre-arranged into a bit-reversed order. The communication exchanges required for this pre-arrangement of the input into a bit-reversed order is not discussed and the very significant cost of power and time consumed by it is not considered.

The proposed reduced-communication implementation of 1-D FFT on WSN array improves upon the conventional in-place approach and computes the algorithm with the input data-points in-order. In the conventional in-place approach several redundant communication exchanges are seen to take place just to ensure that all the N sensor nodes participate in every stage of the FFT computation. In the proposed reduced-communication approach this redundancy in communications is eliminated by judiciously selecting the $\frac{N}{2}$ sensor nodes to do the butterfly computation for each of the $\frac{N}{2}$ butterfly pairs of data points for every stage of the FFT computation.

The WSNs are mainly used in industries, infrastructure management, military scenarios, health care, consumer products and environmental protection (14). Some of the applications include process automation, building management, patient monitoring, automobile maintenance, battlefield surveillance, crop monitoring, environmental hazard prevention and management, traffic control etc.

The thesis is organized as follows. Chapter 2 of this thesis provides the background about DFT, FFT and WSNs. Chapter 3 contains an explanation of previous related published work. Chapter 4 provides a thorough explanation of the proposed reduced-communication implementation of the FFT algorithm on WSN. Chapter 5 presents theoretical and experimental comparison with the same already existing works explained in chapter 3 which shows very significant improvements obtained. Chapter 6 contains the conclusion and description of related possible future work in the same field.

CHAPTER 2

BACKGROUND

2.1 Discrete Fourier Transform

Fourier analysis decomposes spatial or temporal signals are into their constituent harmonically related sinusoidal or complex exponential components. The resultant components are termed as the frequency domain representation of the signal and they represent the spectral characteristics of the signal.

Discrete Fourier Transform (DFT) provides the Fourier analysis of a signal that is discretized in time or spatial signals into their frequency domain representation.

Let the N finite data points sensed at equally spaced discrete 1-D spatial points be represented as $x_n, 0 \leq n \leq N-1$, where $N = 2^m,\ m \epsilon Z^+$.

The N-point DFT (20) of the x_n sample data points is given by

$$X_k = \begin{Bmatrix} \sum_{n=0}^{N-1} x_n W_N^{kn}, \quad W_N^{kn} = e^{\frac{-j2\pi kn}{N}} & k = 0,1,..,N-1 \\ 0 & \text{elsewhere} \end{Bmatrix} \tag{2.1}$$

The resultant frequency component data points, $X_k, 0 \leq k \leq N-1$ corresponds to the signal content at the frequencies $\frac{2\pi k}{N}$ in the frequency domain.

2.2 Fast Fourier Transform

The Fast Fourier Transform (FFT) algorithm is an efficient method of computing DFT by minimizing the number of computations required by taking advantage of the properties of trigonometric functions (20).

There are several types of FFT algorithms, for the purpose of this thesis we focus on Decimation in Time (DIT) radix-2 FFT approach.

By splitting the index of the summation in Equation 2.1 into odd and even terms, we get

$$X_k = \sum_{p=0}^{N/2-1} x_{2p} W_N^{2pk} + \sum_{p=0}^{N/2-1} x_{2p+1} W_N^{(2p+1)k} \quad k = 0,1,..,N-1 \tag{2.2}$$

Let u represent the even terms of x and v represent the odd terms, i.e $u_p = x_{2p}$ and $v_p = x_{2p+1}$

$$X_k = \sum_{p=0}^{N/2-1} u_p W_N^{2pk} + W_N^k \sum_{p=0}^{N/2-1} v_p W_N^{2pk} \quad k = 0,1,..,N-1 \tag{2.3}$$

From the properties of complex exponentials

$$W_N^2 = W_{N/2} \tag{2.4}$$

Substituting the property shown in equation 2.4 in equation 2.3

$$X_k = \sum_{p=0}^{N/2-1} u_p W_{N/2}^{pk} + W_N^k \sum_{p=0}^{N/2-1} v_p W_{N/2}^{pk} \quad k = 0,1,..,N-1 \tag{2.5}$$

Each term in equation 2.5 represents the formula for a $\frac{N}{2}$ point DFT, the same formula as in equation 2.1. Let U and V denote the respective $\frac{N}{2}$ point DFTs of u and v.

Using the properties

$$W_N^{k+N/2} = -W_N^k \tag{2.6}$$

$$W_{N/2}^{N/2} = 1 \tag{2.7}$$

Equation 2.5 can now be written as

$$\mathrm{X_k} = U_k + \mathrm{W_N^k} V_k \quad k = 0,1,2..,\frac{N}{2} - 1$$
$$\mathrm{X_{k+N/2}} = U_k - \mathrm{W_N^k} V_k \quad k = 0,1,2..,\frac{N}{2} - 1 \tag{2.8}$$

Equation 2.8 represents the computation of X_k by using the computed DFTs of $\frac{N}{2}$ even and odd terms of x_k i.e. U_k and V_k. The same procedure can again be used for computing the DFT terms of U_k and V_k by splitting u_k and v_k each into their respective odd and even terms and computing the $\frac{N}{4}$ point DFTs those terms. This recursive process can be continued till it is reduced to a two point DFT. This results in $log_2 N$ stages of computation.

So starting from a two point DFT recursively in $log_2 N$ stages we can compute the N point DFT of the N sensed data points by following the procedure described above.

The pair of computations shown in equation 2.8 for any fixed data point k is called a butterfly computation. It is represented as a flow chart in figure 1. The flow chart is referred to as a butterfly diagram. The factor $\mathrm{W_N^k}$ is termed as a twiddle factor.

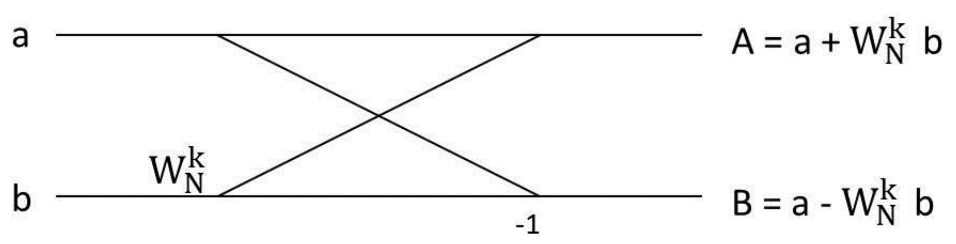

Figure 1. Butterfly diagram for DIT FFT

A 16-point DIT FFT computation is illustrated in the form of a flow diagram in figure 2. Note that in the first stage the input terms are arranged in a bit-reversed order of their actual spatial order. An alternative flow chart for a 16-point DIT FFT is shown in figure 3 with the input data point arranged in-order. Note that figures 2 and 3 represent the same DIT computation with the same pairs of data points computed together in a butterfly computation and having the same twiddle factors, they only vary in the arrangement of the data points.

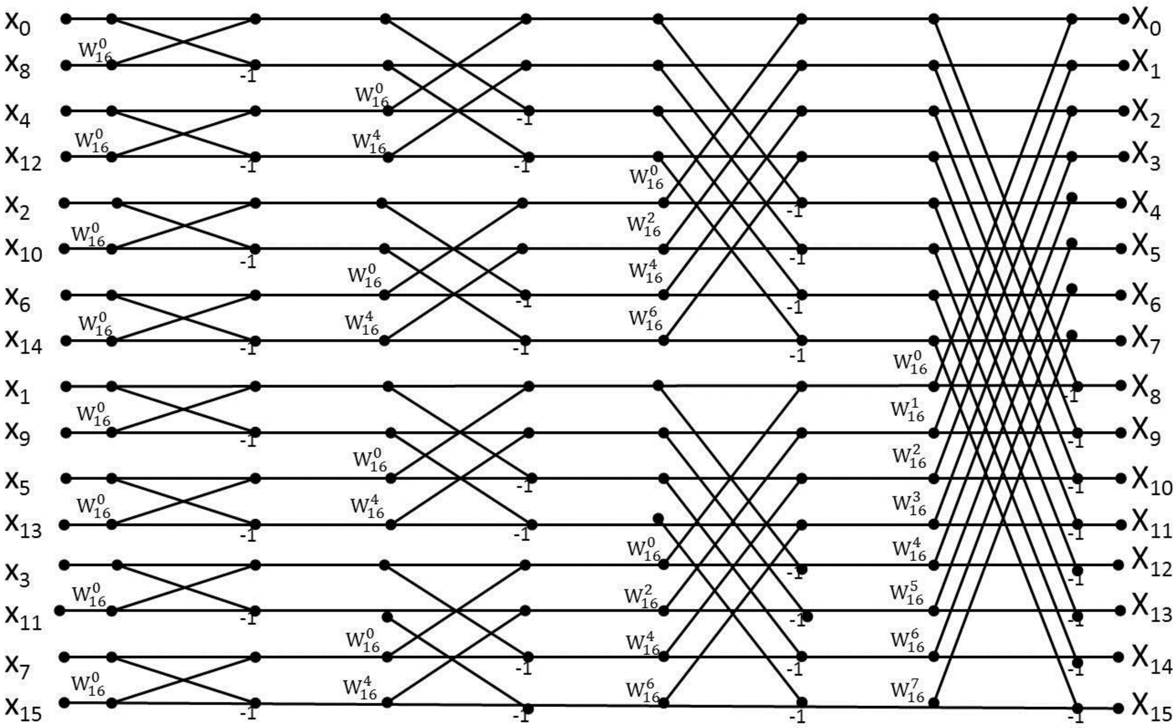

Figure 2. DIT FFT flow chart with input in bit reversal order.

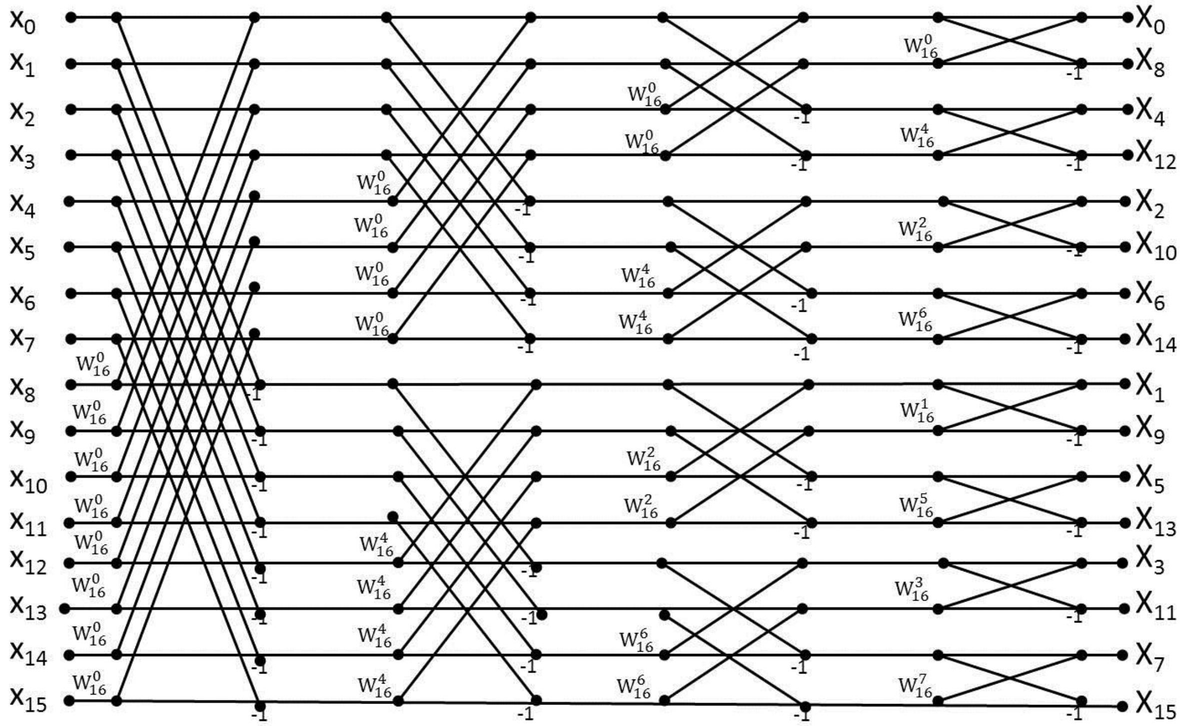

Figure 3. DIT FFT flow chart with input in order.

2.3 <u>Wireless Sensor Networks</u>

Wireless Sensor Network (WSN) consists of sensor nodes each equipped with a wireless network interface thereby collaboratively inter-connected to each other. The size of the network varies with the application and is scalable from few to several thousands. The main impact of the WSN is through the distributed coordinated activity of sensing, processing and networking by all the sensor nodes and thereby overcoming the limitations in resources of each individual node.

The operation and design of a WSN encompass several fields including but not restricted to embedded systems, digital signal processing, data acquisition, distributed algorithms, network and database management.

The cost, size and capabilities of individual sensor nodes vary based on the terrain and the applications. Their flexibility, heterogeneity and scalability allow them to be easily and densely deployed close to physical phenomena for the purpose of monitoring through sensing and also control of the phenomena through real-time processing of the sensed data. They are especially designed to operate unattended and with limited maintenance, this makes them well suited for hazardous inaccessible environments.

2.3.1 Challenges associated with Wireless Sensor Networks

1. **Power consumption**: Wireless sensors are deployed densely in remote locations and their operation is unattended. For this reason they are battery equipped and usually these batteries cannot be replaced. Due to the size and weight limitations of the wireless sensor node the battery power capacity available to the unit is very scarce. The battery life of the nodes defines the lifetime of the network.

   In this thesis the main focus is on the in-network processing energy costs. There are two types of costs associated with in-network processing one being the CPU cost for data processing and the other the radio cost for communication exchange.

   - Radio cost: The maximum usage of power is usually for the radio communication unit in the sensor node. The radio transceiver consumes the largest power during transmission, reception, and also when it is in an idle state. During transmission energy is taken up by power amplifiers. So transmission power consumption varies with the range of transmission. During reception decoding circuitry is a major consumer of power as it carries out error detection and correction in addition to extraction of data from the encoded message. For short range low-power communication the transmission energy cost is roughly same as that of the reception cost.

     The energy consumed by the radio unit in sleep state is approximately 99% less than the energy consumed while it is in the active idle state. The transition from one mode to another mode of the radio transceiver consumes time and energy, but they are designed to be power-efficient in the case of wireless nodes. So the transceiver unit can be brought to active state only when required and is in the sleep state for the rest of the time to conserve power. The transition time and energy becomes a nonnegligible quantity in the case of smaller packets and frequent radio activity. The transition from sleep to active state, called startup time, is about hundreds of micro-second. Care needs to be taken to decide when a frequent transition of the radio unit between sleep and active modes is suitable and when it is preferable to leave the radio unit in active state constantly.

The total radio energy cost P_r can be computed as the sum of the transmission cost P_{tx}, reception cost P_{rx}, idle in active state cost P_i and sleep state cost P_s.

$$P_r = P_{tx} + P_{rx} + P_i + P_s \tag{2.9}$$

- Processor cost: The computation energy cost per instruction executed is drastically less than communication cost per Kb transmitted. Short range low-power communication is associated with Rayleigh fading and fourth-power loss with transmission range. So in such a case the energy that is consumed to transmit a 1 kb packet over a range of 100m is approximately equal to that consumed by the CPU to execute 3 million instructions (13). So in a trade-off between communication and computational energy costs, minimizing communicational energy costs provide several times more saving of power. This also reiterates the advantage of a local in-network processing versus a data aggregated processing in a sink node requiring several multi-hop communications.

  The total CPU cost P_{CPU} can be computed as a sum of the energy cost of the active CPU P_{CPU_Active} and the energy cosumed by the CPU in idle state P_{CPU_Idle}

$$P_{CPU} = P_{CPU\_Active} + P_{CPU\_Idle} \tag{2.10}$$

2. **Scalability of algorithms and protocols**: The main advantages of WSN stems from the fact that wireless sensors can be densely deployed and the size of the network along with the topology are flexible. To support the varying densities of the WSN, all the distributed processing algorithms and networking protocols implemented on the network need to be

scalable from few to several thousand nodes. For this reason specialized algorithms and protocols need to be designed specifically for WSN and cannot be directly ported from those of other wireless networks.

3. **Unattended operation and low scope for maintenance:** All the hardware, algorithms and protocols required for the operation of the WSN needs to be autonomous and robust. This is because most applications of WSN are unmanned and in locations where maintenance is difficult and expensive.
4. **Topology:** Topology is very important for DSP as it affects the properties of the data being sensed, which in turn decides the processing algorithm to be applied. Topology also affects how well computational and communicational load can be distributed in the case of an in-network processing of the algorithm.
5. **Radio communication:** Low power RF communication is the standard means of communication in WSN, the advantage being the saving in power consumed. Most of the standard hardware platforms come equipped with low power RF radio unit. Low-power RF communication has the disadvantage of low data rate and high path loss leading to short transmission range. But this disadvantage of low data rate and high path loss is exploited in WSN as this allows for frequency reuse.
6. **Multi-hop routing:** Most commonly in a WSN the sensors are linked to each other through a multi-hop routing. The limited range of the low-power RF radio communication, dense deployment of sensor nodes and possible obstacles to communication within the terrain of the network are the reasons for multi-hop routing being the standard in WSN instead of a single-hop direct routing between source and destination in WSN. Multi-hop routing is illustrated in figure 4.

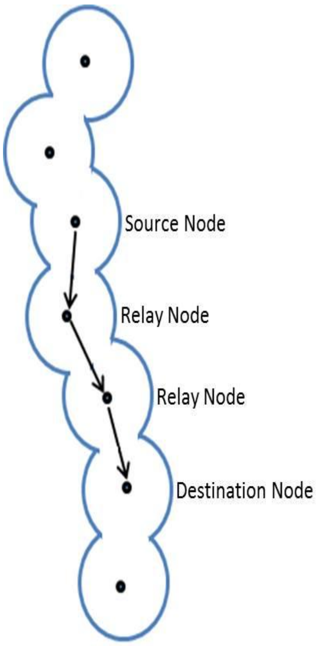

Figure 4. Illustration of multi-hop routing.

7. **Performance metric:** Most networks performances are measured by their throughput, uptime and computational latency. For WSN network lifetime extension through power conservation is the primary performance metric. Computational and Communication latency form secondary performance metrics.

2.3.2 <u>Wireless Sensor Node Hardware</u>

Hardware components of wireless sensor nodes:

Wireless sensor nodes can broadly be classified to have the following components, though they vary widely based on their size requirements, the type of application and cost.

1. Multiple types of sensors
2. Analog to Digital Converter (ADC)
3. Microcontroller for computation along with programming interfaces like USB, Serial Port etc.
4. Low capacity RAM and flash memory
5. Low data rate wireless transceiver, and low range antenna
6. Battery power source

Hardware constraints of wireless sensor nodes:

1. **Size:** The size of the nodes is usually very small and varies depending on the application. Sometimes the entire embedded system needs to be fitted into a coin sized unit or even a cubic-centimeter unit. The weight of the sensor nodes is also an important factor, as the nodes may need to be very light depending upon the physical phenomena being monitored. Thus the hardware design needs to be done keeping the size and weight of the sensor node in mind. The hardware unit will need to be autonomous and robust, needing no maintenance and manual intervention. Smaller and more sophisticated the sensor node, fewer is the hardware resources available to it and higher is the cost.
2. **Power consumption:** Due to the deployment of the sensor nodes into remote regions they are usually powered by a limited battery supply. Limited power consumption is the single most important design consideration for a wireless sensor node and also in the network. The lifetime of the Network is determined by the battery life of the nodes. Size and cost constraints make power a very scarce and valuable resource. For example a Mica2 node uses two AA batteries, giving the node a limited battery capacity of 1400 - 3400 mAh.

   The transceiver unit of the sensor node is the major source of power consumption within the embedded system. Low-power RF communication is the standard in WSN chosen for its reduced power consumption even at the cost of poor throughput and high delay. The transceiver needs to perform modulation, filtering, demodulation and multiplexing all of which consume a lot of the available power. Low power ON and power OFF delay in the circuitry of the transceiver allows for the unit to be turned OFF when not in use which is the case for most of the time.

   Power limitation also results in the sensor node having comparatively poor processing capability and memory availability.

Wireless Sensor Node Hardware Platforms:

For the scope of this thesis which deals with in-network processing, only low-end platforms are applicable. Low-end platforms are used at the lowest level of the WSN hierarchy. They have a comparatively low-power supply resources thereby have limited processing and memory, shorter communication ranges and poorer data rates. Following are few of the common low-end Platforms and some their main features.

1. **Mica:**
   - The nodes are Mica, Mica2, MicaZ, and IRIS
   - 8-bit Atmel AVR microcontroller (4–16MHz) and (128–256 kB) programmable flash
   - RAM (4–8 kB) and data memory (512 kB)
   - Transceiver speed and data rate: 916/433MHz at 40 kbps(Mica), 433/868/916MHz at 40 kbps (Mica2), 2.4GHz at 250 kbps IEEE 802.15.4 compliant (MicaZ and IRIS)
2. **Telos/Tmote:**
   - Same transceiver and data memory as MicaZ .
   - Several integrated sensors along with the platform
   - 8MHz TI MSP430 microcontroller
   - 10 kB RAM
3. **EYES:**
   - 16-bit microcontroller with 60 kB of program memory
   - TR1001 transceiver data rate 115.2 kbps
   - Similar architecture to MicaZ and Telos/Tmote.
   - Several integrated sensors along with the platform

2.3.3 Wireless Sensor Network Software Platforms

The software platforms of the WSN need to be energy efficient and robust enough to handle the diverse challenges that come with the densely distributed nature of the network. TinyOS is the most common and widespread software platform used in the WSN research and industry. It is modeled to be event-driven and provides for component based architecture, both of which allows TinyOS platform to be flexible, power-efficient and requiring a reduced code-size for implementing protocols and algorithms. It is an open-source platform and much of the existing WSN software is on TinyOS platform. There are now several other newer and possibly more efficient software platforms for WSNs like LiteOS, Contiki and SunSPOT. But due to the large variations in the design and modeling of each of the software platforms, porting from one to another is a challenge and for this reason there is a continued persistence with TinyOS.

2.3.4 Protocol Stack

The unique challenges in networking that comes with WSN as described in section 2.3.1 needs to be addressed by the protocol stack. The aim is to lengthen the lifetime and improve the quality of the autonomous application, through support from the various layers of the networking protocol stack of the WSN. Due to short packet sizes the overhead contributed to the packet by each layer of the protocol stack becomes a point of consideration. WSN network protocol stack is very loosely defined and few standards exist due to the sharp variations in the design objectives of widely varying applications. This thesis deals with local digital signal processing on the WSN which forms a part of the application layer of the protocol stack.

1. **Physical Layer:** The physical layer deals with the wireless networking hardware that is used to interface and link the wireless sensor nodes. Signal generation and detection is the responsibility of the physical layer. The hardware associated with the physical layer is the radio unit of the sensor node which was discussed in section 2.3.2. The IEEE 802.15.4 standard for low-power wireless communication for the physical layer is usually adopted in WSN applications.
2. **Data Link Layer:** This layer ensures reliable data transfer within the WSN by defining the functionality and procedure for it. Functioning includes medium access control (MAC), data frame detection and generation and possible error control.

The MAC regulates the access to the shared radio channel to avoid interference between contending transmissions. Energy efficiency takes priority over the fairness and throughput efficiency with regards to channel allocation. The IEEE 802.15.4 standard for low-power wireless communication for the MAC layer is generally followed in several of the WSN applications.

3. **Network Layer:** Network layer defines the routing protocol to decide upon the path of the packet between source and destination nodes for the multi-hop communication from among several possible intermediate relay nodes. The shortest path routing passing through fewest possible nodes usually implies shorter communication latency and energy efficiency. Other consideration also being a requirement to balance energy consumption across the network to further the lifetime of the network. The routing protocols are usually data-centric meaning the route is decided base on the type and content of the data instead of it being decided based on the sensor node addresses.
4. **Transport Layer:** Transport layer protocols deal with external access to the network which is beyond the scope of this thesis.
5. **Application Layer:** Application layer is made up of the code for the application. The in-network processing of DSP algorithms would be a part of the application layer. Network management for the application is also a part of this layer. Network management includes time synchronization, power management, data management and also keeping track of neighboring nodes. In WSN network it is also shown that sometimes a cross-layered protocol stack instead of clearly defined layer functionality saves considerable code space and power.

2.3.5 Distributed In-Network Processing

In WSN the collaboration among the sensor nodes is the key principal to its applications. Every individual node by itself has an insignificant impact due to its negligible resources. It is the collaboration of these insignificant nodes that transforms the network into a very important tool as a link between the physical world and the cyber infrastructure. Power conservation is everything in

WSN. Also in applications shortest possible processing latency is critical for the network to respond to events in real time. The power and time consumed for data aggregation into sink nodes for the purpose of digital signal processing can be ill-afforded (1). Communication is the major consumer of power and data aggregation into a sink node requires several multi-hop communications. Also the time latency in performing this data aggregation into the sink node is very high along with the time that the sink node will consume to process the data. So an alternative load sharing collaborative processing of the sensed data by sensing nodes themselves is always favored. This is termed as an in-network processing.

The cost of communication energy cost is several times that of the computation but in the case of in-network processing of computationally complex DSP algorithms the energy footprint of computation cannot be ignored. For power-efficiency and network lifetime, the communication and the computational load has to be very well distributed among the sensors. In in-network processing usually there is trade-off in communication saving versus computation saving. Since communication energy cost is the significantly much higher than that of computation, minimizing communications saves much more of the power of the network. But an attempt needs to be made to ensure that the computational load is also as well distributed as possible as it is also a substantial energy cost in the case of DSP algorithms. Parallel computing of the algorithm using as much of the available computing resources of the network as possible without unnecessary additional communications is the goal of in-network processing.

CHAPTER 3

PREVIOUS WORK

3.1 Conventional In-Place Approach

A distributive implementation on WSN of DIT N-point radix-2 1-D FFT algorithm was put forward in (1) as a means to show that local processing is both faster and power-efficient when compared with a data aggregated processing in a sink node.

The implementation proposed follows an in-place approach. In every stage each sensor holds one of the N complex data points, assuming there are N sensors computing. For each stage computation, the pair of sensors holding data points in them that forms a butterfly pair, communicate and exchange their respective data points so that both sensors hold both data points of the butterfly pair in them for the purpose of computation. This communication exchange can be called a ‘butterfly exchange’. The communication approach and arrangement of data points within the sensors is similar to the flow chart for a 16-point FFT shown in figure 3 in section 2.2, where the initial order of the input data points is in-order.

The following algorithm is useful in recognizing the pairs of sensor that would be performing a butterfly exchange for each of the $log_2 N$ stages of the FFT computation.

1. Let $k = 1, 2 \ldots m$ be the stages of FFT computation, where $N = 2^m$. From left to right $\frac{N}{2^{k-1}}$ consecutive sensors be grouped together, from left to right. Where the sensor nodes physical IDs range from 0 to N-1.
2. In each these 2^{k-1} groups thus formed in each stage, the data points contained in sensors that are $\frac{N}{2^k} - 1$ sensors distance apart are the butterfly pairs for that stage.

Let the complex data point, held originally before the butterfly exchange, in the sensor to the spatial-left in the butterfly pair of sensors, be the data point $'a'$ and the complex data point in the other sensor be $'b'$. The expressions $a + b * W$ and $a - b * W$, needs to be computed, where W is the twiddle factor with those butterfly pair of data points referred to as ‘butterfly computation’. This is as explained in section 2.2. Two possible methods of achieving this computation are discussed.

Method one is during the butterfly exchange the initial pair of data points $'a'$ and $'b'$ are exchanged by the two sensors. After the exchange both sensors now contain $'a'$ and $'b'$. Sensor to the spatial left performs and stores the complex computation $a + b * W$ and the other sensor $a - b * W$. It is important to notice that the complex multiplication $b * w$ is performed redundantly by both sensors in this method, but this gives a balanced computation approach. Diagrammatic representation for method one is shown in Figure 5.

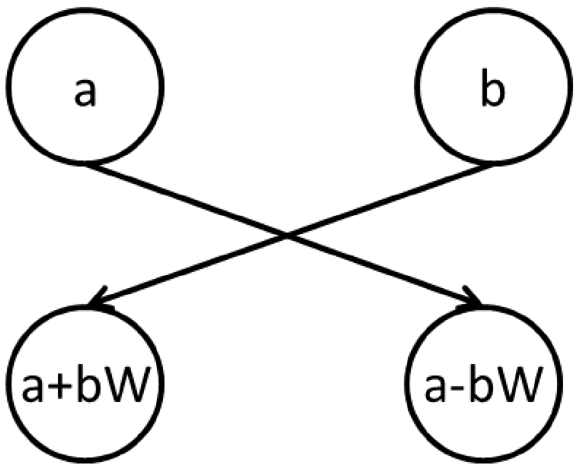

Figure 5. Method one of butterfly computation in conventional in-place approach.

In Method two the sensor on the spatial right that holds the complex data point $'b'$ and does the complex multiplication $b * W$. Then during the butterfly exchange $'a'$ and $b * W$ are exchanged. After the exchange both sensors now contain $'a'$ and $b * W$. Sensor to the spatial left now performs and stores the complex addition $a + b * W$ and the other sensor does the complex subtraction $a - b * W$. This method is better in that it reduces a redundant multiplication being performed by both the sensors in the previous method which is now performed by only one of the sensors. Diagrammatic representation for method two is shown in Figure 6.

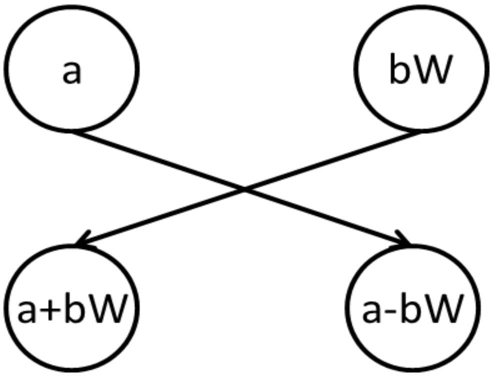

Figure 6. Method two of butterfly computation in conventional in-place approach.

The communication exchange among the sensors for a 16-point FFT computation is illustrated in figure 7.

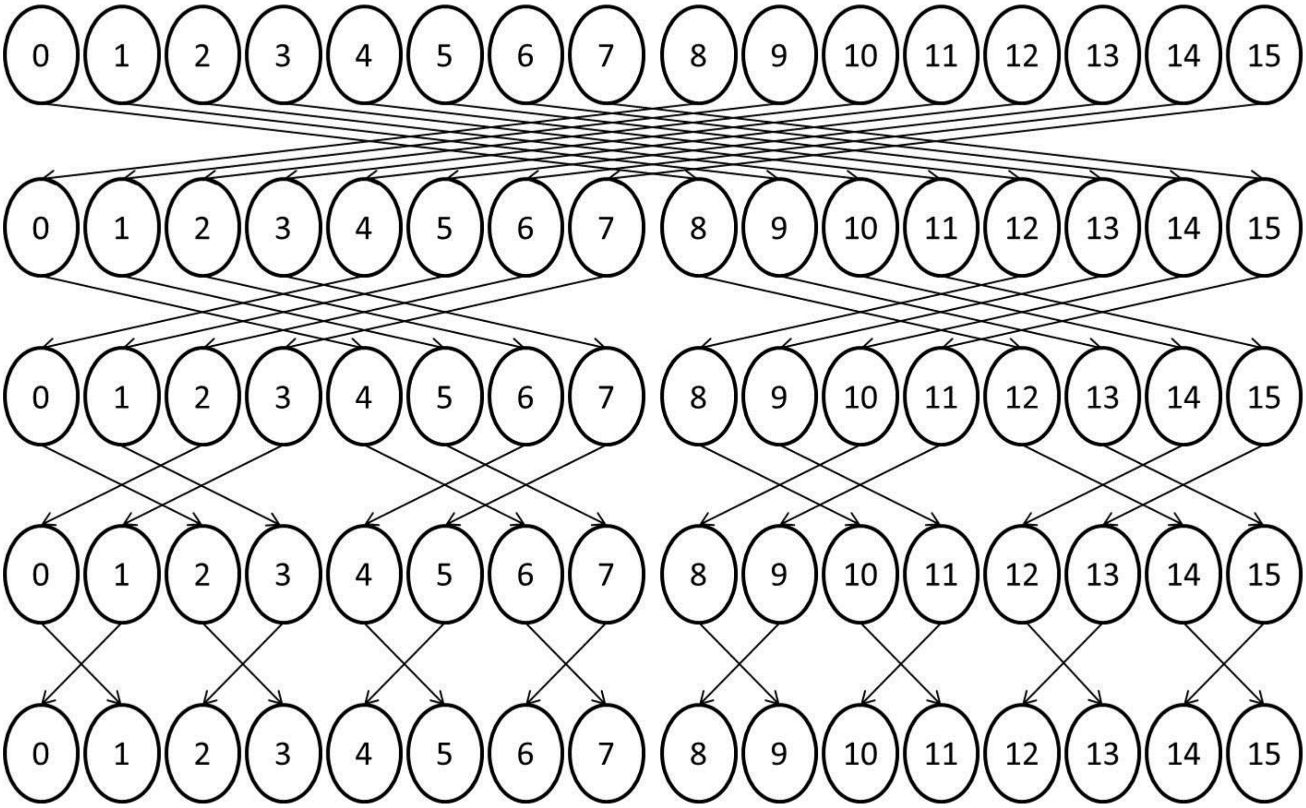

Figure 7. Conventional in-place approach communication exchange diagram

3.2 Inverse-Shuffle Complement Approach

A distributed implementation of a DIT N-point 1-D FFT on WSN was also put forward in (2). The proposed computation method is similar to the conventional in-place approach in (1) described in section 3.1, in that this is also an in-place approach. This is because in this approach also each of the N sensors holds one of the N data points in them and after butterfly computation replaces them with a newly computed data point. The sensors holding the butterfly pairs of data points exchange their data points and do the butterfly computation for each stage which is again similar to the approach in (1).

In section 2.2 where DIT FFT is explained, there are two alternative arrangements of data point performing the same set of computation illustrated in the flow charts that are shown in figure 2 and 3. The inverse-shuffle complement approach follows the communication pattern and the data point arrangement in flow chart in figure 2 where the input is assumed to be bit reversed arranged. Whereas the conventional in-place approach is modeled on flow chart shown in figure 3 where the input is assumed to be in-order as explained in section 3.1. There is no explanation in the inverse-shuffle complement approach paper (2) about how initially the input gets arranged in the bit-reversed order format before processing. This pre-arrangement of the input would require a considerable number of communications to achieve.

Since the inverse-shuffle complement method follows the arrangement in the flow chart in figure 2, it can be seen that in the initial stages the sensors that are closer to each other exchange data-points. As the stage number increases, the distance between the sensors doing the butterfly exchange increases. Assuming $k = 1, 2, \ldots m$ be the stages of FFT, where $\mathrm{N} = 2^{\mathrm{m}}$ then the sensors that do a butterfly exchange are $2^{\mathrm{k}-1} - 1$ sensors apart. This is an inverse of the case in the conventional approach where they are $\frac{\mathrm{N}}{2^{\mathrm{k}}} - 1$ sensor apart. This is an advantage in that as the sensors that are closer together finish exchanging and computing in the initial stages then a rearrangement of the data can be done so that the data points that are further apart can be brought closer, so that the butterfly pairs of data-points for the later stages are closer to each other.

So a rearrangement after $\frac{log_{2}N}{2}$ stages is proposed which is termed as an inverse-shuffle complement. For this rearrangement the data held in each sensor is now placed in the sensor with the physical ID which is a bit-complement followed by a bit-reversal of the physical ID of the previous sensor node. The result of this rearrangement is that for the remaining stages of FFT computation the sensors that will be doing a butterfly exchange are now $\frac{\mathrm{N}}{2^{\mathrm{k}}} - 1$ sensors apart. This is an inverse of the stages before the rearrangement. It is proven that this reduces the overall communications when compared with the conventional in-place approach despite requiring several asymmetric multi-hop communications to carry out the rearrangement.

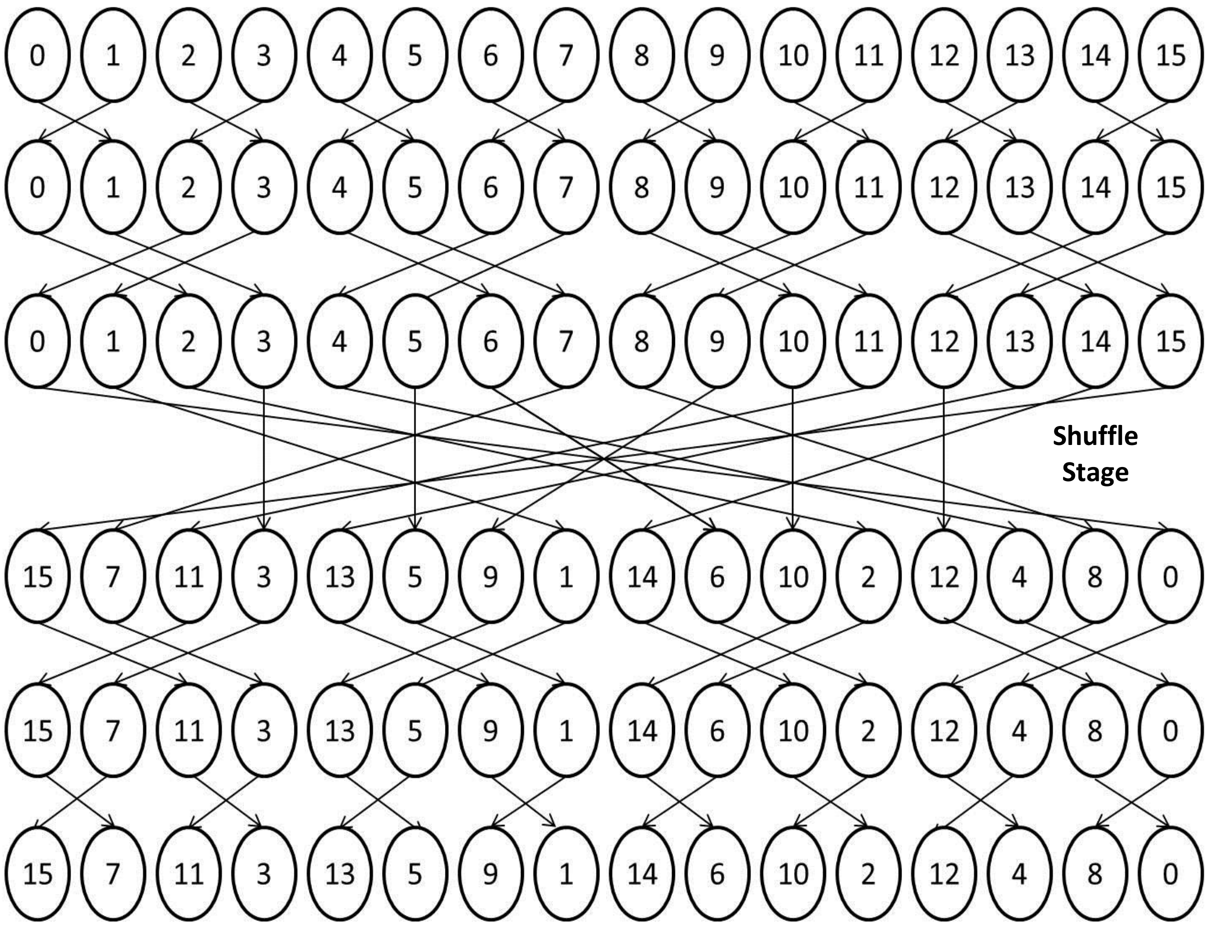

Figure 8. Inverse-shuffle complement approach communication exchange diagram

The communication exchange taking place for a 16-point FFT is illustrated in figure 6. The following algorithm is useful in identifying the pair of sensors that would be doing the butterfly exchange for each stage:

1. Before rearrangement:

   For stages $k = 1, 2, \ldots, \frac{log\ _2N}{2}$

   - From left to right 2^k consecutive sensors be grouped together in each stage.
   - In each these $\frac{N}{2^k}$ groups thus formed, the data points contained in sensors that are $2^{k-1} - 1$ apart are the butterfly pairs for that stage

2. After rearrangement:

   For stages $k = \frac{log_{\,2}N}{2} + 1, \frac{log_{\,2}N}{2} + 2, \ldots, m$ where $N = 2^m$

   - Let $k = 1, 2 \ldots m$ be the stages of FFT computation, where $N = 2^m$. From left to right $\frac{N}{2^{k-1}}$ consecutive sensors be grouped together.
   - In each these 2^{k-1} groups thus formed in each stage, the data points contained in sensors that are $\frac{N}{2^k} - 1$ sensors distance apart are the butterfly pairs for that stage.

In the stages before the rearrangement, for butterfly pairs of sensors, let the sensor to the spatial left in the pair hold the complex data point $'a'$ and the sensor to the right holds the complex data point $'b'$. After rearrangement it is reversed where for the pair of butterfly sensors the sensor to the spatial right holds the complex data point $'a'$ and the other holds $'b'$.

To perform the complex butterfly computation $a + b * W$ and $a - b * W$, where W is the twiddle factor, first the sensor that holds the complex data point $'b'$ does the complex multiplication $b * w$ in all the stages. Then the data point $'a'$ and $b * W$ are now exchanged between the butterfly pair of sensors so that after the exchange both sensors contain $'a'$ and $b * W$ within them. The sensor held data point $'a'$ before the exchange does the complex addition $a + b * W$ and the sensor that held $'b'$ does the complex subtraction $a - b * W$. As a result of the rearrangement after $\frac{log_{\,2}N}{2}$ stages, it is shown that every sensor performs equal number of multiplication and additions. So there is a uniform distribution of computational load among all the sensors. In the illustration of the 16-point FFT using the inverse-shuffle complement approach shown in figure 6, the sensors doing the complex multiplication in each of the stages are highlighted to show how the multiplication load gets uniformly distributed among the sensors.

CHAPTER 4

PROPOSED REDUCED-COMMUNICATION APPROACH

In the conventional in - place approach outlined in section 3.1, several redundant communications can be seen to take place. Let us assume stage numbers are denoted by $k = 1, 2, \ldots m$, where $N = 2^m$ and m is a positive integer. In each of the $\frac{N}{2^{k-1}}$ group of consecutive sensors described in section 3.1, the center $\frac{N}{2^k}$ sensors form the mid-points (equal-hop distance), of all the butterfly pairs of sensors that are $\frac{N}{2^k}$ hops apart. In any of the groups of sensors, for any butterfly pair of sensors, one of the constituent sensors is always at the midpoint of some other butterfly pair of sensors. The sensor that is not at the mid-point of some other butterfly pair of sensors in the present stage will form a butterfly pair in the next stage with the very same sensor that is at the mid-point of its present butterfly pair. So the sensor, with which a butterfly exchange will take place in the next stage, lies at the exact center of the pair of sensors which are performing the butterfly exchange in the present stage. Meaning that, in the present stage, the data-point exchange for a butterfly pair traverses through a particular sensor, with which the computed data-point is again going to be exchanged with during the next stage. This would mean that the intermediate nodes in between butterfly exchanges can be used to do some of butterfly computations and thereby save several redundant communications that are taking place.

To elucidate further the redundancy in communication during the conventional in–place implementation, an example of the butterfly pair of node 0 and node 8 in stage 1 of a 16–point FFT illustrated in figure 7 is considered. The resultant left-half data point of the butterfly operation between these nodes is stored in node 0. In stage 2, the resultant data-point of stage 1 that is in node 0 is exchanged with node 4, which forms the new butterfly pair with node 0. In a multi-hop arrangement, the butterfly exchange of data-points between node 0 and node 8 will pass through node 4. So the value computed by node 0 in stage 1 and later sent to node 4 in stage 2, could have instead been computed and stored in node 4 during stage 1 itself. This would have saved several redundant communications from taking place in stage 2. Also, node 4 is at equal hops distance from node 0 and node 8, making it the ideal sensor for performing the butterfly computation between the node 0 and node 8 data points.

This redundancy in radio communication, caused by not choosing the best possible sensor to perform the computation of a butterfly pair of data, takes place in every stage of the implementation with all the butterfly pairs.

In the proposed reduced-communication approach, it is proposed that for every stage, the sensor that would have been at the mid-point between the butterfly pair of sensors in the conventional in - place approach, be the one that is used to perform the entire butterfly computation, for that butterfly pair of data-points.

As the butterfly pairs in the last stage of the in-place approach are adjacent to each other, no midpoint can be found between them. So for the last stage, in the proposed method, the same sensors as the penultimate stage are used, to do the butterfly computations. This is convenient as the butterfly pair of data points required to do the butterfly computation for the last stage, can be brought into the right sensors, by an exchange of single data points between adjacent sensors.

In the first stage it is assumed initially that all the sensors contain the data point sensed by itself at the same time interval. And the FFT processing is done on these sensed equally spaced spatial data points. So initially every sensor contains one data point.

In all the stages of the N-point 1-D FFT computation on N sensors, specific $\frac{N}{2}$ sensors will be performing computation and can be identified by the following steps. Assuming stage numbers are denoted by $k = 1, 2, \ldots m$, where $N = 2^m$. The exception to the rule explained below is for the last stage $k = m$, where the computing sensors are the same ones as in the stage $k = m - 1$

1. $\frac{N}{2^{k-1}}$ Consecutive sensors form groups in each stage of the computation.

2. For each group, the center $\frac{N}{2^k}$ nodes will be the ones storing the one butterfly pairs of data each and performing both the butterfly computations.

The computing sensors in this algorithm receive and store two complex butterfly data points required for the corresponding butterfly computation. Let the complex data-points received in each sensor be $'a'$ and $'b'$, where $'a'$ is the data-point received from the sensor which is spatially to the left from among the two sensors that are transmitting to the present sensor. The computing sensors will perform both the butterfly computations $a + b * W$ and $a - b * W$, where W is the twiddle factor . They will now hold both the computed data sets, to be sent to two different sensors for the next stage computation.

The last stage is an exception, where one of the data points already exists in the sensor and other is received. The sensor node with the odd physical ID has 'a' already in it and receives 'b' and the even node has 'b' already stored and receives 'a'. Both $a + b * W$ and $a - b * W$ is then computed.

Before the butterfly computation can take place, the corresponding pairs of butterfly data points needs to be placed in the right sensors. The communication exchange to do this is outlined in the following pseudo-code. It is to be noted that the stage number $k = 1,2,..m$ where $N = 2^m$ is incremented when butterfly computation for that stage is completed. The sensors *NodeID* ranges from 0 to $N - 1$.

1. **if** $k == 1$

   if $\frac{NodeID}{\frac{N}{2}} < 1$ **send data point to** $NodeID + \frac{N}{4}$

   else **send data point to** $NodeID - \frac{N}{4}$

2. **if** $k != 1$ && $k != m$

   if $\left(NodeID \;\%\; \frac{N}{2^{k-2}}\right) < \frac{N}{2^{k-1}}$

   send $a + b * W$ **to** $NodeID - \frac{N}{2^{k+1}}$

   send $a - b * W$ **to** $NodeID + \frac{N}{2^{k+1}} + \frac{N}{2^k}$

else

send $a + b * W$ **to** $NodeID - \left(\frac{N}{2^{k+1}}\right) - \frac{N}{2^k}$

send $a - b * W$ **to** $NodeID + \frac{N}{2^{k+1}}$

3. **if** $k == m$

   if $NodeID$ **is** odd

   send $a - b * W$ **to** $NodeID + 1$

   else

   send $a + b * W$ **to** $NodeID - 1$

Finally a rearrangement is performed to place the final resultant N data points of the FFT after m stages, in the actual sequential computed DIT FFT order, such that each sensor now holds one data point. For this rearrangement of data-points the rule being that from among the sensors that have finished computing in the penultimate stage, the sensor node with an odd numbered physical ID sends the computed complex data-point $a + b * W$ to the sensor node on its spatial-left and the even physical ID sensor sends $a - b * W$ data-point to the sensor node on the spatial-right.

To better understand the algorithm, a stage-by-stage communication exchange for a 16–point implementation is illustrated in figure 9. In the first stage, the center 8 nodes perform the computation, and the communication exchange required to bring the butterfly pair of data points into them, involves all 16 sensors. The communication pattern in the second stage and third stage is the main pattern for the bulk of the stages of the algorithm. The last stage reuses the same sensor as the penultimate stage for performing computations and requires only one-hop communications for arranging the right data points into the right sensors. A final rearrangement stage, after which each sensor will hold one of the N complex data point, for the sake of placing the data points in the DIT FFT computed sequence.

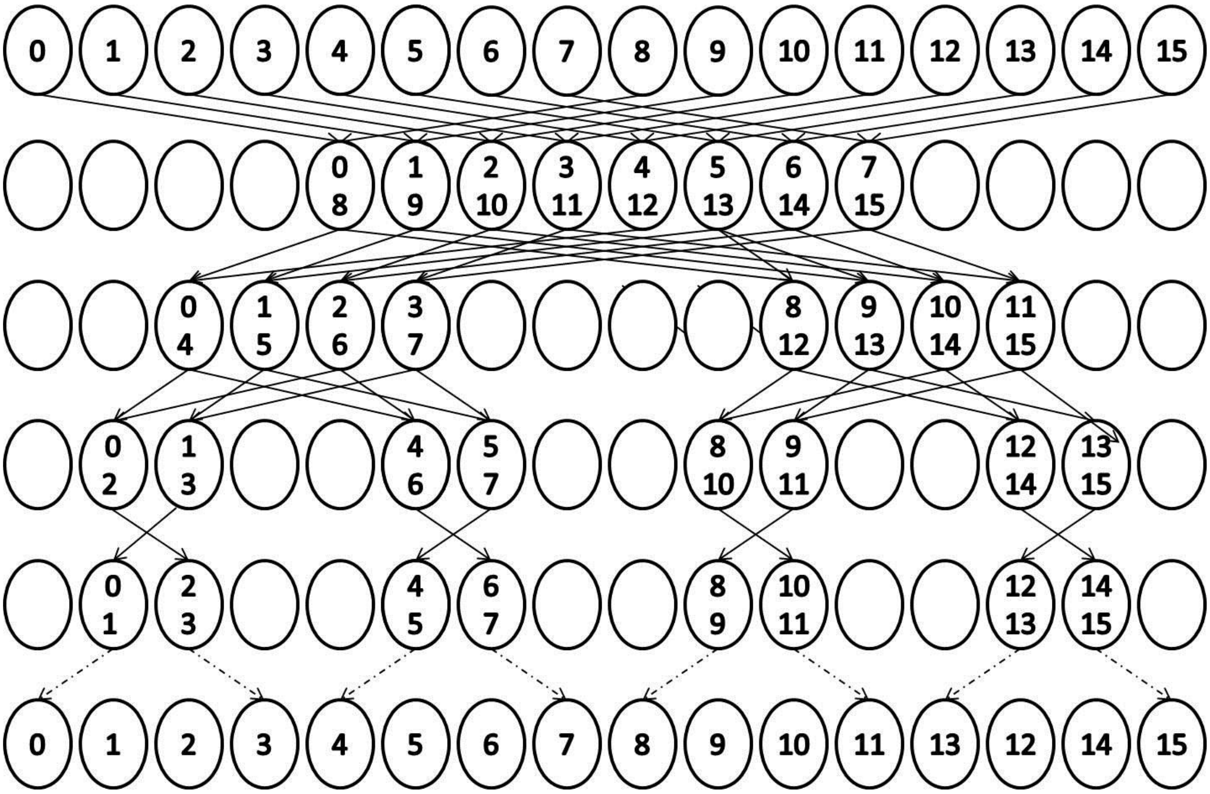

Figure 9. Proposed reduced-communication approach communication diagram

CHAPTER 5

PERFORMANCE COMPARISON

This chapter deals with a comparison of the performance of the proposed reduced-communication algorithm and the two previously published algorithms that were discussed in chapter 2.

5.1 Comparison with Inverse-Shuffle Complement algorithm

For the inverse-shuffle complement algorithm proposed in (2), described in section 3.2, there would be a need to pre-arrange the data-points into a bit-reversed order before computation could be performed as described. The method to do this pre-arrangement of input data-points is not discussed in (2).

The inverse-shuffle complement approach is shown to be 36% more power-efficient than the conventional in-place approach, but this is done without taking into account the cost for the pre-arrangement of the input data-points. Factoring in the communication energy cost for the pre-arrangement would significantly increase the power consumed.

No such pre-arrangement of the data is required in the proposed reduced-communication method in this thesis. For this reason, a fair comparison cannot be made in the performances of the two methods.

5.2 Comparison with Conventional In-Place algorithm

5.2.1 Theoretical Comparison

In each of the stages $k = 1,2,\dots,m$, where $N = 2^m$ of the conventional in-place approach outlined in section 3.1, the butterfly sensor nodes are $\frac{N}{2^k} - 1$ sensors apart. So each butterfly exchange requires $2 * \frac{N}{2^k}$ transmissions and $2 * \frac{N}{2^k}$ receptions. In each stage there are a total of $\frac{N}{2}$ butterfly pairs. Let us assume that $TCost$ represents transmission energy cost and $Rcost$ reception energy cost. So assuming an ideal synchronized lossless communication the total communication energy cost for all the stages in the in-place approach is given by:

$$N^2(TCost + RCost)\sum_{k=1}^{m}\frac{1}{2^k} \tag{5.1}$$

For proposed reduced-communication there are four cases that need to be considered while calculating the total number of communications.

1. First stage $k = 1$: In the first stage the butterfly pair of data points is $\frac{N}{2} - 1$ sensors apart and each pair requires $2 * \frac{N}{4}$ transmissions and $2 * \frac{N}{4}$ receptions for them to be placed in the node at their midpoint.
2. All other stages except the first and last stage: For a butterfly pair of data-points to be placed in the sensor that will be computing, one of data point requires $\frac{N}{2^{k-1}}$ transmissions and receptions and the other data point requires $\frac{N}{2^k} + \frac{N}{2^{k-1}}$ transmissions and receptions. So each butterfly pair requires $2 * \frac{N}{2^k}$ transmissions and $2 * \frac{N}{2^k}$ receptions.
3. Last stage $k = m$: For each butterfly pair one of the data points is already stored in the sensor that will be computing. The other data point needs just a single hop communication. So each butterfly pair requires one transmission and reception.
4. Final rearrangement: After the last stage of the FFT computation, $\frac{N}{2}$ sensors contain two data points and remaining $\frac{N}{2}$ does not hold any data points. For the final rearrangement each of the $\frac{N}{2}$ sensors holding two data-points needs to make a single hop communication of one of the data points to the adjacent sensor node. This results in a total of $\frac{N}{2}$ transmissions and $\frac{N}{2}$ receptions.

Since there are $\frac{N}{2}$ butterfly pairs in all stages of FFT computation, the total communication cost for all the stages of FFT and final rearrangement stage in the proposed reduced-communication algorithm is given by:

$$N^2(TCost + RCost)\left(\frac{1}{4} + \sum_{k=2}^{log_2N-1} \frac{1}{2^k} + \frac{1}{2N} + \frac{1}{2N}\right) \quad (5.2)$$

The savings in the total number of communications for the proposed reduced-communication algorithm when compared with the conventional in-place approach is given by the subtraction (5.1) - (5.2) which is:

$$\frac{N^2}{4}(TCost + RCost) \quad (5.3)$$

It can be seen that the savings in the overall communication is proportional to a square of N.

In the conventional in-place approach, for every packet travelling in one direction there is always a packet travelling in opposite direction towards it, in every stage due to the exchanges taking place. In the proposed algorithm, in every stage of the FFT computation, only half the packets have a packet travelling in the opposite direction towards them. When number of pairs of packets travels towards each other and overlap each other during an exchange, the sensors at the middle of the exchange have significantly higher communication load and there is a higher likelihood of packet collision. So the proposed algorithm considerably reduces the probability of packet collisions and provides a much better distributed communicational load.

A numerical count of the number transmissions and reception of each sensor node for the proposed reduced-communication algorithm and also for the conventional in-place algorithm is done assuming an ideal case of no packet loss.

The node that has the maximum communication load in the network will be the one that uses up the maximum energy from its battery among all the sensors in the network. This is because radio communication is the major consumer of power of the wireless sensor node. So the lifetime of the network is decided by the lifetime of the node with the maximum communication load. In order to compare the lifetimes of the network for the proposed reduced-communication algorithm versus the conventional in-place algorithm, a comparison is made of the sum of transmission and reception of the node that does the maximum number of communication in each of the algorithms for varying N-points. This is plotted in figure 10.

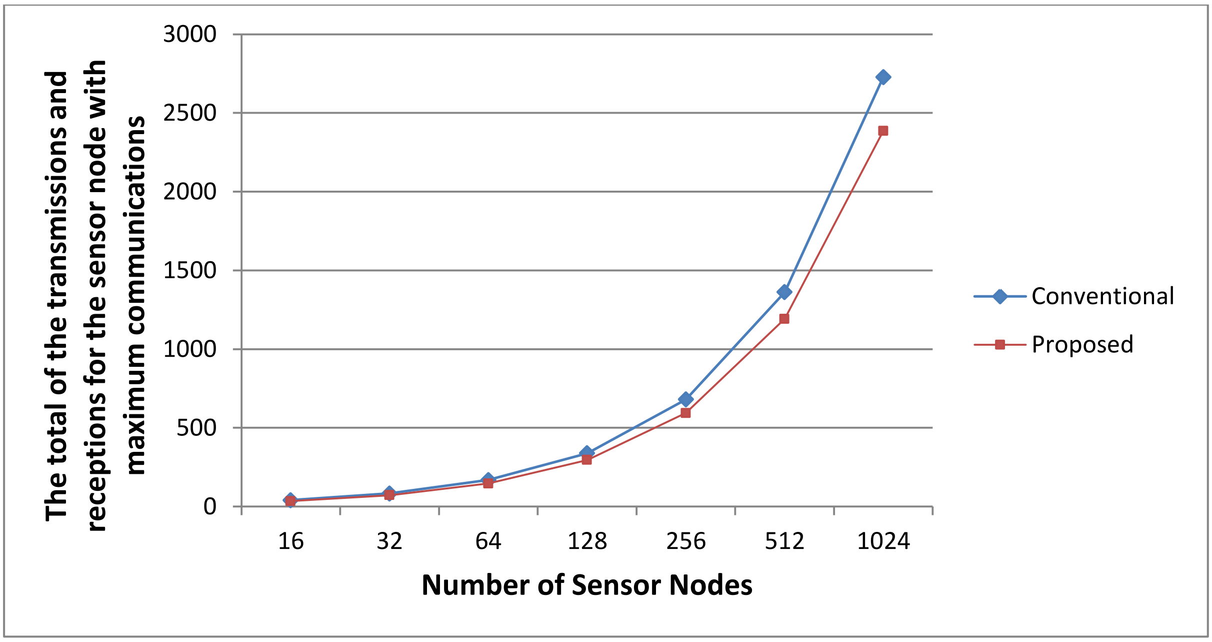

Figure 10. Plot comparing the node with the maximum communication load among all the nodes for varying N points between the conventional in-place approach and the proposed reduced-communication approach.

In the proposed reduced-communication implementation, at the end of each stage, the sensors that perform the computations hold two resulting data points each. After computation the data points need to be rearranged into the appropriate sensors for the next stage computation. Since the present stage sensors hold two data points each and both of them need to be placed into different sensors for the next stage, there is a need to transmit twice by the sensors, except in the last and the first stage. So a delay will occur before second data point can be transmitted, as only one transmission can take place at a given time instant. This delay can be negated by transmitting the data point that travels a longer distance first. This is because the longer distance travelling data points form the bottle neck for the next stage computation. It is to be noted that the sensors that will have to perform double transmissions in the proposed algorithm will have to do the same in the conventional in-place approach too. These sensors would in the conventional in-place approach have formed the mid-point in the exchange of butterfly pairs of sensors, for the same stage. So in an ideal synchronized communication scenario, both the data points exchanged will theoretically arrive at the midpoint sensor at the same time and will both wait to be transmitted at the same time.

The computation load in the proposed algorithm becomes comparatively more non-uniform as a trade-off to the elimination of several redundant communications taking place in the in-place approach. There is a trade-off between significant gains in-terms of reduction in communication load and the comparatively smaller loss in energy cost due to a slightly more non-uniform distribution of the computation load. This trade-off has been discussed in (4, 10). It is shown that communication cost drastically exceeds the computation cost, so there is an overall gain in power efficiency. This is confirmed by the experimental comparison discussed in section 5.2.2.

5.2.2 Experimental Results

A simulation of the proposed FFT implementation on WSN was done on SIDnet-SWANS design environment. The SIDnet-SWANS design environment consists of a user-friendly graphical user interface (GUI) called SIDnet for runtime verification of the simulation. The SIDnet GUI is wrapped on to the JiST-SWANS WSN simulator. The performance of JiST-SWANs has been verified and shown to be comparable to and better than the well-known WSN simulator ns-2 in terms or memory and time consumption in (9).

A simulation also of the conventional in-place FFT implementation that was described in section 3.1 is done in order to show a comparative improvement of the proposed implementation.

The mote characteristics are that of Mica MPR500CA. Battery power is assumed to be 75mAh for the nodes. The physical layer and the MAC layer are the IEEE 802.15.4 standard implementation available in the simulator library. The routing layer is a multi-hop shortest path routing with each node only able to communicate with its immediate neighbors on either side of it. The topology is of an equispaced 1-D uniform horizontal array of 16, 32 and 128 nodes in a field of length of 400, 800 and 1600 meters respectively. Packet re-transmission is only due to packet collision. Free space path loss is assumed. No fading model is assumed due to limited transmission range.

Table I shows the comparison of the maximum and the average percentage battery consumed per iteration by the proposed reduced-communication algorithm and the conventional in-place for varying N. The percentage gain the proposed reduced-communication algorithm has over the conventional in-place algorithm in terms of overall average power consumption throughout the network is very significantly high of around 20%. The improvement in the battery power consumption of the node that consumes the maximum battery power in the network by around 10% signifies that the lifetime of the network is extended by 10%. Also there is a trend of increasing comparative gain in power-efficiency of the proposed algorithm as N increases.

N	**Per Iteration Of FFT Computation**	**Conventional In-Place Approach**	**Proposed Approach**	**Gain%**
16	Maximum Battery Percent Consumed	0.009769	0.00894	8.48
64	Maximum Battery Percent Consumed	0.02244	0.02023	9.84
128	Maximum Battery Percent Consumed	0.04703	0.042009	10.67
16	Average battery Percent Consumed	0.00804	0.00648	19.4
64	Average battery Percent Consumed	0.017858	0.01387	22.33
128	Average battery Percent Consumed	0.03641	0.02793	23.29

TABLE I: COMPARING SIMULATION POWER CONSUMPTION BETWEEN CONVENTIONAL IN-PLACE AND PROPOSED APPROACH.

Table II shows the comparison of the time taken per iteration by the proposed reduced-communication algorithm and the conventional in-place for varying N. Here again an improvement in processing latency of around 7% per iteration can be observed.

N	**Per Iteration Of FFT Computation**	**Conventional In-Place Approach**	**Proposed Approach**	**Gain%**
16	Time taken (msec)	1560.36	1524.89	2.27
64	Time taken (msec)	3822.77	3425.49	10.39
128	Time taken (msec)	7845.69	7243.99	7.66

TABLE II: COMPARING SIMULATION TOTAL PROCESSING LATENCY BETWEEN CONVENTIONAL IN-PLACE AND PROPOSED APPROACH.

Table III shows the comparison of the packet delivery latency in the two algorithms. The improvement of around 30% in both the maximum and the average packet delivery latency is very significant in proving that the proposed algorithm has a very much improved communication performance in terms of reduction in packet collisions, channel availability, reduced number of overall communications and shortened distances that packets need to travel.

N	Per Iteration Of FFT Computation	Conventional In-Place Approach	Proposed Approach	Gain%
16	Maximum Packet Delivery Latency (msecs)	1143	803	29.74
64	Maximum Packet Delivery Latency (msecs)	2649	1908	27.97
128	Maximum Packet Delivery Latency (msecs)	4921	3681	25.19
16	Average Packet Delivery Latency (msecs)	324	233	28.08
64	Average Packet Delivery Latency (msecs)	644	418	35.09
128	Average Packet Delivery Latency (msecs)	1082	714	34.01

TABLE III: COMPARING SIMULATION PACKET DELIVERY LATENCY BETWEEN CONVENTIONAL IN-PLACE AND PROPOSED APPROACH.

Figures 11, 12, and 13 show that there is a significant comparative improvement in the power consumption of every single node on the network by implementing the proposed algorithm. Figures 14, 15, and 16 show that the trend in the radio energy cost defines the trend in the power consumption of the nodes in the network.

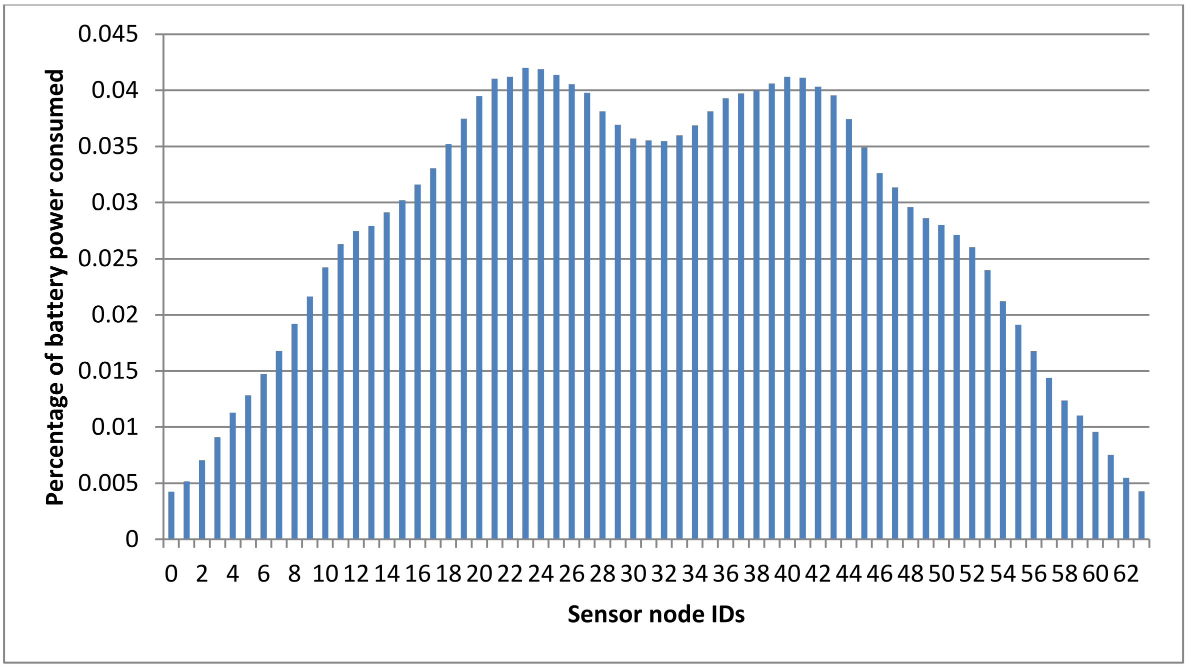

Figure 11. Plot of the percentage battery power consumption for each sensor node in a 64 point DIT FFT implementation of the proposed reduced-communication algorithm per iteration.

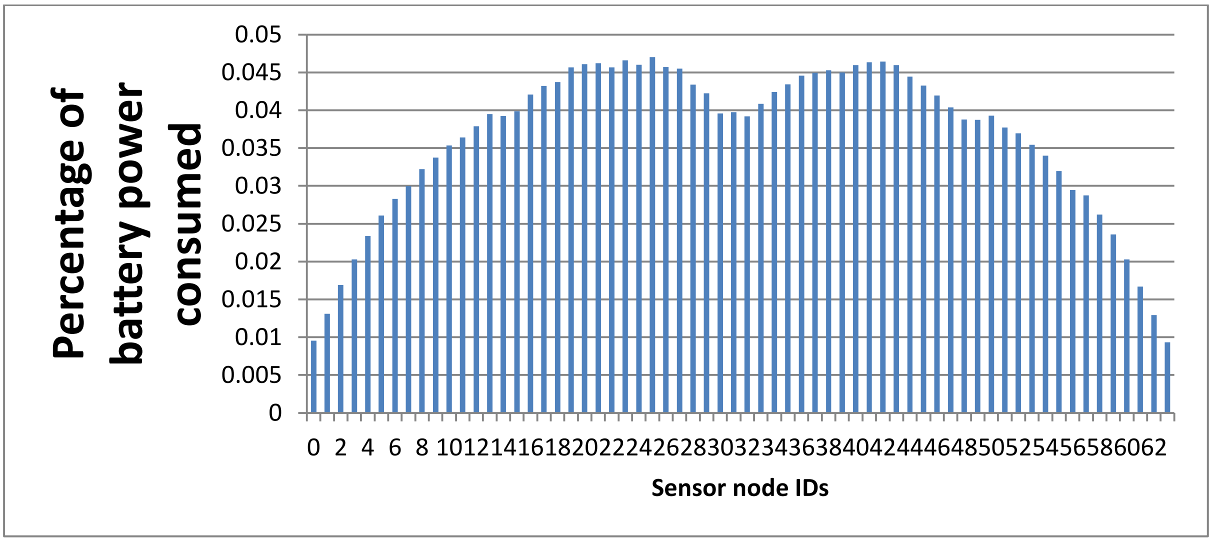

Figure 12. Plot of the percentage battery power consumption for each sensor node in a 64 point DIT FFT implementation of the conventional in-place algorithm per iteration.

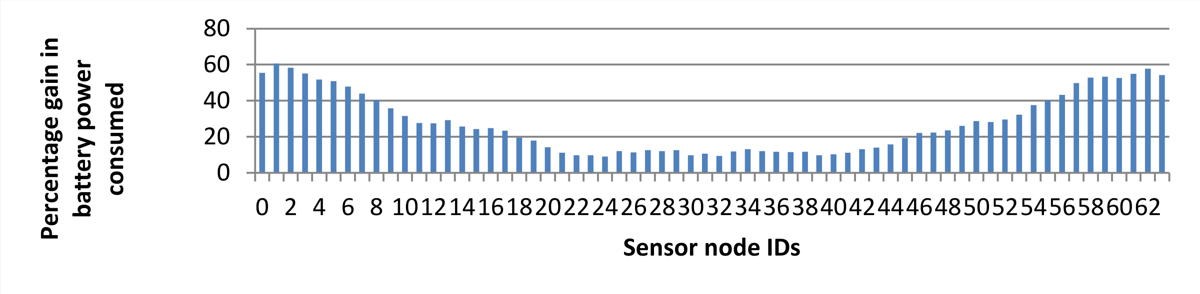

Figure 13. Plot of the sensor by sensor percentage gain in power of the proposed reduced-communication algorithm over the conventional in-place algorithm for 64 point DIT FFT per iteration.

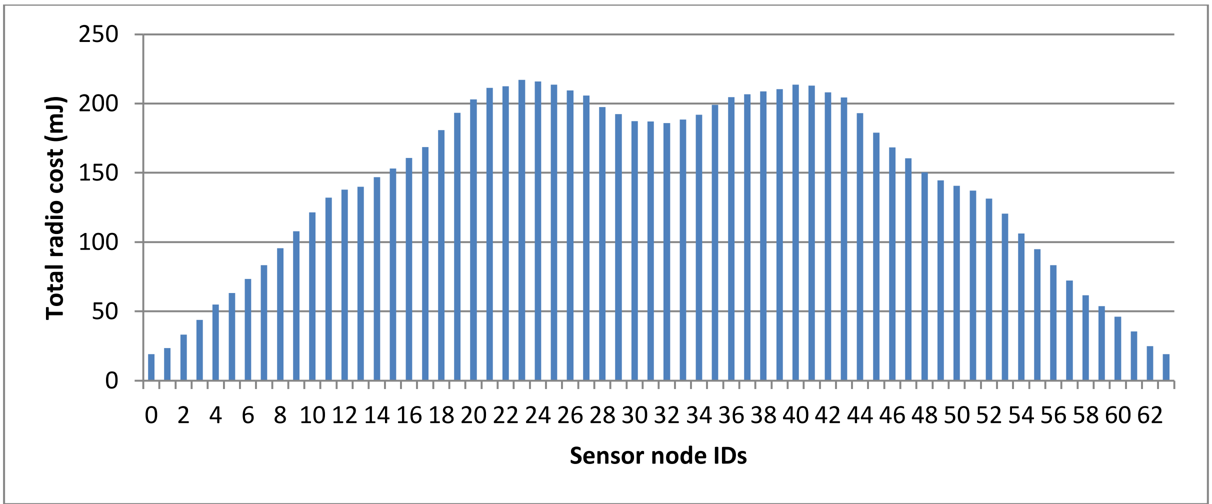

Figure 14. Plot of the radio energy cost of each sensor node for a 64 point DIT FFT implementation of the proposed reduced-communication algorithm per iteration.

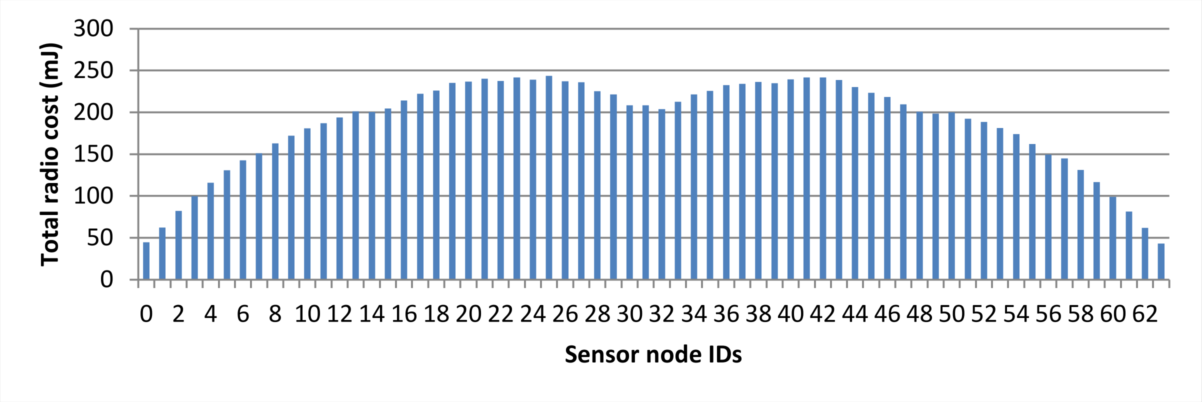

Figure 15. Plot of the radio energy cost of each sensor node for a 64 point DIT FFT implementation of the conventional in-place algorithm per iteration.

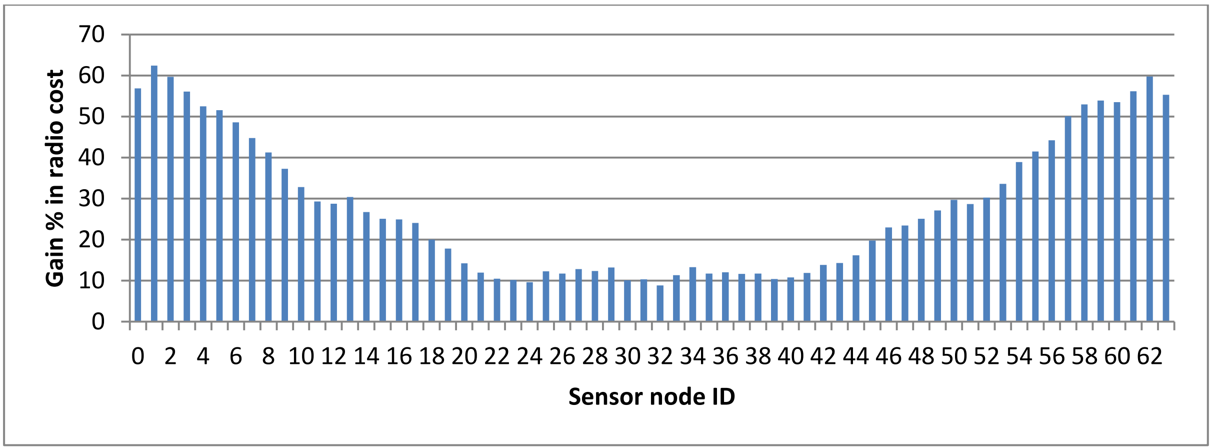

Figure 16. Plot of the sensor by sensor percentage gain in radio energy cost of the proposed reduced-communication algorithm over the conventional in-place algorithm for 64 point DIT FFT per iteration.

CHAPTER 6

CONCLUSIONS

The goal of this thesis is to design a power-efficient minimal latency distributed in-network one-dimensional FFT processing on a WSN array. This can be used for the FFT processing of equispaced spatial data points sensed at the same fixed time intervals by the WSN array, where the sensing nodes are arranged on a uniform gird

The main idea of the thesis is to select the best possible $\frac{N}{2}$ sensor nodes to do the butterfly computation for each of the $\frac{N}{2}$ butterfly pairs of data points for every stage of the FFT computation, thereby reduce the communication exchanges of the algorithm.

Performance comparison with two previously existing distributed in-network one-dimensional FFT processing algorithms on WSN, consisting of theoretical analysis and experimental results, establish that the proposed that reduced-communication algorithm results in:

1. 10% improvement in network lifetime.
2. 20% overall reduction in power-consumption on the network.
3. 7% improvement in processing time latency.
4. 30% reduction in packet delivery latency.

CHAPTER 7

FUTURE WORK

In certain terrains, it is difficult to achieve a uniform grid arrangement of the wireless sensor nodes of the WSN. For applications in such terrains the conventional FFT processing is not suitable, since it cannot accurately analyze non-equispaced spatial-temporal data. So a good extension of this thesis would be an implementation of an algorithm for one-dimensional non-uniform sampled FFT processing on the WSN. A study was conducted and the non-uniform FFT algorithm in (18) is found to be suitable and can be implemented by modifying the algorithm for the one-dimensional conventional FFT proposed in this thesis. Further extending it to a two-dimensional non-uniform FFT would be very useful in frequency analysis of data sensed randomly deployed sensors.

CITED LITERATURE

1. Chiasserini, C., Rao, R.: On the Concept of Distributed Digital Signal Processing in Wireless Sensor Networks. IEEE MILCOM: 2002.

2. Canli, T., Gupta, A., Khokhar, A.: Power Efficient Algorithms for Fast Fourier Transform over Wireless Sensor Networks. IEEE Computer Systems and Applications; 549-556: 2006.

3. Ong, K.S.H., Siew-Peng, Y., Keck-Voon, L.: Implementation of fast fourier transform on body sensor networks. Proc. Int. Conf. Body Sensor Netw.; 197–202: 2010.

4. Min, R., M. Bhardwaj, M., Chu, S., Shih, E., Sinha, E., Wang, A., Chandrakasan, A.: Low-power wireless sensor networks. VLSI Design; Invited Paper, Bangalore: January 2001.

5. Tahir, A., Farrell, R.: Optimal Communication-Computation Tradeoff for Wireless Multimedia Sensor Network Lifetime Maximization. Proc. IEEE Wireless Comm. and Network Conf. (WCNC): 2009.

6. Proakis, J. G., Manolakis, D. K.: Digital Signal Processing. Principles, Algorithms and Applications. 4th ed. Upper Saddle River, NJ; Prentice-Hall: 2006.

7. Cooley, J.W., Lewis, P.A., Welch, P.D.: The Fast Fourier Transform and its Application to Time Series Analysis: Statistical Methods for Digital Computers. Wiley, New York: 1977.

8. Loan, C.L.: Computational Frameworks for the Fast Fourier Transform. SIAM Journal, Frontiers in Applied Mathematics: 1992.

9. Schoch, E., Feiri, M., Kargl, F., Weber, M.: Simulation of ad hoc networks: ns-2 compared to JiST/SWANS. Proceedings of the First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems (SIMUTools2008'): March 2008.

10. Eswaran, S., Misra, A., La Porta, T.F.: Adaptive In-network Processing for Bandwidth and Energy Constrained Mission-oriented Wireless Sensor Networks. Technical Report, Dept. of CSE, Pennsylvania State University: October 2008.

11. Perillo, M.A., Heinzelman, W.B.: Wireless Sensor Network Protocols. International Conference On Communication Technology: 2006.

CITED LITERATURE (Continued)

12. Akyildiz, I.F., Melodia, T., Chowdhury, K.: A survey on wireless multimedia sensor networks. Comput. Netw., vol. 51: 921 2007.

13. Ian F. Akyildiz, Mehmet Can Vuran: Wireless Sensor Networks. John Wiley & Sons: 2010.

14. Raghavendra, C.S., Sivalingam, K.M., Znati, T., (eds): Wireless Sensor Networks. Springer: 2005.

15. Alemdar, H., Ersoy, C.: Wireless sensor networks for healthcare: a survey. Computer Networks 54(15): 2688–2710: 2010.

16. Zhang, C., Kurose, J., Liu, Y., Towsley, D., Zink, M.: A distributed algorithm for joint sensing and routing in wireless networks with non-steerable directional antennas. Proc. of ICNP06.

17. Sadler, C.M., Martonosi, M.: Data compression algorithms for energy-constrained devices in delay tolerant networks. Proc. of ACM SenSys: 265-278: 2006.

18. Dutt, A., Rokhlin, V.: Fast Fourier Transforms for nonequispaced data. SIAM Journal on Scientific Computing: Vol. 14: No. 6: 1368-1393: Nov. 1993.

19. Lewis, F. L., Wireless sensor networks: in Cook, D. J., Das, S. K. (Eds.).: Smart Environments: Technology, Protocols, and Applications: Wiley: 2004.

20. Oppenheim, A. V., Shaffer, R. W.: Discrete-Time Signal Processing.: Prentice-Hall, Inc., Englewood Cliffs: NJ: 1989.

21. Mitra, S. K.: Digital Signal Processing: A Computer-Based Approach: 2nd ed.: McGraw-Hill: New York: 2001.

22. Walnut, D. F.: An Introduction to Wavelets Analysis, Applied and Numerical Harmonic Analysis: Birkhauser, Basel: 2002.

23. Crossbow technology: http://www.xbow.com.

24. SunSPOT mote specifications: http://www.sunspotworld.com.

CITED LITERATURE (Continued)

25. TinyOS.: Available at http://www.tinyos.net/.

26. EYES project: http://www.eyes.eu.org/: March 2002–February 2005.

27. Shen, C., Plishker, W., Bhattacharyya, S. S., Goldsman, N.: An energy-driven design methodology for distributing DSP applications across wireless sensor networks: In Proceedings of the IEEE Real-Time Systems Symposium: 214-223: Tucson, Az.: December: 2007.

VITA

NAME:	Avinash Athirasala
EDUCATION:	B.S. Equivalent, Electronics and Communications Engineering, Jawaharlal Nehru Technological University, Hyderabad, India, 2010.
	M.S. Electrical and Computer Engineering, University of Illinois at Chicago, Illinois, 2012, under the co-advising of Professor Rashid Ansari and Professor Ashfaq Khokhar.